ENSEMBLE FEATURE SELECTION USING FEATURE RANKING METHODS

Lappeenranta–Lahti University of Technology LUT
Master’s Programme in Business Analytics, Master’s thesis

2024

Esa Henriksson

Examiners: Assistant Professor Mahinda Mailagaha Kumbure
Professor Pasi Luukka

ABSTRACT

Lappeenranta–Lahti University of Technology LUT
LUT Business School
Business Administration

Esa Henriksson

Ensemble Feature Selection Using Feature Ranking Methods

Master’s thesis
2024
79 pages, 10 figures and 12 tables
Examiner(s): Assistant Professor Mahinda Mailagaha Kumbure and Professor Pasi Luukka
Keywords: classification, feature selection, ensemble methods, ensemble feature selection

This thesis investigates the efficacy and comparative performance of single feature selection ranking methods versus ensemble feature ranking methods in the context of classification tasks. Feature selection is a crucial step in data preprocessing, significantly impacting the performance and efficiency of machine learning models. Single feature ranking methods individually assess the importance of features based on their contribution to the target variable. In contrast, ensemble feature ranking methods combine the rankings generated by multiple single methods, aiming to create a more comprehensive and robust feature selection strategy.

The primary objective of this research was to determine whether ensemble feature ranking methods provide a significant advantage over single ranking methods in terms of improving classification performance. The performance of the feature selection methods is evaluated using several classification performance metrics. To conduct this comparative analysis, a series of experiments were performed on multiple datasets with varying characteristics. The ensemble methods were constructed by combining the ranking lists provided by the individual single feature selection methods.

The empirical results of this study reveal that ensemble feature ranking methods exhibit a slight but consistent improvement in classification performance compared to single feature ranking methods. However, the difference in performance metrics is generally subtle. This suggests that while ensemble methods can provide marginal enhancements, the benefits may not always justify the additional computational complexity and effort involved in their implementation. In conclusion, this thesis provides insights into the practical implications of using ensemble feature ranking methods versus single feature ranking methods for feature selection. The findings indicate that single feature ranking methods remain competitive and viable, offering a simpler and often equally effective alternative to ensemble approaches. Future research could explore the applicability of these findings to other classification algorithms and investigate the impact of feature selection on more complex and high-dimensional datasets.

TIIVISTELMÄ

Lappeenrannan–Lahden teknillinen yliopisto LUT
LUT-kauppakorkeakoulu
Kauppatieteet

Esa Henriksson

Yhdistelmäpiirteenvalinta hyödyntäen yksittäisiä piirteenvalintametodeja

Kauppatieteiden pro gradu -tutkielma
2024
79 sivua, 10 kuvaa ja 12 taulukkoa
Tarkastaja(t): Apulaisprofessori Mahinda Mailagaha Kumbure ja Professori Pasi Luukka
Avainsanat: luokittelu, piirteenvalinta, yhdistelmämetodit

Tämä työ vertailee yksittäisten piirrevalintamenetelmien ja yhdistelmämenetelmien tehokkuutta ja suoritusta luokittelutehtävissä. Piirrevalinta on olennainen vaihe datan esikäsittelyssä, ja sillä on merkittävä vaikutus koneoppimismallien suorituskykyyn ja tehokkuuteen. Yksittäiset piirrevalintamenetelmät arvioivat yksittäisten piirteiden merkittävyyttä kohdemuuttujaan. Sen sijaan yhdistelmäpiirrevalintamenetelmät yhdistävät useiden yksittäisten menetelmien tuottamat sijoitukset pyrkien luomaan kattavamman ja vankemman piirrevalintastrategian.

Tutkimuksen ensisijaisena tavoitteena on selvittää, tarjoavatko yhdistelmämenetelmät merkittävää etua yksittäisiin menetelmiin verrattuna luokittelusuorituskyvyn parantamisessa. Piirrevalintamenetelmien suorituskykyä arvioidaan useiden luokittelusuorituskykymittarien avulla. Tämän vertailuanalyysin suorittamiseksi tehtiin sarja kokeita useilla erityyppisillä dataseteillä. Jokaiselle datasetille suoritettiin sekä yksittäiset että yhdistelmäpiirrevalintaprosessit, minkä jälkeen luokittelualgoritmi koulutettiin ja testattiin. Yhdistelmämenetelmät rakennettiin yhdistämällä yksittäisten piirrevalintamenetelmien tuottamat sijoitukset eri piirteille.

Tutkimuksen tulokset osoittavat, että yhdistelmäpiirrevalintamenetelmät parantavat luokittelusuorituskykyä hieman mutta johdonmukaisesti verrattuna yksittäisiin piirrevalintamenetelmiin. Suorituskykymittarien ero on yleensä hienovarainen. Tämä viittaa siihen, että vaikka yhdistelmämenetelmät voivat tarjota vähäisiä parannuksia, niiden hyöty ei aina oikeuta lisättyä laskennallista monimutkaisuutta ja toteutukseen liittyvää vaivaa.

Tämä työ tarjoaa arvokasta tietoa yhdistelmäpiirrevalintamenetelmien ja yksittäisten piirrevalintamenetelmien käytännön vaikutuksista piirrevalinnassa luokittelutehtävissä. Tulokset osoittavat, että yksittäiset piirrevalintamenetelmät säilyttävät kilpailukykynsä ja tarjoavat yksinkertaisemman ja usein yhtä tehokkaan vaihtoehdon yhdistelmämenetelmille. Tulevat tutkimukset voivat tutkia näiden löydösten soveltuvuutta muihin luokittelualgoritmeihin ja tutkia piirrevalinnan vaikutusta monimutkaisempiin datasetteihin.

Table of contents

1 Introduction

In this section background to the study is introduced as well as the motivation, objective and scope. Meanwhile when explaining the objective, the research questions are introduced, and the structure and possible limitations are also set.

1.1 Background

Classification is used in many applications ranging from business intelligence to medical practices and further within technological mechanisms and therefore there is a constant need to develop classification in general towards more accurate and efficient models. Through classification organizations and companies can categorize and organize their data in a more structured way and achieve a situation where crucial decisions can be made based on the data and even accurate predictions can be made. (Katsikopoulos et al 2020) Furthermore, feature selection has been one of the components which has improved the classification model accuracy, and this is where ensemble learning comes into question as it can be used for classification but also for improving feature selection (Huang, Liu & Wu 2023) Ensemble learning happens when multiple methods are combined for an output to produce a stronger, more robust predictive model for example. Rather than relying on the predictions of a single model, ensemble methods leverage the diversity of multiple models to improve prediction accuracy, generalization performance, and robustness. Ensemble learning is based on the principle that combining the predictions of multiple models often leads to better results than any individual model alone (Bolón-Canedo & Alonso-Betanzos 2019)

The idea of this research is to compare the results of different methods that are using data from aggregated rankings through a feature selection ensemble and investigate whether the aggregated ranking provides improved results compared to just having a single ranking. Aggregated ranking is a method used to combine the rankings or scores produced by multiple individual ranking algorithms or models into a single unified ranking. Basically, the research would compare whether a feature ensemble would result in improvements in different aspects. A feature selection ensemble, also known as a feature selection ensemble method, is an approach that combines multiple feature selection techniques to identify a subset of the most informative features from a given dataset. Similar to traditional ensemble learning methods that combine multiple models to improve predictive performance, feature selection ensembles aim to enhance the effectiveness and robustness of feature selection by leveraging the diversity of multiple feature selection algorithms or strategies (Huang, Liu & Wu 2023). Hence, the motivation for research is to discover possible differences between single methods and ensemble methods and whether there is a clear dominance for using ensemble methods over single methods.

1.2 Objective and scope

Primary objective of this thesis is to explore and study whether building feature ensembles serves the end purpose of achieving better classification results. Furthermore, understanding how ensembles work within feature selection will help to gain knowledge about the usage on other instances and within machine learning as a whole. The aim is to answer the following questions:

- Do feature subsets generated by ensemble feature ranking provide higher classification performance compared to a single feature selection method?
- How does the selected aggregation method impact classification performance?

An aggregation method, in the context of machine learning and data analysis, refers to a technique used to combine multiple sources of information or predictions into a single unified output. Aggregation methods are employed to harness the collective knowledge or insights from diverse sources, such as individual models, algorithms, or experts, to improve the accuracy, reliability, and robustness. In ensemble learning, aggregation methods combine the predictions or decisions of multiple base learners (individual models) to produce a more accurate and robust prediction. Ensemble methods, such as bagging, boosting, and stacking, leverage aggregation techniques to integrate diverse predictions and exploit the wisdom of the crowd. (Bolon-Canedo & Alonso-Betanzos 2019) By aggregating predictions from multiple models, ensemble methods can mitigate the shortcomings of individual models, reduce variance, and improve generalization performance.

The first question aims to research whether there are significant differences between ensembles and single methods and naturally includes the challenge of choosing the methods for the ensemble as well as the single ranking method which the ensemble is evaluated against. For clarity, within this thesis the ensemble is compared several single ranking methods. The second question then again continues from the first one and tries to grasp the challenge of choosing the best aggregation method. There are guidelines for choosing the most suitable method within feature selection, within classification and within other instances too, but sometimes a compromise has to be made.

When it comes to the scope, the main thing is to compare whether ensembles provide better results compared to just single methods. However, there could be many other choices or things that have an effect on the final outcome such as classification algorithms or datasets, but here we focus mainly on the comparison of ensemble methods versus single ranking methods within feature selection.

Timewise the scope is quite flexible, as the datasets could contain data from several years and furthermore data that is used is from open sources, making it easily accessible and possible to replicate the study. From a data point of view sets with multiple features and instances are favoured, as they provide more possibilities for feature selection.

1.3 Structure

This thesis as a whole is constructed out of 8 Chapters, each containing a brief introduction about the section itself. First an introduction to the whole thesis is given with the objective and the actual motivation in Chapter 1. Secondly the theory behind feature selection is explained in Chapter 2 and after that the theory behind classification is presented in Chapter 3. Chapter 4 builds on existing literature review about the topic and within Chapter 5 the data used in the research is presented. After that the methodology behind the comparison is explained in Chapter 6 and in Chapter 7 the results of the comparison are presented. Lastly any conclusions and future research remarks are given.

2 Feature selection

In this chapter background for feature selection in general is given in order to understand the different limitations and on the other hand goals and targets of feature selection. Furthermore, some real-world examples of feature selection usage are given to emphasize the importance of the idea behind feature selection and how it can affect different industries.

2.1 Types and methods of feature selection

In general feature selection methods can be separated into three main categories: wrapper methods, embedded methods, and filter methods (Guyonn and Elisseeff 2003).

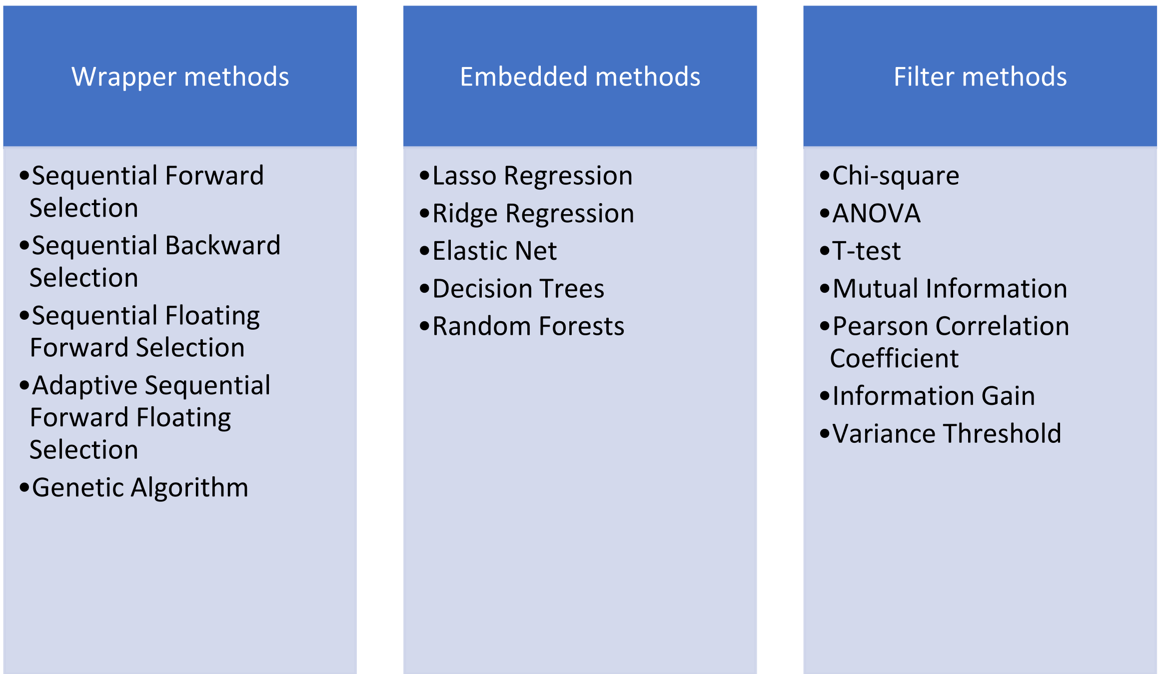

Figure 1. Feature selection methods by different categories. (Chandrashekar & Sahin 2013)

Figure 1 above breaks down different feature selection methods into different categories. Each naturally has its own advantages and disadvantages as well as use cases. This section will explore these methods in detail and provide examples of when to use each method.

2.1.1 Wrapper methods

Wrapper methods are a class of feature selection techniques used in machine learning and statistics to determine which subset of features or variables in a dataset are most relevant for building a predictive model (Kohavi & John 1997). Wrapper methods evaluate feature subsets by training and testing the model using different combinations of features. This process can be computationally expensive, but it often leads to more accurate feature selection.

Here's how wrapper methods work (Kumar & Bharti 2019):

1. Subset generation: Wrapper methods generate different subsets of features by selecting a subset of features from the original dataset. These subsets can range from a single feature to all the features in the dataset.
2. Model training and evaluation: For each feature subset, a machine learning model is trained and evaluated using a performance metric (e.g., accuracy, F1-score, or cross-validation). The model's performance is measured on a validation dataset or through cross-validation to get a more accurate estimate.
3. Iterative search: Wrapper methods typically use a search strategy to explore different feature subsets. Common search algorithms include exhaustive search (trying all possible combinations), forward selection (starting with no features and adding one at a time), backward elimination (starting with all features and removing one at a time), and recursive feature elimination (removing the least important feature iteratively).
4. Performance assessment: The quality of each feature subset is assessed based on the model's performance on the validation dataset. The goal is to find the feature subset that maximizes the chosen performance metric.
5. Stopping criteria: The search process continues until a stopping criterion is met, such as a predefined number of features, a certain level of model performance, or a specified runtime (Li et al, 2017).

Advantages of wrapper methods include that they consider feature interactions, which can lead to more accurate feature selection. They are also model-specific, making them suitable for optimizing the performance of a particular machine learning algorithm and they can provide a clear indication of which features are essential for a specific task (Venkatesh & Anuradha 2019).

When thinking about the disadvantages, the methods can be computationally expensive, especially for a large number of features, as they require training and evaluating multiple models. They may not work well with high-dimensional datasets, as the search space grows exponentially with the number of features. Overfitting can also occur if the search process is not carefully controlled (Seijo-Pardo, Bolón-Canedo & Alonso-Betanzos 2015).

Common wrapper methods include Recursive Feature Elimination (RFE), Forward Selection, Backward Elimination, and Genetic Algorithms. The choice of wrapper method and the performance metric used for evaluation depends on the specific problem and dataset, and it often requires experimentation to find the best feature subset for a given task.

2.1.2 Embedded methods

Embedded methods for feature selection are techniques that incorporate feature selection as an integral part of the model building process. Unlike filter and wrapper methods, which perform feature selection as a separate step, embedded methods select the most relevant features while the model is being trained. This approach often results in more efficient and accurate feature selection, as it considers the interactions between features and the model's learning process (Guyon & Elisseeff 2003). The key aspect of embedded methods is that the feature selection is directly integrated into the model training process. As the model learns from the data, it assesses the importance of features and selects the most relevant ones.

Examples of Embedded Methods (Pudjihartono, Fadason, Kempa-Liehr & O’Sullivan 2022; Chandrashekar & Sahin 2013):

- L1 Regularization (Lasso): Many machine learning algorithms, such as linear regression and logistic regression, can be regularized using L1 regularization, which encourages the model to set the coefficients of unimportant features to zero. This effectively performs feature selection as part of the training process.

- Tree-Based Methods: Decision trees and ensemble methods such as Random Forest and Gradient Boosting Trees inherently perform feature selection by giving higher importance scores to features that are more relevant in the tree's splitting decisions.
- Elastic Net: Elastic Net combines L1 (Lasso) and L2 (Ridge) regularization to both penalize feature coefficients and select a subset of features.
- SVM with Recursive Feature Elimination (SVM-RFE): Support Vector Machines (SVM) can be used with an embedded feature selection technique, such as SVM-RFE, which iteratively ranks and removes the least important features while fitting the SVM.

Embedded methods have several advantages, and they are often computationally more efficient than wrapper methods because they eliminate irrelevant features during model training. They can also handle high-dimensional datasets more effectively as they naturally select relevant features and embedded methods can be less prone to overfitting compared to wrapper methods for example (Li et al 2017).

However, there are also disadvantages, as embedded methods are limited to the specific machine learning algorithms they are embedded in. If a particular algorithm does not support feature selection, embedded methods cannot be used. (Bolón-Canedo, Sánchez-Maroño & Alonso-Betanzos 2005) The effectiveness of embedded methods depends on the choice of the algorithm and its hyperparameter and they may not consider feature interactions, which could be important for some tasks. (Pudjihartono et al 2022)

Embedded methods are particularly useful when you have a strong prior belief that a specific machine learning algorithm is well-suited for your problem, and you want to simultaneously train the model and perform feature selection in a more integrated manner.

2.1.3 Filter methods

Filter methods are generally computationally efficient and can serve as a preprocessing step to reduce the dimensionality of the dataset before model training. Next, we will go through the key aspects of filter methods.

Independence of algorithms and classifiers is one of the characteristics and filter methods evaluate the relevance of each feature without considering the interaction between features or the specifics of a machine learning algorithm (Pudjihartono et al 2022). They rely on statistical or information-theoretic measures to score and rank features based on their individual characteristics.

There is a number of different common filter metrics that can be used to evaluate the methods (Venkatesh & Anuradha 2019):

- Correlation: One of the most straightforward methods is to calculate the correlation between each feature and the target variable (for regression problems) or between features (for feature-feature correlation). Features with low correlation to the target or high inter-feature correlation may be considered less important.
- Mutual Information: Mutual information measures the statistical dependency between two random variables. It can be used to evaluate the information gain provided by each feature with respect to the target variable.
- Chi-squared Test: This statistical test is used to determine the association between categorical features and the target variable.
- ANOVA F-statistic: It is employed to assess the variance between groups in a categorical target variable and can be used to select features that contribute significantly to this variance.
- Information Gain, Gini Index, or Entropy: These measures are commonly used in decision tree-based methods to evaluate the quality of features for splitting the data.

Eventually it all comes down to feature ranking, where filter methods typically compute a score for each feature based on one of the metrics mentioned above. Features are then ranked according to their scores, and the top-ranked features are selected for inclusion in the model. The good thing is that filter methods are computationally efficient, making them suitable for high-dimensional datasets and they are model-agnostic, meaning you can use them with any machine learning algorithm. Filter methods provide a quick way to identify potentially irrelevant features, reducing the dimensionality of the dataset (Biswas, Bordoloi & Purkayastha 2016).

However, filter methods may not consider feature interactions, which can be essential for some tasks, and they do not account for the combined effect of multiple features, focusing solely on individual feature relevance. Furthermore, the selected features are not necessarily optimal for a specific machine learning algorithm, which may require a different set of features to perform well (Pudjihartono et al 2022).

Filter methods are often a good starting point for feature selection, especially when you have a large number of features and want to quickly identify a subset that is likely to be relevant. However, they are not a one-size-fits-all solution and should be combined with other feature selection techniques, such as wrapper or embedded methods, for a more comprehensive approach to feature selection, considering feature interactions and the specific requirements of the chosen machine learning algorithm.

Within this research the main focus is on filter methods regarding the single feature selection ranking methods. The first and foremost reason behind this is the simplicity of those methods and the easy implementation of those with the available tools such as MATLAB for example. Furthermore, the computational load of those methods is quite often lighter and the whole process is faster as there is no model training required in order to utilize filter methods.

2.2 Challenges in feature selection

Feature selection is a crucial step in machine learning, as it involves choosing a subset of relevant features from the original set of features to improve model performance. However, it comes with its own set of challenges. Here are some common challenges in feature selection (Subbiah & Chinnappan 2021; Pudjihartono et al 2022; Bolón-Canedo et al 2012; Guyon & Elisseeff 2003; Rong, Gong & Gao 2019; Li et al 2017; Venkatesh & Anuradha 2019):

- Curse of dimensionality: As the number of features increases, the volume of the feature space grows exponentially. This can lead to sparsity of data and increased computational complexity.
- Irrelevant features and correlated features: Identifying irrelevant features can be challenging, especially when dealing with large datasets. Some features may not

contribute meaningful information to the target variable but can still be present in the data. Then again strongly correlated features provide redundant information. Selecting one from a set of correlated features is often sufficient, and choosing the wrong one may negatively impact model performance.

- Non-linearity: The relationship between features and the target variable may be non-linear. Linear methods for feature selection may not capture the complexity of these relationships.
- Interaction effects: Certain features may only contribute meaningfully when considered in combination with others. Traditional feature selection methods might overlook such interaction effects.
- Large datasets: For large datasets, exhaustive search methods become computationally expensive. Efficient algorithms are needed to handle the vast number of possible feature subsets.
- Noise in data: Noisy or irrelevant features in the dataset can mislead feature selection algorithms. Filtering out noise is challenging, and the presence of noise can lead to the selection of suboptimal feature subsets.
- Data imbalance: In imbalanced datasets, where one class is underrepresented, feature selection may prioritize features that discriminate the majority class, neglecting the minority class. This can lead to biased models.
- Inconsistency across models: Different machine learning models may prioritize different features. The challenge is to find a set of features that is robust across various models and not overly tailored to a specific algorithm.
- Dynamic data: In dynamic environments, where data distributions change over time, the relevance of features may also change. Adaptive feature selection methods are required to handle such scenarios.
- Computational complexity: Some feature selection algorithms may be computationally intensive, especially for large datasets. Balancing accuracy with efficiency is crucial, and selecting an appropriate algorithm depends on the specific application and dataset.

Addressing these challenges often involves a combination of domain knowledge, experimentation, and leveraging advanced feature selection algorithms that are suitable for the characteristics of the dataset and the problem at hand.

2.3 Real-world applications regarding feature selection

Feature selection techniques find extensive application across various domains due to their pivotal role in enhancing model performance, reducing computational overhead, and improving interpretability. Some real-world applications of feature selection include:

- Healthcare and Medicine (Panda, Ray & Dash 2020): In medical diagnostics, selecting the most relevant features from patient data (such as symptoms, medical history, and lab results) can assist in building accurate predictive models for diseases. It aids in identifying key biomarkers or risk factors crucial for early disease detection or treatment planning.
- Finance and Risk Management (He, Fataliyew & Wang 2013): Feature selection plays a critical role in financial analysis and risk assessment. Identifying essential indicators from economic data helps in building robust models for stock market prediction, credit scoring, fraud detection, and portfolio management, optimizing decision-making processes in finance.
- Natural Language Processing (NLP) (El-Hajj & Hajj 2022): In NLP, selecting informative features from text data is vital for sentiment analysis, text classification, machine translation, and summarization tasks. Identifying relevant words, phrases, or linguistic patterns enhances the accuracy and efficiency of language-based models.
- Image and Video Processing (Yan, Shen, Liu, Ma, Gao & Sebe 2014): Feature selection is crucial in computer vision applications for image recognition, object detection, and video analysis. Extracting discriminative features from visual data improves model performance and reduces computational complexity in tasks such as facial recognition or autonomous driving.
- Genomics and Bioinformatics (Tadist, Najah, Nikolov, Mrabti & Zahi 2019): In genomics research, feature selection aids in identifying genes or genetic markers associated with specific traits or diseases. It helps in analysing high-dimensional genetic data to uncover meaningful patterns and associations.
- Manufacturing and Industry (Shao, Paynabar, Kim, Jin, Hu, Spicer, Wang & Abell 2013): Feature selection techniques are utilized in predictive maintenance, quality

control, and process optimization in manufacturing. Selecting relevant sensor data or operational parameters assists in identifying potential faults or anomalies in machinery and improving overall efficiency.

Marketing and Customer Analytics (Lessmann & Voß 2009): For customer segmentation, recommendation systems, and market analysis, feature selection helps in identifying influential customer attributes, purchasing behaviour patterns, and demographic factors. This facilitates targeted marketing strategies and personalized recommendations.

Environmental Science and Climate Prediction (Effrosynidis & Arampatzis 2021): Feature selection aids in identifying critical environmental variables or climate indicators from complex datasets. It supports accurate climate modelling, weather forecasting, and ecological analysis.

In these real-world applications and many more, feature selection techniques enable the extraction of information from large and diverse datasets, contributing to the development of accurate, efficient, and interpretable machine learning models tailored to specific domains and tasks.

2.4 Ensemble feature selection

Ensemble feature selection involves combining multiple feature selection techniques or models to enhance the robustness, accuracy, and stability of the selected features. It harnesses the strengths of diverse feature selection methods to mitigate the weaknesses inherent in individual techniques. The end goal is to achieve better results in general by combining multiple models in contrast to just utilizing an individual model and this approach often results in improved performance and more reliable feature subsets for machine learning tasks (Bolón-Canedo & Alonso-Betanzos 2018).

Ensemble feature selection integrates various feature selection techniques or models into a unified framework, and it often involves aggregating the outcomes of individual feature selectors to identify the most relevant features. There are various strategies to implement ensemble feature selection and in order to create ensembles within machine learning in general:

- Voting or Consensus Methods (Brown 2011): In this approach, different feature selection methods or models are applied to the same dataset. Features that are consistently selected or commonly favoured across multiple models are retained, considering them as more reliable or important for predictive performance.
- Sequential Feature Selection (Tsymbal, Pechenizkiy & Cunningham 2004): This technique involves running multiple feature selection methods sequentially, where the output of one method becomes the input for the next. Features selected by each method are combined or voted upon to create a final subset.
- Meta-Learners or Stacking (Kwon, Park & Lee 2019): Utilizing meta-learning models or ensemble techniques such as stacking to combine the outcomes of individual feature selection models. The predictions or importance scores generated by these models are used to create an aggregated feature ranking or selection.
- Bagging or Bootstrap Aggregating (Brown 2011): Employing bootstrapping techniques to generate multiple subsets of the data. Feature selection methods are then applied to each subset, and the resulting feature rankings or selections are combined to form a consensus.

The key advantages of ensemble feature selection include for example improved robustness. (Pes 2019) By leveraging multiple methods, ensemble approaches can reduce the bias introduced by any single feature selection technique or model. Another advantage is enhanced performance, which can occur by combining diverse feature selection strategies. This often leads to more accurate and stable feature subsets, potentially improving the overall performance of machine learning models. (Huang, Liu & Wu 2023) Thirdly ensemble feature selection can aid in selecting features that generalize better across different datasets or domains, reducing overfitting to specific datasets, leading to increased generalization. (Rokach 2009)

However, there are considerations to keep in mind such as computational complexity and the dependency on base models (Bolón-Canedo & Alonso-Betanzos 2018). Ensemble techniques might be computationally more intensive due to the execution of multiple models or methods, potentially increasing the time required for feature selection. Furthermore, the success of ensemble feature selection heavily relies on the diversity and quality of the base feature selection methods or models used within the ensemble.

Naturally, there are also different metrics which can be used to evaluate the performance of the constructed ensemble and the generalization of the ensemble as a whole. In the end these metrics evaluate the performance of the classifier where the ensemble is used. Performance evaluation of ensemble feature selection methods is crucial to assess their effectiveness in selecting relevant features and improving the performance of machine learning models. Several evaluation metrics and techniques can be employed to measure the performance of ensemble feature selection methods (Owusu-Adjei, Hayfron-Acquah, Frimpong & Abdul-Salaam 2023). Such common metrics include accuracy (proportion of correctly classified instances), precision (proportion of true positives among all predicted positives), recall (proportion of true positives among all actual positives), F1 score (harmonic mean of precision and recall) and area under the ROC (receiver operating characteristic) score. These can be all grouped under the category of subset evaluation metrics.

Applying ensemble feature selection requires careful consideration of the dataset, the choice of base feature selection methods, and the ensemble strategy employed. When appropriately implemented, ensemble feature selection can be a powerful approach to improve the reliability and effectiveness of feature selection in machine learning tasks. As seen, ensemble feature selection has applications in various domains, including bioinformatics (Lopez-Rincon et al 2019), image processing (Tuan et al 2020) and predictive modelling (Huang, Liu & Wu 2024). Within machine learning it can be applied to tasks such as classification, regression, clustering, and dimensionality reduction.

Within this research the focus on ensemble feature selection methods is on voting (or election) methods, where the outcomes of the single feature selection methods are combined into a joint outcome. Voting methods in ensemble feature selection are straightforward to implement and thus aggregate the different focus areas of single methods while leveraging the strengths of each method. They combine diverse perspectives of different single methods to form a robust, reliable set of features. This aggregation mitigates the biases and weaknesses of individual methods, enhancing the stability and generalizability of the final feature set. By synthesizing the strengths of different selectors, voting methods help to identify features that consistently contribute to model performance, reduce the risk of overfitting, and provide a comprehensive view of feature importance, ultimately leading to better predictive accuracy and more resilient models. There is a vast amount of different voting aggregation methods to utilize regarding feature selection and within this research the ones chosen were simple methods such as Borda count, arithmetic mean, and Copeland score.

3 Classification

In the vast domain of machine learning, supervised learning serves as the guiding principle for tasks that involve learning from labelled data. It is within this framework that classification finds its roots. Supervised learning entails training a model on a dataset where each example is associated with a specific outcome. In the context of classification, this outcome is a class label, and the model learns to generalize patterns from the labelled training data to make predictions on unseen instances. In this chapter background and theory for classification is introduced as well as the link between feature selection and classification.

3.1 Introduction to classification, classes, labels, and feature space

Before going deeper into the intricacies of classification, it is crucial to understand the distinction between regression and classification in supervised learning. While regression deals with predicting continuous values, such as house prices or temperature, classification focuses on assigning predefined labels or categories to input data. The latter is particularly adept at solving problems where the goal is to categorize data into distinct classes, laying the foundation for numerous real-world applications (Zhou 2021). At the heart of classification lies the concept of classes. These are the distinct categories or groups that the model endeavours to identify within the data. Whether it is discerning between spam and non-spam emails, identifying different species of plants, or classifying images into predefined categories, the classes define the boundaries of the predictive task. Each class is assigned a label, and the goal is to train the model to accurately assign these labels to new, unseen instances based on learned patterns.

The cornerstone of a successful classification model is labelled training data. This dataset consists of examples where each instance is paired with its correct class label. Through exposure to this labelled information, the model learns to recognize patterns and relationships between the input features and the corresponding labels. The process involves adjusting the model's parameters to minimize the difference between predicted and actual labels, laying the groundwork for accurate predictions on new, unseen data.

In the classification journey, features play a pivotal role. These are the attributes or characteristics of each instance in the dataset that the model uses to make predictions. The combination of features forms the feature space, a multi-dimensional space where each point represents a unique instance. Understanding how to select and represent features is essential for the model to discern relevant patterns and make accurate predictions in the complex landscape of real-world data (Li 2024).

3.2 Common classification algorithms

The choice of classification algorithm can significantly impact the performance and outcomes of a predictive task and therefore it is crucial to gain practical insights into selecting the most suitable algorithm for specific scenarios. We now look at various classification algorithms and briefly explore their strengths and weaknesses.

Logistic regression is a foundational classification algorithm that models the probability of an instance belonging to a particular class. Despite its name, logistic regression is used for binary classification tasks, where it predicts the probability of an instance belonging to one class versus another (Zhou 2021). It is particularly well-suited for linearly separable data and serves as a simple yet powerful baseline model for classification tasks. Examples of usage include credit risk analysis (Dumitrescu, Sullivan, Hurlin & Tokpavi 2021) and medical diagnosis. (Li et al, 2022) Logistic regression is easy to implement and interpret, making it suitable for scenarios where model transparency is desired. Transparency in this context is how well the decision-making processes of a model can be understood and interpreted by humans and is therefore crucial, especially in cases where the models are used to make decisions which affect people’s lives for example. It performs well on small to moderately-sized datasets and does not require extensive computational resources. However, the method assumes a linear relationship between input features and the log-odds of the outcome, which may not capture complex relationships in the data.

Decision trees are intuitive models that partition the feature space into a hierarchy of binary decisions, enabling them to make predictions by traversing the tree from root to leaf (de Ville 2013). Random forests, a powerful ensemble method built upon decision trees, aggregate the predictions of multiple decision trees to improve performance and robustness. They excel in handling complex, nonlinear relationships in the data and are resilient to overfitting. Customer segmentation (Han, Lu & Leung 2012), fraud detection (Sahin, Bulkan & Duman 2013), and medical diagnosis (Azar & El-Metwally 2012) are examples where decision trees could be used as their strength is in capturing nonlinear relationships between features and the target variable. Weakness again is that decision trees can overfit to noise in the training data, leading to poor generalization on unseen instances.

Support vector machines (SVMs) are versatile classifiers that seek to find the optimal hyperplane that separates instances belonging to different classes while maximizing the margin between classes (Li 2024). SVMs are effective in high-dimensional spaces and are particularly useful when dealing with linearly separable data or data with complex decision boundaries, for example when dealing with image classification or text categorization. They can also be extended to handle nonlinear relationships through kernel methods. Drawback on SVMs is that they can be computationally intensive, particularly when dealing with large datasets or complex kernel functions.

K-nearest neighbours (k-NN) is a simple yet powerful instance-based learning algorithm that classifies instances based on the majority vote of their nearest neighbours in the feature space (Kataria & Singh 2013). k-NN is non-parametric and lazy-learning, meaning it does not explicitly learn a model during training but rather stores the entire training dataset for inference. It is robust to noisy data and performs well in scenarios where decision boundaries are irregular, such as recommendation systems, anomaly detection, and pattern recognition tasks. The method is robust to noisy data and outliers, as it relies on local information rather than global trends and is also quite easy to understand and implement. One major disadvantage of the method is that the choice of distance metric can significantly impact the performance of k-NN, requiring careful selection based on the characteristics of the data.

Neural networks, particularly deep learning models, have revolutionized the field of classification with their ability to learn complex hierarchical representations from data. Deep neural networks consist of multiple layers of interconnected neurons, allowing them to capture intricate patterns and relationships in the data (Reby, Lek, Dimopoulos, Joachim, Lauga & Aulagnier 1996). They excel in tasks with large volumes of data and are capable of automatic feature extraction, for instance when dealing with image recognition, natural language processing, and speech recognition. Neural networks automatically learn feature representations from data, eliminating the need for manual feature engineering and they scale well with large volumes of data, making them suitable for big data applications. On the other hand, neural networks are complex models with many parameters, requiring significant computational resources and expertise for training and tuning. Considering deep neural networks, they are often regarded as black box models, making it challenging to interpret their decision-making process and they also require large amounts of labelled data for training, which may be difficult to obtain in certain domains.

Selecting the right classification algorithm involves considering several factors, including the nature of the data, the complexity of the problem, computational resources, and interpretability requirements which means that how well the user in question can understand and interpret the algorithm and the model. By carefully evaluating these factors and understanding the characteristics of different algorithms, practitioners can make informed decisions to maximize the effectiveness of their classification models.

3.3 Training and evaluating classification models

There are a number of different steps that one should take before starting to build the actual model for classification.

Data preparation is a crucial step in the machine learning pipeline. It involves cleaning, preprocessing, and transforming raw data into a format suitable for training the model (Foxwell 2020). Data splitting is the process of dividing the dataset into separate training and testing sets to assess the model's performance on unseen data.

Furthermore, training a classification model involves fitting the algorithm to the training data, allowing it to learn patterns and relationships between input features and class labels (Li 2024). The model's parameters are adjusted iteratively to minimize the difference between predicted and actual labels. The performance and generalization ability of a model can be also assessed to cross-validation. It involves splitting the dataset into multiple subsets, training the model on a subset, and evaluating it on the remaining subsets iteratively. Cross-validation provides a more reliable estimate of the model's performance than a single train-test split.

When we move on to evaluation, one could use a confusion matrix, which is a table that summarizes the performance of a classification model by comparing predicted and actual class labels. It provides insights into the model's true positive, true negative, false positive, and false negative predictions. The confusion matrix is a valuable tool for diagnosing model errors and assessing its overall performance (Chen, Dewi, Huang & Caraka 2020).

There are also useful evaluation metrics that quantify the performance of a classification model by comparing its predictions to the actual class labels. Common evaluation metrics include accuracy, precision, recall, and F1 score. (Owusu-Adjei et al 2023) These metrics provide insights into different aspects of the model's performance, such as its ability to make correct predictions and avoid false positives and false negatives – creating a link with the earlier mentioned matrix.

One thing to mention about evaluating the model is the balance of overfitting and underfitting, which again are common challenges in machine learning (Jabbar & Khan 2015). Overfitting occurs when the model learns to capture noise and irrelevant patterns in the training data, leading to poor generalization on unseen data. Underfitting, on the other hand, occurs when the model is too simplistic to capture the underlying patterns in the data. Techniques such as regularization, feature selection, and model complexity adjustment can help mitigate these issues.

3.4 Classification and feature selection

Feature selection plays a crucial role in the effectiveness and efficiency of classification algorithms. It involves choosing the most relevant features from the original set of variables to improve model performance, reduce computational complexity, and enhance interpretability.

By selecting only the most informative features, feature selection helps the classification algorithm focus on relevant information, reducing the risk of overfitting and improving generalization to unseen data. Irrelevant or redundant features can introduce noise and negatively impact model performance (Chen, Dewi, Huang & Caraka 2020).

Including irrelevant features in the model can increase computational overhead and training time. Feature selection helps reduce the dimensionality of the dataset, making the classification task more computationally efficient, especially for algorithms that struggle with high-dimensional data (Tang, Alelyani & Liu 2014). Simplifying the feature space through feature selection can make the classification model more interpretable by identifying the key factors driving the predictions. This facilitates understanding of the underlying relationships between input features and the target variable, aiding in decision-making and model validation.

Some classification algorithms may perform better with a reduced set of features, as they are less prone to overfitting and can capture the essential patterns in the data more effectively. Feature selection allows for tailoring the input feature space to the strengths and limitations of the chosen algorithm. Overall, feature selection is an essential preprocessing step in classification tasks, enabling practitioners to enhance model performance, reduce computational complexity, improve interpretability, and mitigate the challenges associated with high-dimensional data. It empowers classification algorithms to focus on the most relevant aspects of the data, ultimately leading to more accurate and efficient predictive models.

The link between feature selection and classification has been diversely researched previously and for example Punch, Goodman, Pei, Chia-Shun, Hovland, & Enbody (1993) studied how feature selection and classification can be combined by using Genetic Algorithms. The research was conducted so that optimal feature weights were searched and then classification process was optimized, resulting in a significant improvement in the processing time itself. Another extensive study on the different combinations of feature selection methods and classification algorithms was conducted by Spencer, Thabtah, Abdelhamid & Thompson (2020) who also used real life medical data in order to identify the combination that performs the best in terms of accuracy and precision for example.

4 Brief literature review on Ensemble Feature Selection

Although ensemble feature selection was introduced earlier, this section concentrates on the existing literature and research about the topic itself, what is available, where have researchers focused earlier and what is possibly lacking from that research.

Different ensemble methods in general, both applying to feature selection and classification, have been thoroughly reviewed and assessed by Re & Valentini (2012) and more focus was given to supervised methods. Their research identifies several strengths on a more general level for ensemble methods without going too deep on those, making it easy for a beginner to understand what ensembles are all about and why those are used. However, it is not always clear why the ensemble might be more successful compared to a single method for example. They do provide reasons from a practical point of view to utilize ensemble methods.

Robustness has been one of the focus areas where ensemble methods have brought improvement and this was also thoroughly researched by Saeys, Abeel & Van de Peer (2008). This is significantly important as quite often machine learning applications have to deal with new data, and this is exactly where robustness comes into play. Models should not perform well only on the data that they were trained on, but rather also with unseen data and their research concentrated exactly on that. However, robustness was not the only metric that was evaluated but the overall model performance as well. Their results show that in some cases ensemble methods outperform the single methods, but one should carefully consider whether the robustness is the most important factor.

Guan, Yuan, Lee, Najeebullah & Rasel (2014) also have extensively contributed to ensemble-based feature selection and reviewed the topic. They have more focused on stability as a metric and a measure to evaluate the performance and have introduced two different ways to combine ensemble learning and feature selection. Firstly, one could utilize ensemble learning for feature selection and then on the other hand one could utilize feature selection for ensemble learning. These are two different sides of the coin although sound very similar, as they focus on different strategies and goals.

Ensemble feature selection and the link to classification has also been under research and Hoque, Singh & Bhattacharyya (2017) created an ensemble of methods and compared the classification performance against single filter methods. As filter methods evidently include a lot of challenges such as the inconsideration of redundancy of selected features or the accuracy prediction challenges during classification, their work extensively provided valuable results on how the ensemble of methods improves accuracy and therefore tackles the previously mentioned challenges. Their key takeaway was to construct a subset of features which are both relevant and non-redundant and this way overcame the usual problem of ensemble-based feature selection methods where redundant features could have been chosen during feature selection.

When trying to think of practical applications for ensemble feature selection, there are several studies where different methods have been shown to improve the results of classification for example. Borandağ, Özçift and Kaygusuz (2021) did exactly this and researched how ensemble methods could improve the classification performance within text categorization. It was quite specific though given the fact that the methods were tested only on Turkish language texts, but however they were very promising as they generally improved the classification and prediction accuracies when comparing their proposed method against several filter methods.

Another practical application has been introduced by Tasci, Zhuge, Kaur, Camphausen & Krauze (2022) in the medical field. Healthcare in general can benefit from machine learning applications significantly as the diagnosis can be possibly made easier and faster and therefore treatment can also be started earlier. In their study ensemble methods were utilized so that the actual feature selection phase could be improved in terms of performance as a whole, and in the end when their method was compared against a LASSO feature selection method the accuracy rates were higher.

To wrap it all up, Bolón-Canedo and Alonso-Betanzos (2018) have presented a very extensive coverage when it comes to ensemble feature selection as a whole. They do start from the basics and introduce ensemble learning as an idea in itself – focusing not only on ensemble feature selection but other aspects of machine learning as well. The general challenge of single feature selection methods is presented as the choice of method may not be that simple, and this is very ensemble feature selection can be very efficient as the strengths of several methods are combined. In general, they have given a very vast presentation of ensemble feature selection and they have also given especially useful code examples for applications on different tools. They have also introduced several applications from different fields and industries where the methods also differ from each other – one using ranking methods to construct an ensemble and then setting a threshold to choose an appropriate number of features, making it an interesting benchmark for this study as well. Finally, a set of current challenges is introduced which still apply in today's world, especially when companies and organizations need to focus on data even more. The challenges are the same regardless of feature selection or ensemble feature selection, as the number of features seems to grow all the time and also the amount of data itself increases exponentially.

Overall, while ensemble feature selection has shown promise in improving the performance and robustness of feature selection, there are still several areas where research could be expanded to enhance our understanding and utilization of these methods in practical applications.

5 Data

This chapter contains background on the data used in the research as well as the needed steps to ensure the usability of the data. Furthermore, any specific characteristics of the data are described, and emphasis is also given to good data in general and how does one define good data in the end. The data used for the study is described extensively, also including its sources and collection methods for example.

First dataset used for research is about mushrooms. The mushroom dataset is a well-known dataset in the machine learning community, commonly used for classification problems. It contains data on various attributes of different mushroom species and a target variable indicating whether each mushroom is edible or poisonous. The dataset is sourced from the UCI Machine Learning Repository (UCI, 1987) and is widely used for machine learning research and educational purposes. The dataset comprises of 8,124 instances, each representing a different mushroom and there are 23 features in the dataset. They describe various characteristics of the mushrooms such as cap shape, cap surface, cap colour, bruises, odour, gill attachment, gill spacing, gill size, gill colour, stalk shape, stalk root, stalk surface above ring, stalk surface below ring, stalk colour above ring, stalk colour below ring, veil type, veil colour, ring number, ring type, spore print colour, population, and habitat. All features are categorical.

The target variable is a binary classification indicating whether the mushroom is edible (e) or poisonous (p). All features and the target variable are encoded as categorical values, with each feature having multiple possible categories. For example, cap shape can be bell (b), conical (c), convex (x), flat (f), knobbed (k), or sunken (s). There is one feature (stalk root) that contains missing values, which needs to be handled through data imputation or removal. Given the categorical nature of the data, one of the key challenges in using this dataset for machine learning tasks is handling and encoding categorical features effectively. The mushroom dataset is often used to benchmark classification algorithms and assess their performance in handling categorical data.

Due to its simplicity and clear binary classification problem, it serves as a useful dataset for exploring various machine learning concepts, including data preprocessing, feature engineering, model training, and evaluation. Overall, the mushroom dataset provides a straightforward yet effective platform for studying and experimenting with classification tasks in machine learning.

The second dataset is financial and is called the Taiwanese Bankruptcy Prediction dataset. It is a dataset commonly used for research in finance, machine learning, and risk assessment, particularly for predicting bankruptcy in companies based in Taiwan. It contains data on various financial attributes of Taiwanese firms, with a binary classification task indicating whether a company will go bankrupt or not.

The dataset is based on data from Taiwanese firms and is sourced from the UCI Machine Learning Repository (UCI, 2020), originally collected between 1999 to 2009 from the Taiwan Economic Journal. Dataset consists of 6,819 instances, each representing a different Taiwanese firm and each instance has 96 different features, one naturally being the target variable telling us whether the company went bankrupt or not within a specific time frame. The features mostly are different financial indicators such as current ratio, quick ratio, debt ratio, and other liquidity and leverage ratios. There are some profitability ratios such as return on assets (ROA) and return on equity (ROE) and also operational metrics such as sales, inventory, and expenses. The whole dataset consists of numerical attributes. Overall, the dataset contains a very comprehensive list of different financial metrics and measures which provide a very thorough picture of a company’s financial health and operational efficiency as a whole. At the same time as the number of features is remarkably high, one can explore the relationships between different metrics easily.

Working with the dataset involves dealing with a large number of features and potential correlations among them. Feature engineering and selection are essential to handle the high dimensionality and improve model performance, but on the other hand the high dimensionality and focus on financial data and attributes makes the dataset and interesting resource to work with. Although it is only focused on Taiwanese companies and their data, it could be used as a benchmark to identify factors that contribute to corporate bankruptcy. Understanding the factors that contribute to corporate bankruptcy can help policymakers, financial institutions, and businesses develop strategies to mitigate risk and support financial stability.

Third dataset is called The Insurance Company Benchmark (COIL 2000), which is a dataset commonly used in data mining and machine learning research for classification tasks. It represents data from an insurance company and includes information about different types of insurance claims and policies. The dataset is sourced from the *UCI Machine Learning Repository* and is known for being a challenging benchmark for classification algorithms due to its complexity. A Dutch datamining company called Sentient Machine Research owns and supplies the dataset and it is based on real world business data.

This dataset consists of 9,822 instances, each representing a different policyholder's claim. There are 85 attributes (or features) in the dataset, which can be categorized into various types:

- Numerical attributes: There are 24 continuous attributes, such as age, income, and expenses.
- Categorical attributes: There are 61 categorical attributes, such as policy type, policyholder's profession, and area of residence.

The target variable represents the risk category of each policyholder's claim. It is a binary classification indicating whether the claim is high risk (1) or low risk (0). Dataset contains a mix of numerical and categorical data. Numerical features include age, income, and expenses, while categorical features include policy type, residence area, and policyholder's profession. Given the mixture of numerical and categorical attributes, handling and encoding categorical data effectively can be challenging. Additionally, the high dimensionality of the dataset (85 attributes) may present computational and interpretability challenges.

The Insurance Company Benchmark (COIL 2000) dataset is often used to benchmark and assess classification algorithms, particularly those that handle a mix of numerical and categorical data. It serves as a useful resource for studying risk assessment for example, when one needs to understand the factors that contribute to high-risk insurance claims and develop models to predict them. It could also be used to identify patterns indicative of fraudulent claims and explore how different customer characteristics influence claim risk and other insurance-related outcomes. Furthermore, as the dataset provides a wealth of information about policyholders, it allows researchers to segment customers into different risk categories and tailor insurance products and services accordingly. Companies could use it as a benchmark to make informed decisions about policy approvals and pricing.

The fourth dataset used is the Breast Cancer Wisconsin dataset (Wolberg et al, 1995), which is used to predict whether a breast cancer tumour is malignant (cancerous) or benign (non-cancerous) based on various characteristics of the cell nuclei present in the breast mass. The dataset contains 569 observations, where each observation corresponds to a different breast mass. Each observation includes 30 numeric features which are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. These features represent different characteristics of the cell nuclei present in the image. Additionally, there is a target attribute (diagnosis), which indicates whether the tumour is malignant or benign.

Although the dataset is fairly straightforward, it can still be seen as a rich dataset in the sense that the number of features is fairly high. As it is based on real-world data, it is highly relevant, and the comprehensiveness of the dataset makes it an interesting base to build models on. Furthermore, the dataset is quite well balanced regarding the classes, which again is helpful in creating classification models for example. Even though the number of features is quite high, the number of observations in the end is relatively low, which limits the ability to train very complex models and might not capture the full variability present in larger populations. Also, some features seem to correlate highly between each other, resulting in removing the features even before the actual feature selection phase.

As a summary these datasets are quite diverse in the number of features but also in the type of data they include. All of the datasets need some pre-processing in terms of checking for missing values within the features but also transforming the features to a numerical form for example in order to be easily utilized for feature selection and classification.

6 Methodology

This chapter describes the methods used and also the reasoning why those specific methods were chosen – more specifically the feature selection methods and the classification methods that were used. Background to methodology is given and also the strengths of those methods as well as weaknesses compared to others are evaluated within this context.

Before starting the actual work with the data, it needs to be split into training and testing sets. Cross-validation is a common technique used to evaluate the generalization performance of ensemble feature selection methods (Mera-Gaona, López & Vargas-Canas 2021). It involves splitting the dataset into training and testing sets multiple times and averaging the performance metrics across different splits. Common cross-validation techniques include k-fold cross-validation and leave-one-out cross-validation. While splitting, it was ensured that the class label balance was maintained, meaning that both sets contain a representative distribution of the actual target variable. The sample sets were generated using stratified random sampling where the generated sample sets contain the correct ratios of each class.

Furthermore, once the data was split to testing and training sets, both sets were then split into labels (dependent variable) and features (independent variable). This makes it easier to work with the data later on was labels were used as the target outputs while the features were then used to actually train the models.

In order to effectively use the data for feature selection and classification, it needs to be ensured that the classes (or labels) are balanced. Especially with the Taiwan bankruptcy dataset the original class imbalance was significant, when almost 97% of the observations belonged to one class and only 3% to another. Class balancing was carried out by using the SMOTE (Synthetic Minority Over-sampling Technique) method. Class imbalance occurs when the number of instances in one class is significantly higher than the number of instances in another class. This can cause problems for machine learning models, which may become biased towards the majority class, leading to poor performance in predicting the minority class.

SMOTE aims to balance the class distribution by generating synthetic examples of the minority class. First minority class instances need to be identified and then the nearest neighbours need to be selected. For each minority class sample, a synthetic sample is created by interpolating between the original sample and the selected neighbour. The method itself is not directly available in MATLAB but a user-generated, peer-reviewed code and function exists that is very usable for this case. The function itself needs some parameters which were given so that the classes were more balanced in the end.

Ensemble feature selection methods should then be compared against baseline methods or individual feature selection techniques to demonstrate their superiority (Khoder & Dornaika 2021). Baseline methods may include univariate feature selection methods (e.g., correlation-based selection) or wrapper methods (e.g., recursive feature elimination). This comparison was concluded by utilizing the feature subsets generated by each method and using those subsets within classification, and then the classification performance was actually compared between different feature subsets.

6.1 Feature selection phase

The initial feature selection was conducted using ranking methods, which aim to organize and rank the features based on their importance or relevance within the given dataset. Ranking methods in general are good for prioritizing features for further analysis or model building. They also help to reduce dimensionality and potentially improve model interpretability and generalization performance.

6.1.1 Pearsons's correlation

First of the ranking methods used was Pearson correlation, sometimes also noted as Pearson correlation coefficient or Pearson's product moment correlation coefficient. It measures the strength and direction of the linear relationship between two continuous variables and when applied in the context of feature selection, Pearson correlation is used to assess the linear correlation between each feature and the target variable. The formula for calculating the Pearson correlation coefficient between two variables X and Y is:

$$r = \frac{\Sigma(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\Sigma(X_i - \bar{X})^2 \Sigma(Y_i - \bar{Y})^2}} \quad (1)$$

Where:

- r is the Pearson correlation coefficient.
- X_i and Y_i are individual values from the two variables.
- $\bar{X}$ and $\bar{Y}$ are the means (averages) of the variables X and Y, respectively.
- $\sum$ denotes the sum across all observations.

To calculate Pearson's r:

1. Subtract the mean of variable X from each individual value in X, and the mean of variable Y from each individual value in Y.
2. Multiply the deviations from the means (from step 1) for each pair of values and sum them all up.
3. Calculate the sum of the squared deviations from the mean for each variable.
4. Take the square root of the product of the sums calculated in step 3.
5. Divide the sum from step 2 by the square root of the product from step 4.

The result then is the correlation coefficient, and it can range from -1 to 1. A coefficient close to 1 indicates a strong positive linear correlation, meaning that as the feature increases, the target variable also tends to increase and vice versa, as the coefficients is close to -1 indicates a strong negative linear correlation, meaning that as the feature increases, the target variable tends to decrease. A coefficient close to 0 indicates a weak correlation or that there is no correlation at all between the target variable and the feature.

All features in the datasets were assigned a correlation coefficient based on the formula above and then were ranked based on the absolute value of their correlation coefficient. Features with higher absolute values are considered more relevant for predicting the target variable. Once the features are ranked, a threshold could be set to maintain only the features that have correlation above a certain value, or a specific number of features could be selected. This is what was done, so only a number of the features that had the highest correlation values was chosen for further analysis and model building. It was also tested with different numbers depending on the dataset to see how the number of features affects the model performance.

Although Pearson correlation is a straightforward and computationally efficient method for feature selection, it has some limitations such as that only captures linear relationships between features and the target variable, so it may miss nonlinear associations. It can also be sensitive to outliers in the data, which can skew the results and the method itself assumes normally distributed data, which may not always be the case in real-world datasets.

6.1.2 Minimum Redundancy Maximum Relevance

Second feature selection method used was a popular algorithm called Minimum Redundancy Maximum Relevance (mRMR). (Ding & Peng, 2005) It aims to select a subset of features that are both maximally relevant to the target variable and minimally redundant with each other. The algorithm's goal is to improve model performance and interpretability by reducing the feature space while retaining the most informative and unique features.

Relevance is typically measured using mutual information between the feature and the target variable. Mutual information quantifies the amount of information one random variable contains about another. Then again redundancy is measured using mutual information between pairs of features, which quantifies the degree of dependence between them. The mRMR algorithm seeks to maximize the relevance of the selected features to the target variable and the minimizes the redundancy of the selected features.

Both relevance and redundancy are defined in conjunction with mutual information. Mutual information equation can be expressed as follows, where Z and X are discrete random variables:

$$I(X,Z) = \sum_{i,j} P(X = x_i, Z = z_j) \log \frac{P(X = x_i, Z = z_j)}{P(X = x_i)P(Z = z_j)} \tag{2}$$

Where:

- I is the mutual information of discrete random variables X and Z

- P denotes the probability that X or Z would take particular values x_i or z_j

This definition is used by the *fscmrmr* function in MATLAB, and it works for continuous and categorical variables. Then again, the mutual information is used to define maximum relevance and minimum redundancy. The equation for mmaximising the relevance of the set of features:

$$V_S = \frac{1}{|S|} \sum_{x \in S} I(x, y) \tag{3}$$

Minimizing the redundancy of the set of features:

$$W_S = \frac{1}{|S|^2} \sum_{x,z \in S} I(x, z) \tag{4}$$

Where:

- S is the subset of selected features.
- $|S|$ is the number of features in the subset S.

Algorithm steps (Ding & Peng, 2005):

1. The feature that has the highest relevance to the target variable, denoted as $\max_{\in\Omega} V_x$, is selected from the entire feature space Ω and added to the set S.
2. The complement of S, denoted as S^c, is examined for features that exhibit non-zero relevance and zero redundancy with respect to S.
   - 2.1: If S^c contains at least one feature with non-zero relevance and zero redundancy, the feature with the highest relevance ($\max_{x \in S_c, W_x=0} V_x$) is selected and added to the set S.
   - 2.2: This process is repeated until all features in S_c have some level of redundancy with respect to S.
3. For the remaining features in S_c, select the feature with the highest mutual information quotient (MIQ) value, calculated as follows:

$$\max_{\mathrm{x} \in S^c} \mathrm{MIQ}_{\mathrm{x}} = \max_{\mathrm{x} \in S^c} \left(\frac{I(x,y)}{\frac{1}{|S|} \sum_{z \in S} I(x,z)} \right. \tag{5}$$

   - Where I is the mutual information of x and y and $|S|$ is the number of features in the subset S.

- The feature with the highest MIQ value is added to the set *S*.

4. The previous step is repeated until all features in S_c have zero relevance to the target variable.
5. Add any remaining features from S_c that have zero relevance to the set *S* in a random order.

By selecting features that are most relevant to the target variable, mRMR can improve the accuracy and robustness of the model. The algorithm also helps reduce the feature space, making models more efficient and easier to interpret. Again, also with mRMR similarly to correlation, the features were ranked with the help of the algorithm and a chosen number of features was selected for further analysis and model building.

6.1.3 Analysis of Variance (ANOVA)

Third feature selection method and technique used was Analysis of Variance (ANOVA). It is a powerful statistical technique used to determine whether there are any statistically significant differences between the means of three or more independent groups. In the context of feature ranking for machine learning, ANOVA is employed to assess the importance of each feature by analysing how well it can differentiate between the classes of the target variable.

There are different types of ANOVA, such as one-way, two-way, and repeated measures. Two-way can be used when comparing means with two independent variables (factors) and it can also assess interaction effects between factors. Repeated measures again should be used when the same subjects are used for each treatment (e.g., in a longitudinal study). Within this research one-way ANOVA was used, which can be used when comparing the means of three or more independent (unrelated) groups.

Conducting the ANOVA analysis requires a formulation of hypotheses:

- **Null Hypothesis (H0)**: There is no significant difference in the means of the target variable across different levels of the feature.
- **Alternative Hypothesis (H1)**: There is a significant difference in the means of the target variable across different levels of the feature.

ANOVA also relies on several assumptions to ensure the validity of its results, and the validity of these assumptions should be checked before conducting feature selection. Violations of these assumptions can affect the accuracy and reliability of evaluation tests.

Here are the key assumptions (Rutherford, 2001):

- **Independence**: Observations within each group and between groups should be independent. In other words, the values of one observation should not be influenced by the values of other observations.
- **Normality**: The data within each group should follow a normal distribution. While ANOVA is considered robust to violations of normality, particularly with larger sample sizes, extreme departures from normality can affect the accuracy of results.
- **Homogeneity of Variances (Homoscedasticity)**: The variances of the groups should be approximately equal. This means that the spread of data points around the group means should be consistent across all groups. Violations of this assumption can lead to biased estimates of group variances and affect the validity of the F-test.
- **Independence of Errors**: The errors (residuals) should be independent of each other. This assumption implies that there should be no systematic pattern or correlation in the residuals. Violations of this assumption can lead to biased standard errors and affect the precision of estimated effects.

One-way ANOVA was performed for each feature individually and the function in MATLAB calculates the *F*-statistic and associated *p*-value for each feature. The *F*-statistic measures the ratio of variance between groups to variance within groups and a higher *F*-statistic indicates a higher difference in means between groups. The *p*-value indicates the significance of the *F*-statistic. A low *p*-value (typically below a chosen significance level, e.g., 0.05) suggests that the feature is significant in explaining the variance in the target variable.

The *F*-statistic formula is simply the ratio of *between-group variance* and *within-group variance.* With this context the groups are regarded as classes. The between-group variance represents the variability between the group means. It is calculated by taking the variance of the group means from the overall mean.

$$Between - group\ variance\ = \frac{\sum_{i=1}^{k} n_i \times (\bar{X}_i - \bar{X})^2}{k-1} \tag{6}$$

Where:

- k is the number of groups
- n_i is the number of observations in the ith group
- $\bar{\boldsymbol{X}}_{\boldsymbol{i}}$ is the mean of the ith group
- $\bar{X}$ is the overall mean

The within-group variance represents the variability within each group. It is calculated by taking the average of the variances within each group, weighted by the degrees of freedom.

$$Within - group\ variance\ = \frac{\sum_{i=1}^{k} (n_i - 1) \times s_i^2}{N-k} \tag{7}$$

Where:

- k is the number of groups
- n_i is the number of observations in the ith group
- s_i^2 is the sample variance of the ith group
- N is the total number of observations

F-statistic essentially measures the ratio of the variance explained by the differences between group means to the variance explained by individual differences within each group.

$$F = \frac{Between{-}group\ variance}{Within{-}group\ variance} \tag{8}$$

A higher F-statistic indicates a greater difference between group means relative to the variability within groups, suggesting that the feature may be more influential in explaining the variability in the target variable.

All the features in the datasets were evaluated by ranking the features based on the ANOVA method and the features with the highest rankings were chosen for further evaluation. Different numbers of top features were tested in order to find out the optimal number.

6.1.4 Chi-square

Lastly feature selection was executed with univariate feature ranking by using chi-square tests and in MATLAB this can be done with function *fscchi2*. The method is a feature selection technique based on the chi-square statistic and it evaluates the dependency between each feature and the class labels (target variable) in a classification problem. This method is particularly useful for categorical feature variables and categorical target variables.

Even though the chi-square test is typically used to assess the association between two categorical variables, such as a feature and a categorical target variable, the function in MATLAB can still be applied when the target variable is non-categorical, as long as it represents distinct groups or classes that the features can distinguish between. The method might be less effective for continuous or interval variables due to the assumptions of the chi-square test.

6.2 Ensemble phase

Once the individual feature selection ranking methods were executed and the rankings for all the features were calculated, it was time to move on to the ensemble phase and combine the individual rankings of features to a joint set. As said, voting methods were used as aggregation methods and will be described later, and aggregating here can be understood as the process of combining the feature rankings from multiple individual methods.

However, ensemble methods in general can be categorized to parallel ensemble methods, sequential ensemble methods and hybrid ensemble methods. Voting falls under the category of parallel ensemble methods, where individual methods (base learners) can be trained and generated at the same time and the methods are not dependent on each other. The advantage here is that possible errors that a method can make might be different from those that another method makes, and therefore the effect of errors is balanced. (Mohammed & Kora 2023)

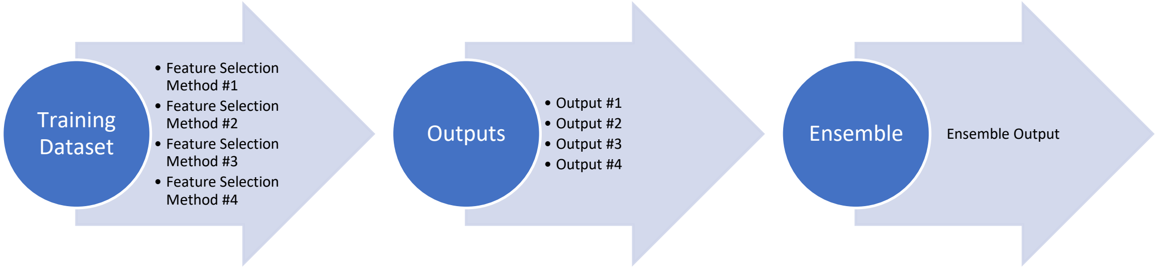

Figure 2. Ensemble Feature Selection process

In Figure 2 a simple workflow of generating an ensemble is depicted and it can also be seen as a framework for parallel ensembles, as the training happens at the same time regardless of the method. The actual individual methods chosen for this research were the 4 methods (Pearson, MRMR, ANOVA, Chi-Square) described earlier, and in theory one could add even more individual methods to the list where the final ensemble is then created from.

6.2.1 Borda count

When moving on to the ensemble phase, it was first decided that the ensemble would be constructed with the Borda count method. (Drotár, Gazda & Gazda 2017) Borda count is a ranked voting system where each classifier's rank is determined based on its performance on the training data. As the earlier feature selection methods ranked the features based on their performance, it is relatively easy to continue with the Borda method as each feature is assigned a weight based on its rank. For example, the feature with rank 1 gets weight (number of features - 1), the feature with rank 2 gets weight (number of features - 2), and so on. This is the essence of Borda count and the methodology provides a systematic way to aggregate predictions from multiple classifiers, giving more weight to the ones that perform better on the validation dataset. This can lead to improved performance and robustness of the ensemble classifier, especially when the base classifiers are diverse and complementary in their predictions.

Once the features were assigned weights based on the Borda method, the classification was executed with different numbers of top features in order to evaluate how well the feature ensemble performs.

6.2.2 Arithmetic mean

Another aggregation method to combine the feature rankings of several feature selection methods to a single ranking was to calculate the arithmetic mean of the rankings for each feature. It is a powerful yet simple technique in ensemble learning, providing an effective way to combine the strengths of multiple models to achieve better predictive performance. (Noureldien & Mahmoud 2021) The method specifically combines the predictions of these base models by averaging them, and as the rankings from different feature selection methods were already produced, it is simple to calculate the arithmetic mean for each feature from those.

For each feature, the arithmetic mean of its ranks across all models was calculated. The formula for the mean rank of item *i* from *M* models is:

$$Mean\ Rank\ (i) = \frac{1}{M} \sum_{m=1}^{M} Rank_m(i) \qquad (9)$$

Where $\mathrm{Rank}_m(i)$ is the rank of item *I* given by model *M.*

Once all the features were assigned a rank the final combined ranking was generated and used with the classification method.

6.2.3 Copeland method

A third aggregation method to combine the rankings of single feature selection methods was the Copeland method. (Al-Sharrah, 2010) It is a voting system used to combine the rankings of different voters to determine an overall winner in an election or a decision-making process. It is particularly useful in settings where there are multiple candidates or options and voters rank them in order of preference.

Pairwise comparison is the key of the method and each pair of features is first compared against each other in a way where the feature selection methods work as voters and the features are then the candidates. The number of voters that prefer one feature over another are counted and the feature that is most preferred wins the comparison. For each feature, the number of times they win a comparison needs to be counted. Then the Copeland score is:

$$Copeland\ Score\ =\ (Number\ of\ wins) - (Number\ of\ losses) \tag{10}$$

The features (candidates) are then ranked based on the score so that the one with the highest score is the winner. Again, this final ranking was then used with the chosen classification method.

6.3 Classification phase

The k-Nearest Neighbours (kNN) algorithm is a simple and intuitive classification algorithm used for both classification and regression tasks. The kNN algorithm is based on the principle of similarity. It assumes that similar data points tend to belong to the same class. Given a new, unseen data point, the kNN algorithm identifies the k nearest data points to the new point (nearest neighbours) based on a distance metric (typically Euclidean distance).

The class of the new data point is then determined by a majority vote among its k nearest neighbours. In other words, the class label of the new data point is determined by the class label that occurs most frequently among its k nearest neighbours. This is in a way similar to a voting process, where each neighbour "votes" for its class label, and the class with the most votes wins.

Here are the simplified steps of the algorithm:

- **Initialization**:
    - Choose the number of neighbours (k).

- Select a distance metric, the most commonly used distance metric is Euclidean distance, which is calculated as:

$$Euclidean\ distance = \sqrt{\sum_{i=1}^{m}(x_i - y_i)^2} \quad (11)$$

where x_i and y_i are the ith features of the two data points.

- **Training**:
    - The kNN algorithm does not involve explicit training. Instead, it stores all available data points and their associated class labels.
- **Prediction**:
    - Given a new data point to be classified, compute its distance to all other data points in the training set using the chosen distance metric.
    - After computing the distances, the next step is to identify the k nearest neighbours of the new data point. This can be done by sorting the distances in ascending order and selecting the k smallest distances.
    - Determine the class label of the new data point based on a majority vote.
- **Evaluation**:
    - Evaluate the performance of the kNN classifier using appropriate evaluation metrics such as accuracy, precision, recall, F1-score, or area under the receiver operating characteristic curve (ROC-AUC).

While the algorithm itself does not have a single overarching formula, these steps encapsulate the main calculations involved in the kNN algorithm. The simplicity and flexibility of the algorithm make it easy to understand and implement, but its performance can vary depending on factors such as the choice of *k*, the distance metric, and the characteristics of the dataset.

7 Results

Within this chapter the results of different feature selection methods and the ensemble feature selection method are described when it comes to different classification metrics. This tells us how well the feature selection phase has succeeded in selecting the most optimal set of features and comparison of the utilized feature selection methods against the ensemble feature selection method is also described and evaluated.

As the classification model is executed, predictions of the classes are generated at the same time. By generating a confusion matrix out of predicted classes as well as the true classes one can see the number of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). With the help of the matrix one can assess the performance of the classifiers by using the following metrics:

- **Accuracy**: The proportion of correct predictions, meaning that the total number of correct predictions is compared against the total number of predictions

  $$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{12}$$

- **Precision**: The proportion of true positives among all positive predictions, and here true positive means that the model correctly predicts the positive class. All positive predictions can also include false positive predictions, meaning that the model incorrectly predicted the positive class.

  $$Precision = \frac{TP}{TP+FP} \tag{13}$$

- **Recall**: The proportion of true positives among all actual positives. The model can predict false negatives, which are actually positives – these false negatives are then summed up with the true positives giving us the number of actual positives.

  $$Recall = \frac{TP}{TP+FN} \tag{14}$$

- **F1-score**: The harmonic mean of precision and recall. F1-score combines those two important metrics and the larger the score, the better the model is in terms of precision and recall. To calculate the score, one needs to first calculate precision and recall, and then:

$$F1 - score \; = \frac{2*Precision*Recall}{Precision+Recall} \tag{15}$$

- **ROC-AUC**: The Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (AUC) are the two things here, and the ROC curve is a graphical representation of a classifier's performance across various threshold settings and basically plots the true positive rate against the false positive rate. The AUC again is a single value that summarizes the overall performance of the classifier. It represents the area under the ROC curve and is used to compare different models.

It is important to look at all these scores and metrics together and assess how well the classifier performs as a whole, and not just concentrate on accuracy for example. Classifier performance in the end also depends significantly on the dataset and the feature selection method, meaning that one should carefully evaluate whether the chosen methods are best for the task at hand.

Regarding pre-processing of the data and the steps to take before conducting feature selection, the datasets were checked for multicollinearity in order to see whether the features correlate strongly with each other. In case there was a very high correlation, the feature was then removed from the dataset. Furthermore, all the datasets were cross-validated by splitting the dataset to training and testing data with k-fold cross-validation. The number of folds for all datasets was 10 and within cross-validation also stratifying was ensured, meaning that the class balance was preserved amongst the cross-validation datasets.

7.1 Mushroom dataset

As the nature of the dataset is quite simple and straightforward, it was perhaps also expected that all the methods would perform quite well in terms of classification metrics. The focus was however on feature selection phase and the classification phase executed with the same parameters regardless of the feature selection method. For this dataset, the classes were already balanced, so no oversampling was needed, and the actual values of the features were on the same range so also no normalization was needed. However, one feature contained only a single value, so it was removed from the dataset.

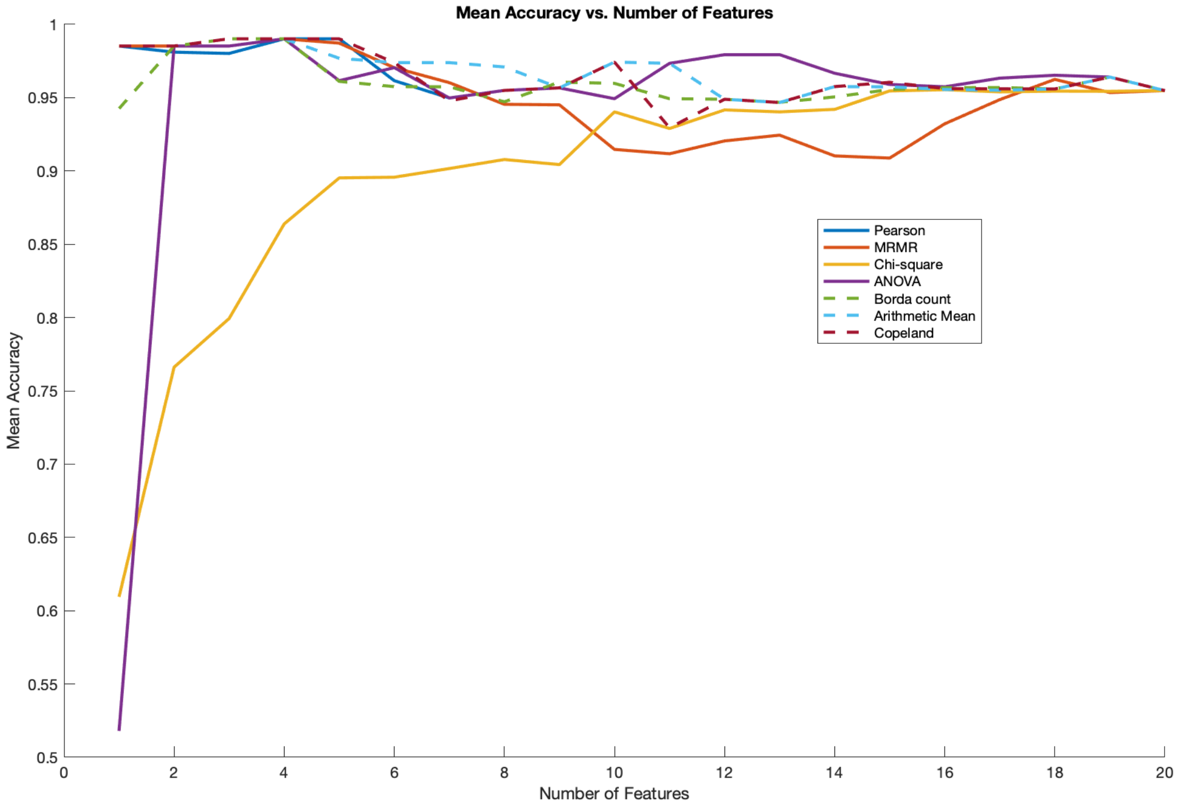

Figure 3. Mean classification accuracy of Mushroom dataset with different numbers of features broken down to each feature selection method.

From Figure 3 we can see that while the number of features increases, so does the classification accuracy of all the feature selection methods utilized. ANOVA does not perform so well with a small number of features and neither does Chi-square which requires more features to match the performance of other methods. Within this dataset, it cannot be said that the Borda count ensemble method would significantly outperform the single ranking methods when it comes to accuracy. When evaluating other metrics such as F1-score and precision, the difference is still not that clear, but what can be said that the ensemble method perhaps stabilizes quicker when features are added to the set.

Table 1. Classification accuracy by FS method with different numbers of features selected from the Mushroom dataset.

Number of features / Method	Pearson	MRMR	Chi-square	ANOVA	Borda	Arithmetic Mean	Copeland
1	0.9852	0.9852	0.6094	0.5180	0.9425	0.9852	0.9852
2	0.9810	0.9852	0.7661	0.9852	0.9852	0.9852	0.9852
3	0.9802	0.9852	0.7993	0.9852	0.9902	0.9902	0.9902
4	0.9902	0.9902	0.8639	0.9902	0.9902	0.9902	0.9902
5	0.9902	0.9872	0.8953	0.9616	0.9611	0.9769	0.9902
6	0.9616	0.9705	0.8957	0.9705	0.9575	0.9739	0.9739
7	0.9498	0.9602	0.9017	0.9498	0.9574	0.9739	0.9478
8	0.9549	0.9455	0.9078	0.9549	0.9472	0.9709	0.9549
9	0.9567	0.9451	0.9044	0.9567	0.9606	0.9567	0.9567
10	0.9493	0.9147	0.9403	0.9493	0.9598	0.9743	0.9743
11	0.9734	0.9117	0.9290	0.9734	0.9493	0.9734	0.9295
12	0.9793	0.9205	0.9417	0.9793	0.9489	0.9489	0.9489
13	0.9793	0.9244	0.9403	0.9793	0.9468	0.9467	0.9467
14	0.9666	0.9103	0.9420	0.9666	0.9505	0.9575	0.9575
15	0.9590	0.9088	0.9546	0.9590	0.9556	0.9574	0.9605
16	0.9574	0.9322	0.9554	0.9574	0.9568	0.9556	0.9562
17	0.9633	0.9487	0.9540	0.9633	0.9569	0.9551	0.9561
18	0.9653	0.9625	0.9545	0.9653	0.9559	0.9559	0.9559
19	0.9641	0.9535	0.9543	0.9641	0.9641	0.9641	0.9641
20	0.9548	0.9548	0.9548	0.9548	0.9548	0.9548	0.9548

Although the accuracy was quite high even with a small number of features, Table 1 depicts the classification accuracy with different numbers of features and with different feature selection methods. increasing the number of features above 5, all the methods except Chi-square start to present a decrease in accuracy, although only a very slight one.

As can be seen from Figure 4, the F1 score is remarkably high with all the different methods and even with a small number of features. Together with a high accuracy and other metrics it can be stated that the model identifies relevant instances and minimizes the false positives very well.

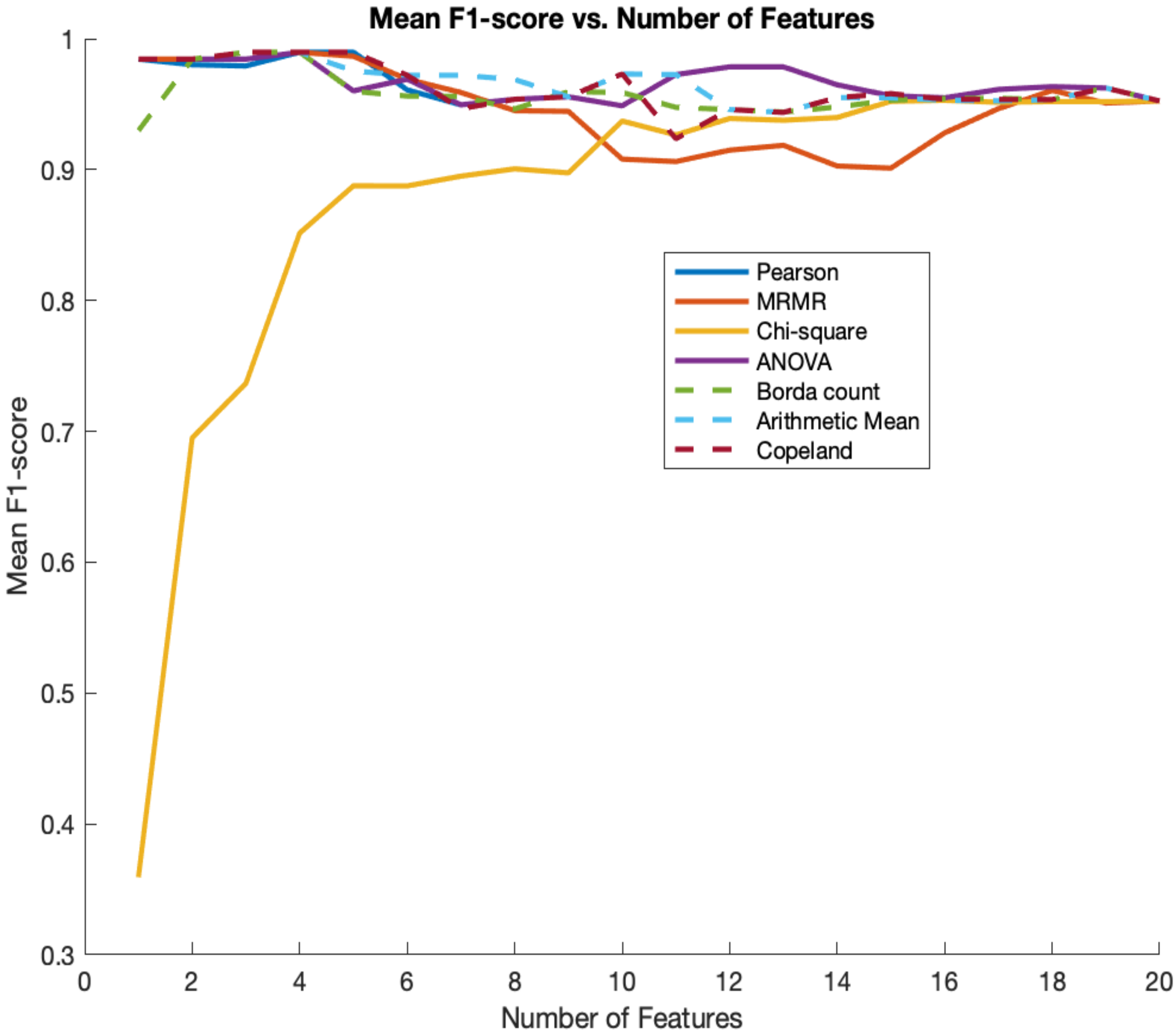

Figure 4. Mean F1 score of Mushroom dataset with different numbers of features broken down to each feature selection method.

It seems that the conclusion with the mushroom dataset especially is that the ensemble methods do not outperform the single ranking methods. The explanation to this might be that the dataset itself is pretty straightforward and that there are some features that have a significant impact on the target variable. The same results seem to happen with the other two ensemble methods, which can be also verified from Table 2 when studying the impact of the number of features to the F1-score. Although evaluation metrics might not be that much better compared to single ranking methods with ensembles, it somehow seems that the ensemble methods achieve stability much faster when adding new features.

Table 2. Classification F1-score by FS method with different numbers of features selected from the Mushroom dataset.

Number of features / Method	Pearson	MRMR	Chi-square	ANOVA	Borda	Arithmetic Mean	Copeland
1	0.9844	0.9844	0.3593	0.9821	0.9298	0.9844	0.9844
2	0.9801	0.9844	0.6951	0.9844	0.9844	0.9844	0.9844
3	0.9792	0.9844	0.7367	0.9844	0.9897	0.9897	0.9897
4	0.9897	0.9897	0.8515	0.9897	0.9897	0.9897	0.9897
5	0.9897	0.9866	0.8876	0.9602	0.9598	0.9753	0.9897
6	0.9609	0.9691	0.8875	0.9691	0.9562	0.9721	0.9721
7	0.9495	0.9590	0.8950	0.9495	0.9560	0.9721	0.9465
8	0.9538	0.9451	0.9006	0.9538	0.9463	0.9688	0.9538
9	0.9556	0.9445	0.8975	0.9556	0.9596	0.9556	0.9556
10	0.9488	0.9080	0.9371	0.9488	0.9589	0.9731	0.9731
11	0.9726	0.9062	0.9265	0.9726	0.9476	0.9726	0.9235
12	0.9785	0.9149	0.9391	0.9785	0.9459	0.9459	0.9459
13	0.9785	0.9185	0.9377	0.9785	0.9440	0.9437	0.9437
14	0.9648	0.9028	0.9398	0.9648	0.9479	0.9551	0.9551
15	0.9565	0.9011	0.9522	0.9565	0.9531	0.9549	0.9581
16	0.9549	0.9281	0.9530	0.9549	0.9543	0.9530	0.9537
17	0.9613	0.9465	0.9514	0.9613	0.9545	0.9525	0.9536
18	0.9634	0.9605	0.9520	0.9634	0.9535	0.9535	0.9535
19	0.9624	0.9509	0.9519	0.9624	0.9624	0.9624	0.9624
20	0.9525	0.9525	0.9525	0.9525	0.9525	0.9525	0.9525

Table 2 presents the mean F1-score for different methods and a different number of features, and it follows the same kind of pattern as accuracy, so for all the other methods except Chi-square the highest score is obtained with a relatively small number of features but with Chi-square all the features are needed to obtain the highest score. As an overall result with this specific dataset, it can be stated that single ranking methods provide comparable or even outperforming results when compared to the ensemble methods chosen. However, as the ensembles were also straightforward to implement and execute, there is no actual downside to utilizing those ensemble methods with simple datasets. Ensembles grasp the strength of each individual method and diminish the weaknesses at the same time.

Table 3. Maximum classification evaluation metrics by feature selection method, containing the number of features from Mushroom dataset after the value.

	Pearson	MRMR	Chi-square	ANOVA	Borda	Arithmetic Mean	Copeland
Mean Accuracy	0.9902 (4)	0.9902 (4)	0.9561 (20)	0.9902 (4)	0.9902 (3)	0.9902 (3)	0.9902 (3)
Mean Precision	0.9796 (4)	0.9796 (4)	0.9415 (20)	0.9796 (4)	0.9796 (3)	0.9796 (3)	0.9796 (3)
Mean Recall	1 (1)	1 (1)	0.9865 (3)	1 (2)	1 (1)	1 (1)	1 (1)
Mean F1 score	0.9897 (4)	0.9897 (4)	0.9538 (20)	0.9897 (3)	0.9897 (3)	0.9897 (3)	0.9897 (3)

Table 3 above depicts the number of features needed for each method to reach the maximum value for each evaluation metric, and what is interesting to see that the individual methods match the performance of the ensemble methods, except for Chi-square. The maximum metrics for ensemble methods are reached with only 3 features, while the individual methods need at least 4 features, Chi-square again being the different one needing all the 20 features to reach the maximum, when recall is not included. These results also showcase the fact that individual methods and ensemble methods can generate comparable results and with simple datasets the construction of ensembles doesn't always produce better results, however the fact that ensemble methods reached the stability faster with a smaller number of features.

7.2 Taiwanese Bankruptcy dataset

When using the same methods for the dataset on Taiwanese companies that went bankrupt, the results are quite interesting in a sense that in some points the ensemble methods perform better in terms of classification accuracy for example when compared to single ranking methods. As more features are added to the subset, the results of all the other methods improve as well. Before using the data, multicollinearity was checked and in case there were features that correlated significantly with each other, they were removed. The class imbalance for this dataset was significant and therefore oversampling was needed and was executed with the SMOTE method. After the classes were balanced, it was also easy to use the stratifying option to ensure that the classes remained balanced in the cross-validation datasets. Also, the data was normalized as the feature values had quite different ranges.

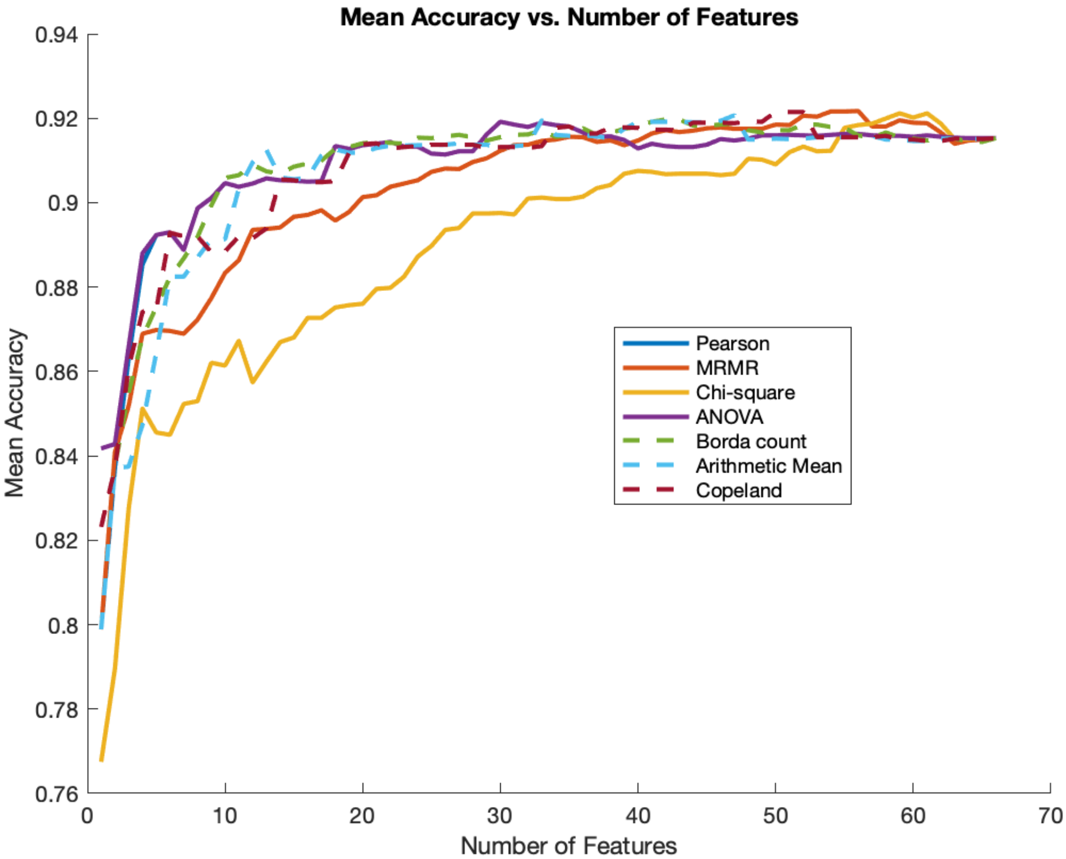

Figure 5. Mean classification accuracy of Taiwan bankruptcy dataset with different numbers of features broken down to each feature selection method.

As can be seen above from Figure 5, the accuracy improves very quickly with all the methods as feature are added to the subset. Again, as the problem is binary, it might be that the methods chosen suit the dataset quite well. When combining the accuracy evaluation with F1 score and the other chosen metrics, the model can be validated, and it can be said that the largest difference in the metrics comes from the chosen feature selection methods. Table 4 below depicts the development of the mean accuracy when using different numbers of features and how the different methos perform in terms of accuracy.

Table 4. Classification accuracy by FS method with different numbers of features selected from the Taiwan bankruptcy dataset.

Number of features	Pearson	MRMR	Chi-square	ANOVA	Borda	Arithmetic Mean	Copeland
2	0.8371	0.8409	0.7895	0.8428	0.8377	0.8371	0.8371
4	0.8854	0.8689	0.8512	0.8880	0.8681	0.8473	0.8742
6	0.8930	0.8696	0.8450	0.8930	0.8824	0.8825	0.8929
8	0.8987	0.8723	0.8530	0.8987	0.8919	0.8870	0.8920
10	0.9046	0.8834	0.8614	0.9046	0.9059	0.8913	0.8885
12	0.9045	0.8936	0.8574	0.9045	0.9090	0.9095	0.8916
14	0.9053	0.8941	0.8669	0.9053	0.9069	0.9060	0.9055
16	0.9050	0.8971	0.8727	0.9050	0.9093	0.9059	0.9049
18	0.9134	0.8958	0.8751	0.9134	0.9127	0.9126	0.9051
20	0.9136	0.9013	0.8760	0.9136	0.9141	0.9119	0.9138
22	0.9144	0.9038	0.8798	0.9144	0.9142	0.9135	0.9129
24	0.9133	0.9053	0.8872	0.9133	0.9154	0.9136	0.9134
26	0.9114	0.9081	0.8936	0.9114	0.9157	0.9137	0.9137
28	0.9122	0.9096	0.8974	0.9122	0.9155	0.9137	0.9137
30	0.9192	0.9123	0.8975	0.9192	0.9155	0.9128	0.9132
32	0.9180	0.9137	0.9010	0.9180	0.9162	0.9136	0.9130
34	0.9184	0.9150	0.9009	0.9184	0.9156	0.9161	0.9177
36	0.9166	0.9155	0.9014	0.9166	0.9176	0.9160	0.9164
38	0.9157	0.9147	0.9042	0.9157	0.9165	0.9152	0.9178
40	0.9129	0.9147	0.9075	0.9129	0.9181	0.9192	0.9177
42	0.9134	0.9172	0.9068	0.9134	0.9197	0.9192	0.9173
44	0.9132	0.9171	0.9069	0.9132	0.9183	0.9191	0.9189
46	0.9151	0.9178	0.9065	0.9151	0.9185	0.9194	0.9189
48	0.9152	0.9175	0.9104	0.9152	0.9172	0.9150	0.9193
50	0.9160	0.9185	0.9091	0.9160	0.9172	0.9152	0.9215
52	0.9160	0.9206	0.9133	0.9160	0.9185	0.9152	0.9215
54	0.9161	0.9216	0.9123	0.9161	0.9180	0.9155	0.9155
56	0.9163	0.9217	0.9184	0.9163	0.9160	0.9155	0.9155
58	0.9158	0.9181	0.9199	0.9158	0.9166	0.9150	0.9158
60	0.9156	0.9190	0.9202	0.9156	0.9152	0.9146	0.9150

As the number of features on the processed Taiwan bankruptcy dataset was higher, Tables 4 and 5 only present the number of features with an interval of two. However, what can be observed is that all the methods need more than 30 features to reach the maximum accuracy, some even more than 50 features. Another observation is that ensemble methods tend to reach a more stable accuracy value faster with a smaller number of features when compared to individual methods.

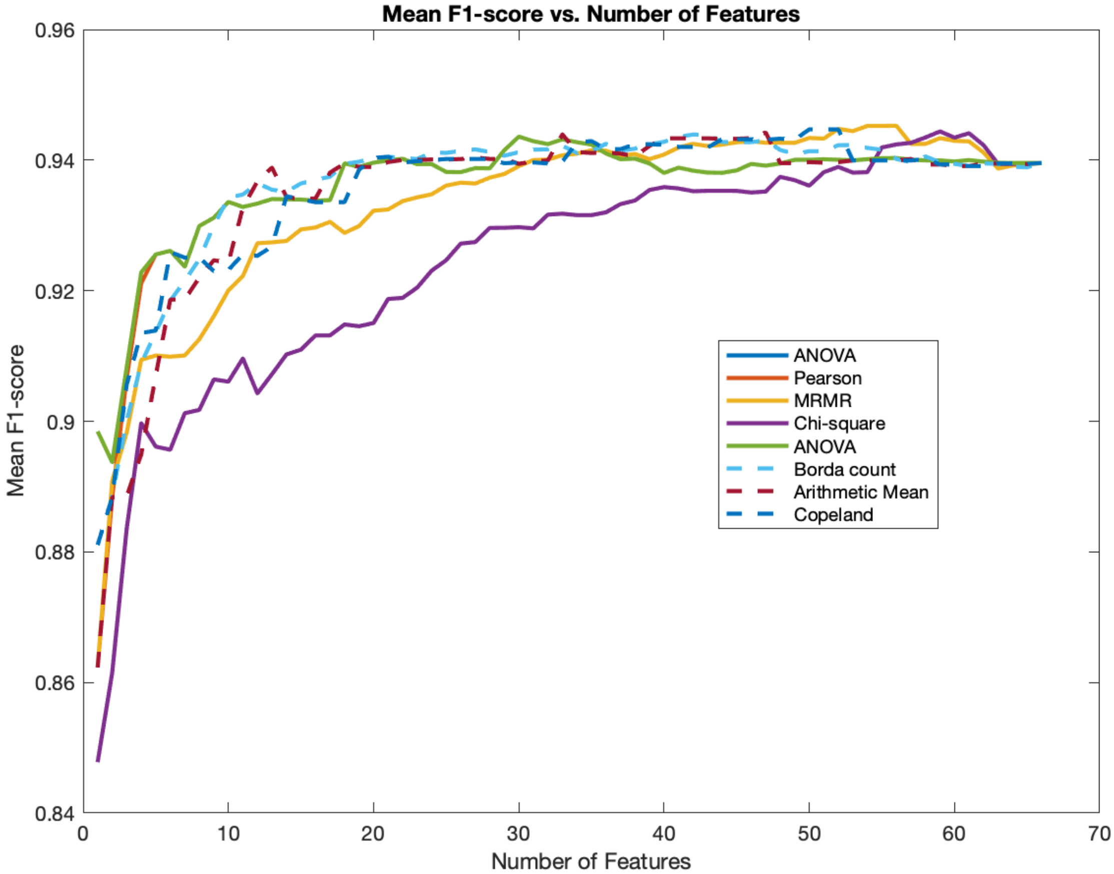

Figure 6. Mean F1-score of Taiwan bankruptcy dataset with different numbers of features broken down to each feature selection method.

The F1 score for the KNN classification model with different numbers of features follows the same kind of curve as the accuracy numbers, which tells us that also precision and recall are on a good level. When looking at all the metrics at the same time, it can be seen that overall, all the feature selection methods perform quite well in terms of classification, but especially the Copeland ensemble method seems to outperform all the other methods in several cases. Table 5 presents the development of F1-score when different numbers of features are selected and how the different methods match against each other regarding the F1-score.

Table 5. Classification F1-score by FS method with different numbers of features selected from the Taiwan bankruptcy dataset.

Number of features	Pearson	MRMR	Chi-square	ANOVA	Borda	Arithmetic Mean	Copeland
2	0.8884	0.8909	0.8614	0.8937	0.8886	0.8884	0.8884
4	0.9211	0.9094	0.8997	0.9228	0.9089	0.8949	0.9135
6	0.9261	0.9099	0.8956	0.9261	0.9185	0.9186	0.9259
8	0.9299	0.9126	0.9017	0.9299	0.9248	0.9220	0.9251
10	0.9336	0.9200	0.9061	0.9336	0.9343	0.9243	0.9232
12	0.9333	0.9273	0.9043	0.9333	0.9365	0.9368	0.9253
14	0.9340	0.9276	0.9102	0.9340	0.9351	0.9344	0.9345
16	0.9338	0.9297	0.9131	0.9338	0.9370	0.9341	0.9336
18	0.9395	0.9288	0.9148	0.9395	0.9394	0.9395	0.9336
20	0.9396	0.9322	0.9151	0.9396	0.9403	0.9389	0.9403
22	0.9402	0.9337	0.9189	0.9402	0.9403	0.9400	0.9396
24	0.9394	0.9348	0.9231	0.9394	0.9412	0.9401	0.9400
26	0.9382	0.9365	0.9272	0.9382	0.9414	0.9402	0.9402
28	0.9387	0.9373	0.9296	0.9387	0.9412	0.9402	0.9402
30	0.9436	0.9391	0.9297	0.9436	0.9412	0.9394	0.9396
32	0.9424	0.9401	0.9316	0.9424	0.9417	0.9399	0.9395
34	0.9427	0.9410	0.9316	0.9427	0.9411	0.9412	0.9427
36	0.9410	0.9414	0.9320	0.9410	0.9425	0.9412	0.9417
38	0.9402	0.9409	0.9338	0.9402	0.9418	0.9406	0.9424
40	0.9380	0.9409	0.9359	0.9380	0.9428	0.9433	0.9423
42	0.9384	0.9425	0.9352	0.9384	0.9439	0.9433	0.9420
44	0.9381	0.9424	0.9353	0.9381	0.9428	0.9433	0.9432
46	0.9394	0.9429	0.9350	0.9394	0.9427	0.9434	0.9432
48	0.9395	0.9427	0.9374	0.9395	0.9415	0.9395	0.9433
50	0.9400	0.9434	0.9361	0.9400	0.9414	0.9397	0.9447
52	0.9401	0.9448	0.9390	0.9401	0.9423	0.9397	0.9447
54	0.9401	0.9452	0.9382	0.9401	0.9419	0.9399	0.9399
56	0.9403	0.9453	0.9424	0.9403	0.9402	0.9399	0.9399
58	0.9400	0.9425	0.9435	0.9400	0.9406	0.9393	0.9402
60	0.9398	0.9429	0.9434	0.9398	0.9394	0.9391	0.9393

From Table 5 it can be observed that again, as the number of features increases, the F1-score seems to improve as well but after a certain threshold the number quickly stabilizes. For all the methods 30 or more features were needed to reach the maximum and some methods required more than 50 features in order to get to the maximum value.

From Table 6 we can see the maximum metric for each feature selection method and the number of features that the subset included when the maximum was reached. For Pearson method it seems that 30 was the optimal number of features as most of the maximums were reached with that, but for other methods the number of features required for the maximum accuracy was relatively higher.

Table 6. Maximum classification evaluation metrics by feature selection method, containing the number of features from Taiwan bankruptcy dataset after the value.

	Pearson	MRMR	Chi-square	ANOVA	Borda	Arithmetic Mean	Copeland
Mean Accuracy	0.9192 (30)	0.9217 (56)	0.9212 (61)	0.9192 (30)	0.9197 (42)	0.9207 (47)	**0.9215 (51)**
Mean Precision	0.9244 (30)	0.9323 (17)	0.9298 (22)	**0.9565 (1)**	0.9257 (19)	0.9271 (18)	0.9285 (21)
Mean Recall	0.9835 (61)	0.9825 (66)	0.9825 (65)	0.9835 (61)	**0.9836 (59)**	0.9825 (66)	0.9825 (66)
Mean F1 score	0.9436 (30)	**0.9453 (56)**	0.9444 (59)	0.9436 (30)	0.9439 (42)	0.9442 (47)	0.9447 (50)

As the classification model was again tested with different numbers of selected features based on the methods, it was interesting to see how the different evaluation metrics evolve when the number of features increases. When using the features selected by the Chi-square method it was seen that the metrics in some cases actually decreased quite a lot, and although this did actually happen with all the methods, it was more stable with the ensemble methods. The difference in all the metrics between single methods and ensemble methods however is quite minor, but again, when all the metrics are combined and looked into as a group, the ensemble methods perfom the best.

7.3 COIL2000 dataset

With COIL2000 dataset the results were also very interesting in a sense that the accuracy of the classifier seemed to take longer and a larger number of features before it was stabilized. What also happened is that by adding features, the noise of the dataset seemed to increase drastically as with some methods the decrease in accuracy was significant. It was noted that the classes were very imbalanced, so SMOTE method was utilized to oversample the minority class and reach a situation where the classes were more balanced. Multicollinearity was also checked and any features that strongly correlated with each other were removed.

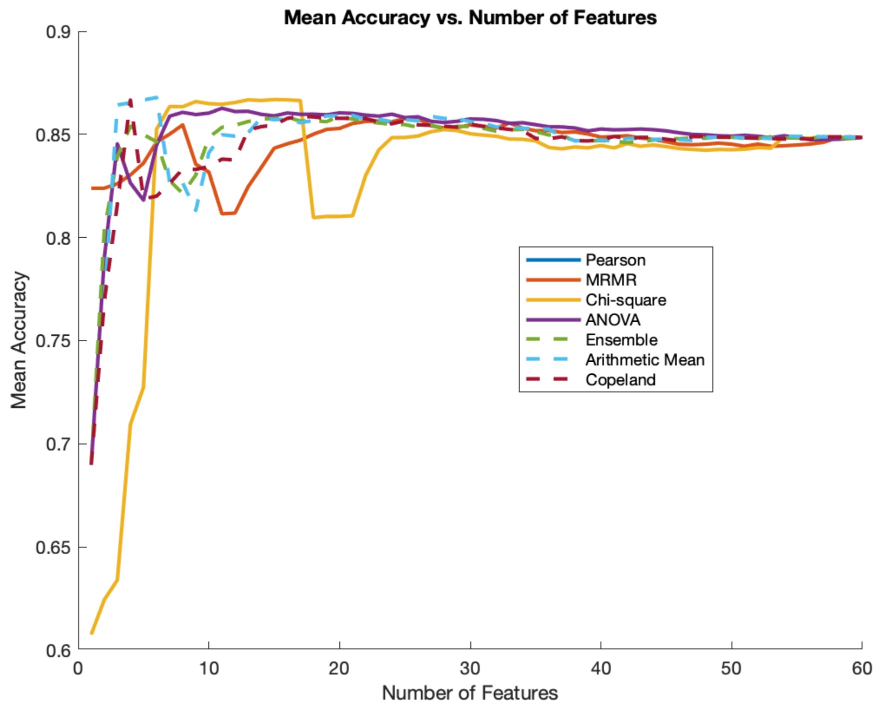

Figure 7. Mean classification accuracy of COIL2000 dataset with different numbers of features broken down to each feature selection method.

What is interesting and can be observed from Figure 7 that with all the methods the accuracy in fact decreases in some instanced when features are added, signalling that the noise in the dataset might be increased at the same time. The accuracy in general however remains on a particularly good level. The dataset in question is much more complex already, contributing to the fact that the feature selection phase is already much more impactful.

Table 7. Classification accuracy by FS method with different numbers of features selected from COIL2000 dataset.

Number of features	Pearson	MRMR	Chi-square	ANOVA	Borda	Arithmetic Mean	Copeland
2	0.7899	0.8239	0.6244	0.7899	0.8035	0.7689	0.7689
4	0.8264	0.8303	0.7095	0.8264	0.8540	0.8653	0.8667
6	0.8448	0.8463	0.8525	0.8448	0.8465	0.8679	0.8200
8	0.8607	0.8546	0.8634	0.8607	0.8213	0.8264	0.8332
10	0.8603	0.8317	0.8649	0.8603	0.8485	0.8411	0.8344
12	0.8611	0.8118	0.8655	0.8611	0.8544	0.8491	0.8376
14	0.8597	0.8338	0.8664	0.8597	0.8574	0.8580	0.8537
16	0.8603	0.8454	0.8667	0.8603	0.8583	0.8574	0.8577
18	0.8598	0.8501	0.8096	0.8598	0.8567	0.8567	0.8584
20	0.8604	0.8529	0.8102	0.8604	0.8588	0.8591	0.8578
22	0.8593	0.8563	0.8302	0.8593	0.8562	0.8573	0.8576
24	0.8598	0.8561	0.8484	0.8598	0.8548	0.8561	0.8551
26	0.8585	0.8577	0.8492	0.8585	0.8534	0.8566	0.8546
28	0.8558	0.8531	0.8523	0.8558	0.8526	0.8577	0.8539
30	0.8575	0.8546	0.8503	0.8575	0.8538	0.8562	0.8549
32	0.8566	0.8531	0.8491	0.8566	0.8515	0.8536	0.8531
34	0.8556	0.8530	0.8476	0.8556	0.8524	0.8530	0.8517
36	0.8537	0.8519	0.8436	0.8537	0.8497	0.8526	0.8479
38	0.8531	0.8512	0.8439	0.8531	0.8468	0.8470	0.8470
40	0.8526	0.8487	0.8446	0.8526	0.8470	0.8473	0.8473
42	0.8523	0.8493	0.8456	0.8523	0.8460	0.8480	0.8480
44	0.8523	0.8476	0.8449	0.8523	0.8473	0.8476	0.8486
46	0.8507	0.8452	0.8431	0.8507	0.8479	0.8473	0.8469
48	0.8498	0.8452	0.8423	0.8498	0.8481	0.8486	0.8489
50	0.8491	0.8452	0.8426	0.8491	0.8486	0.8486	0.8486
52	0.8491	0.8453	0.8436	0.8491	0.8480	0.8483	0.8481
54	0.8493	0.8445	0.8490	0.8493	0.8483	0.8492	0.8477
56	0.8474	0.8453	0.8482	0.8474	0.8481	0.8491	0.8477
58	0.8476	0.8482	0.8483	0.8476	0.8486	0.8487	0.8488
60	0.8485	0.8485	0.8485	0.8485	0.8485	0.8485	0.8485

As the number of included features in the COIL2000 dataset was higher, Tables 7 and 8 only present features with an interval of two. All the methods showed improvement in classification accuracy when features were added to the subset and reached the maximum value with a relatively small number of features, some methods already with less than 10 features. Some methods needed more than 20 features to reach the maximum accuracy, and these methods were mainly individual methods. Ensemble methods again reached the maximum accuracy faster.

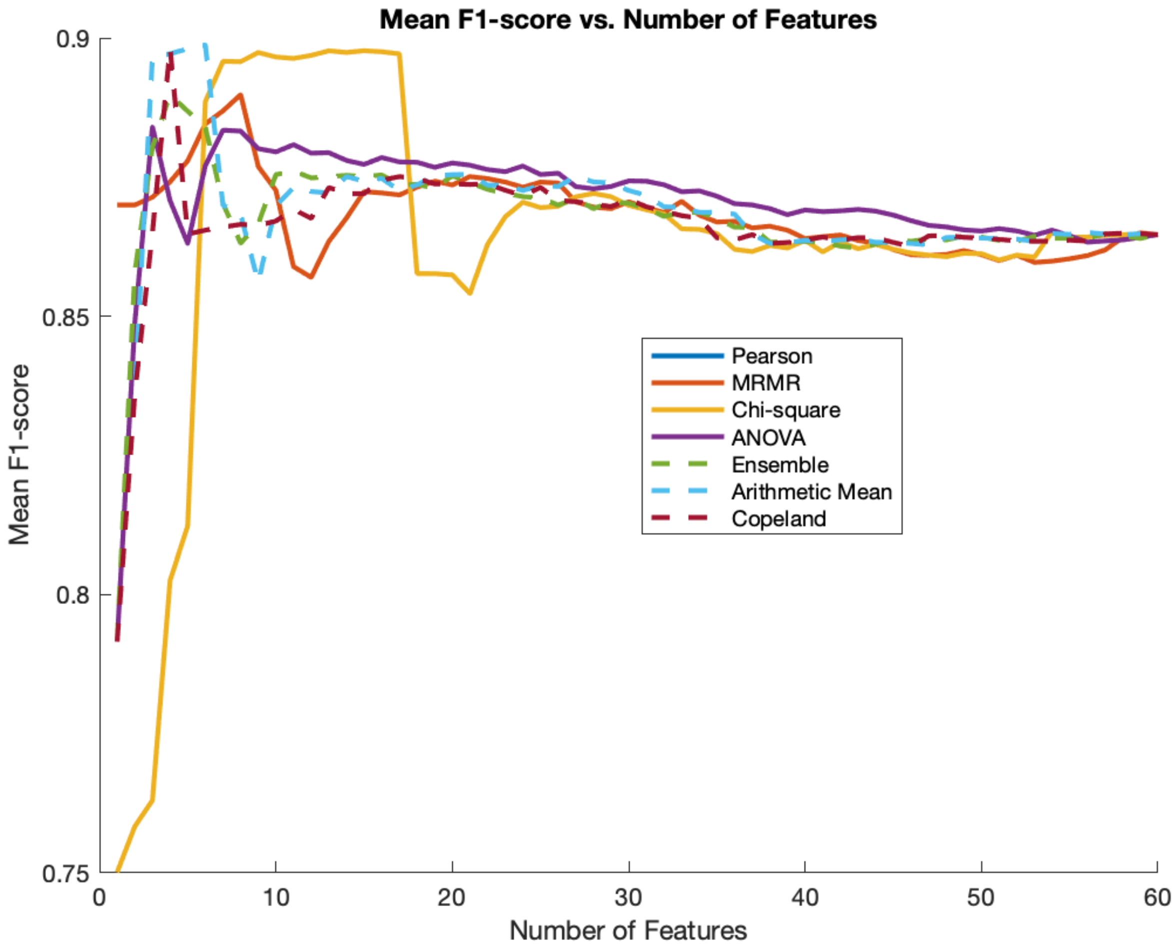

Figure 8. Mean F1-score of COIL2000 dataset with different numbers of features broken down to each feature selection method.

As can be observed from Figure 8, the mean F1-score follows the same kind of curve as the accuracy does with the COIL2000 dataset. However, the scale must be noted, so even though the changes in the actual graph seem to be large, in absolute values the difference is quite minor. What is interesting is the sudden decrease in the score value when around 10 features are included in the subset, and that the decrease is slight also when adding even more features to the subset. Overall, the ensemble methods seem to match the performance of individual methods and outperform those in some instances, while ANOVA method in general seems to be the best FS method regarding the overall F1-score.

Table 8. Classification F1-score by FS method with different numbers of features selected from COIL2000 dataset.

Number of features	Pearson	MRMR	Chi-square	ANOVA	Borda	Arithmetic Mean	Copeland
2	0.8487	0.8700	0.7583	0.8487	0.8579	0.8360	0.8360
4	0.8708	0.8741	0.8025	0.8708	0.8898	0.8972	0.8982
6	0.8771	0.8845	0.8886	0.8771	0.8840	0.8987	0.8655
8	0.8833	0.8898	0.8957	0.8833	0.8631	0.8683	0.8665
10	0.8795	0.8726	0.8966	0.8795	0.8755	0.8698	0.8671
12	0.8793	0.8570	0.8969	0.8793	0.8749	0.8724	0.8676
14	0.8780	0.8676	0.8974	0.8780	0.8753	0.8751	0.8720
16	0.8785	0.8722	0.8975	0.8785	0.8754	0.8748	0.8745
18	0.8777	0.8732	0.8577	0.8777	0.8737	0.8736	0.8748
20	0.8775	0.8736	0.8574	0.8775	0.8752	0.8754	0.8738
22	0.8764	0.8747	0.8629	0.8764	0.8728	0.8735	0.8738
24	0.8770	0.8732	0.8705	0.8770	0.8715	0.8726	0.8717
26	0.8757	0.8739	0.8698	0.8757	0.8700	0.8733	0.8708
28	0.8729	0.8696	0.8721	0.8729	0.8693	0.8741	0.8701
30	0.8743	0.8706	0.8699	0.8743	0.8703	0.8726	0.8710
32	0.8736	0.8686	0.8684	0.8736	0.8680	0.8697	0.8689
34	0.8725	0.8682	0.8656	0.8725	0.8687	0.8687	0.8676
36	0.8702	0.8671	0.8620	0.8702	0.8660	0.8683	0.8637
38	0.8693	0.8662	0.8627	0.8693	0.8633	0.8631	0.8631
40	0.8691	0.8640	0.8636	0.8691	0.8634	0.8636	0.8637
42	0.8689	0.8646	0.8633	0.8689	0.8626	0.8637	0.8637
44	0.8689	0.8631	0.8629	0.8689	0.8630	0.8633	0.8641
46	0.8672	0.8611	0.8614	0.8672	0.8635	0.8631	0.8625
48	0.8660	0.8611	0.8607	0.8660	0.8637	0.8641	0.8645
50	0.8653	0.8611	0.8612	0.8653	0.8642	0.8641	0.8641
52	0.8653	0.8610	0.8610	0.8653	0.8635	0.8639	0.8635
54	0.8655	0.8599	0.8652	0.8655	0.8640	0.8649	0.8635
56	0.8633	0.8609	0.8642	0.8633	0.8640	0.8650	0.8635
58	0.8637	0.8641	0.8646	0.8637	0.8645	0.8646	0.8649
60	0.8647	0.8647	0.8647	0.8647	0.8647	0.8647	0.8647

Table 8 still summarizes the F1-score for different numbers of features and shows that overall, the F1-score stays quite similar for each method even though numbers are added, and the differences are only minor. What is noteworthy that for most methods the maximum F1-score is reached with only 8 features or less, and only one method requires 15 features to reach the maximum.

Table 9 represents the maximum values for each metric by feature selection method. The number of features required to reach the maximum are presented after the actual metric value. For Pearson method it seems that 11 was the optimal number of features to reach the maximum accuracy as for some ensemble methods only needed less than 10 features.

Table 9. Maximum classification evaluation metrics by feature selection method, containing the number of features from COIL2000 dataset after the value.

	Pearson	MRMR	Chi-square	ANOVA	Borda	Arithmetic Mean	Copeland
Mean Accuracy	0.8626 (11)	0.8577 (26)	0.8669 (15)	0.8626 (11)	0.8588 (20)	0.8679 (6)	0.8667 (4)
Mean Precision	1	1	1	**1**	1	1	1
Mean Recall	0.9135 (55)	0.9208 (37)	0.9126 (56)	0.9135 (55)	0.9168 (53)	0.9164 (50)	0.9182 (41)
Mean F1 score	0.8840 (3)	0.8898 (8)	0.8977 (15)	0.8840 (3)	0.8898 (4)	0.8987 (6)	0.8982 (4)

What is notable in Table 9 is that the precision for all the methods has a maximum value of 1. This means that the model in question predicts positive instances perfectly but does not tell us anything on whether the model can actually predict all positive cases, only the true positives. It avoids false positives, but when combining precision with other metrics, it can be seen that the model performs well. Borda count as an ensemble method seems to need more features to reach the maximum accuracy compared to other ensemble methods, but overall, the maximum accuracy for all methods is quite similar. However, arithmetic mean and Copeland score seem to perform the best out of all and reach the maximum values with a smaller number of features included. Again, ensemble methods seem to reach stable results faster and deliver outperforming results but only in some instances. Individual methods are still powerful and constructing ensembles does not always add value in terms of classification performance, although they deliver comparable results.

7.4 Breast cancer Wisconsin dataset

The same methods were tested with the breast cancer dataset as well, which contains information on whether a breast cancer tumour is malignant or not. As the features had very varying ranges of values, they were normalized before moving on to feature selection. Classes were however balanced, so no oversampling was needed. Checks for multicollinearity were executed and in case there were features that correlated strongly with each other, they were removed. The selected ensemble methods provided comparable results to the single ranking methods but however did not seem to overperform all of those.

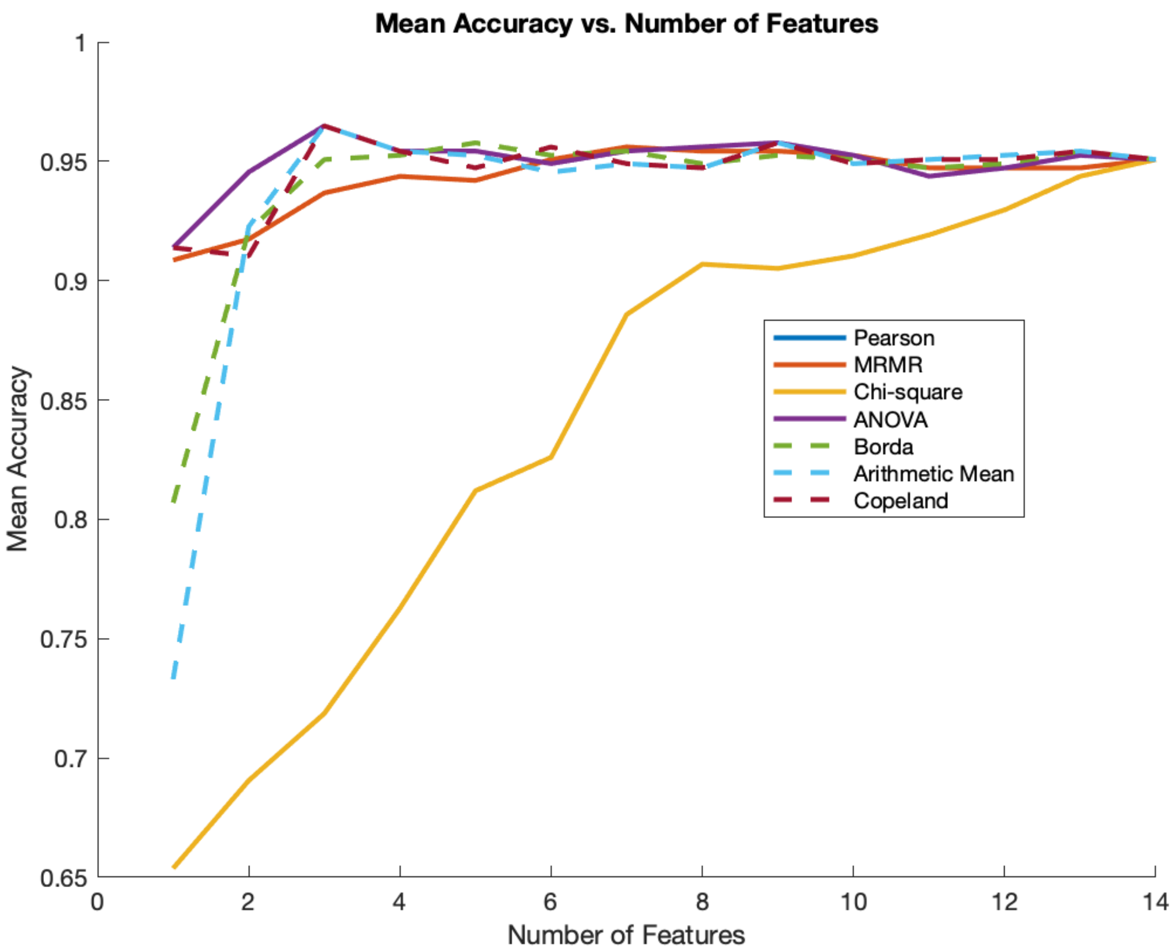

Figure 9. Mean classification accuracy of Breast Cancer dataset with different numbers of features broken down to each feature selection method.

What is also notable is that the Chi-square method performed poorly with this dataset when it comes to classification, mainly because of the nature of the dataset. All the methods however produce high classification accuracy with a high number of features in the end, which is somehow understandable as although noise might be added through irrelevant features, there are also more data points to use for classification.

Table 10. Classification accuracy by FS method with different numbers of features selected from the Breast Cancer dataset.

Number of features / Method	Pearson	MRMR	Chi-square	ANOVA	Borda	Arithmetic Mean	Copeland
1	0.9138	0.9086	0.6538	0.9138	0.8069	0.7329	0.9138
2	0.9455	0.9174	0.6905	0.9455	0.9209	0.9226	0.9103
3	0.9648	0.9367	0.7186	0.9648	0.9508	0.9648	0.9648
4	0.9543	0.9437	0.7627	0.9543	0.9525	0.9543	0.9543
5	0.9543	0.9420	0.8120	0.9543	0.9578	0.9525	0.9472
6	0.9490	0.9508	0.8260	0.9490	0.9525	0.9455	0.9560
7	0.9543	0.9560	0.8858	0.9543	0.9543	0.9490	0.9490
8	0.9560	0.9543	0.9069	0.9560	0.9490	0.9472	0.9472
9	0.9578	0.9543	0.9051	0.9578	0.9525	0.9578	0.9578
10	0.9525	0.9525	0.9104	0.9525	0.9508	0.9490	0.9490
11	0.9437	0.9472	0.9192	0.9437	0.9472	0.9508	0.9508
12	0.9472	0.9472	0.9296	0.9472	0.9490	0.9525	0.9507
13	0.9525	0.9472	0.9437	0.9525	0.9543	0.9543	0.9543
14	0.9508	0.9508	0.9508	0.9508	0.9508	0.9508	0.9508

From Table 10 it can be observed that the accuracy differs based on method especially with a small number of features. Chi-square does not perform so well and needs more than 5 features to reach satisfactory accuracy levels especially when compared to other methods. All the other methods need only 3 features selected to the subset to improve accuracy it can be seen that the ensemble methods and the individual methods except Chi-square deliver comparable results in terms of accuracy, so ensembles do not outperform the individual methods.

Again, when looking at all the evaluation metrics together, it can be stated that the performance does not really differ that much when changing the feature selection method between individual and ensemble. However, ensemble methods do outperform the single methods in some instances, but clearly single methods are also very strong when it comes to classification. The F1-score for the breast cancer dataset quite closely follows the same kind of curve as the classification accuracy, also emphasizing the fact that precision and recall are on a prominent level.

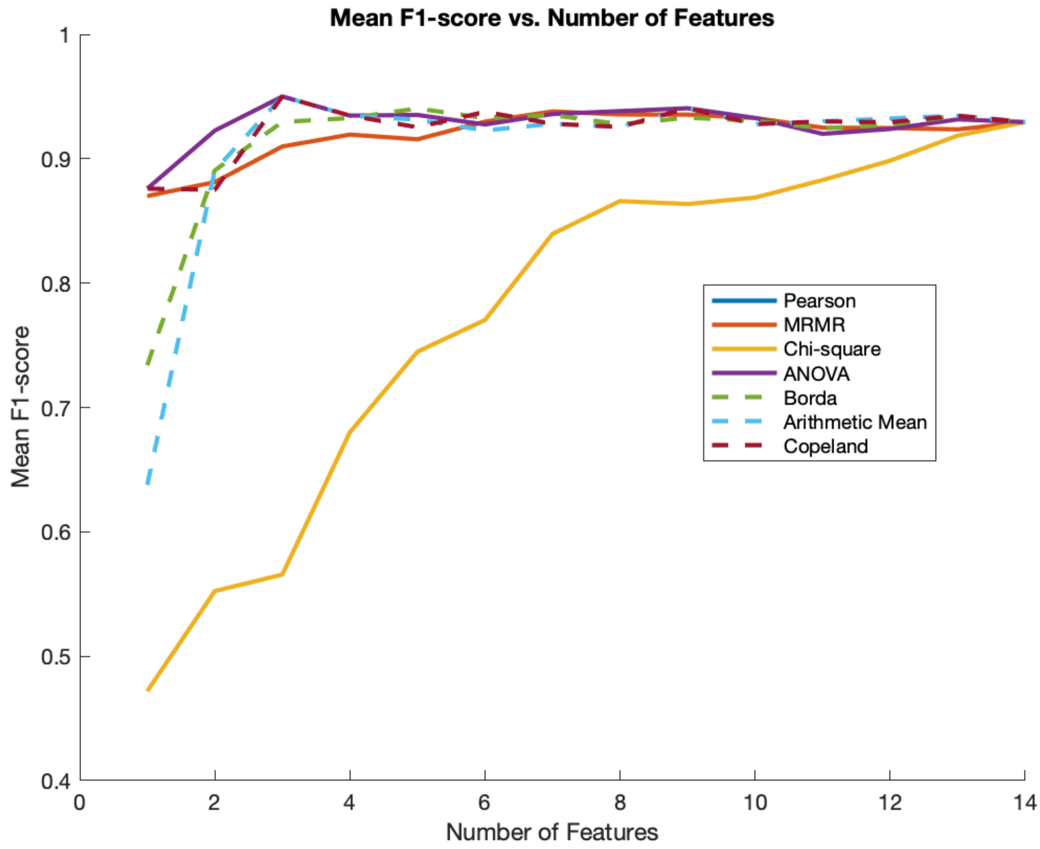

Figure 10. Mean F1-score of Breast Cancer dataset with different numbers of features broken down to each feature selection method.

As the number of features in the subset increases, the classification performance seems to generally improve or at least stay stable in terms of the F1-score as well, although it does also seem that in some cases it decreases very little when adding features. The arithmetic mean ensemble method does indeed slightly outperform the single methods when using 5 best features from the dataset.

Table 11. Classification F1-score by FS method with different numbers of features selected from the Breast Cancer dataset.

Number of features / Method	Pearson	MRMR	Chi-square	ANOVA	Borda	Arithmetic Mean	Copeland
1	0.8763	0.8458	0.5036	0.8763	0.7340	0.8635	0.8635
2	0.9245	0.8869	0.5845	0.9245	0.8980	0.9245	0.9245
3	0.9340	0.9223	0.6218	0.9340	0.9267	0.9183	0.9234
4	0.9328	0.9193	0.7144	0.9328	0.9346	0.9229	0.9427
5	0.9253	0.9252	0.7522	0.9253	0.9296	0.9207	0.9277
6	0.9271	0.9307	0.7643	0.9271	0.9247	0.9269	0.9269
7	0.9273	0.9257	0.8764	0.9273	0.9198	0.9245	0.9245
8	0.9328	0.9270	0.8840	0.9328	0.9181	0.9223	0.9223
9	0.9353	0.9351	0.8838	0.9353	0.9257	0.9269	0.9269
10	0.9343	0.9374	0.8789	0.9343	0.9343	0.9374	0.9374
11	0.9222	0.9354	0.8865	0.9222	0.9293	0.9385	0.9293
12	0.9194	0.9283	0.9270	0.9194	0.9293	0.9293	0.9293
13	0.9269	0.9264	0.9239	0.9269	0.9345	0.9345	0.9345
14	0.9243	0.9243	0.9243	0.9243	0.9243	0.9243	0.9243

From Table 11 it can be observed that all the other methods except Chi-square present satisfactory F1-scores already with a small number of features. What is also notable is that the other individual methods seem to deliver the same kind of results as the ensemble methods, but in most cases the difference is subtle so that the ensemble methods outperform the individual methods. Ensemble methods, namely Borda count and Copeland score in this case reach the maximum F1-score quite rapidly with only 4 features selected to the subset, while for all the other methods almost 10 or even more features are needed. The stability of the F1-score is also very visible in Table 11 also, as for all methods except Chi-square the F1-score seems to float around 0.93, regardless of the number of features added once the maximum value is reached.

Table 12. Maximum classification evaluation metrics by feature selection method, containing the number of features selected from the Breast Cancer dataset after the value.

	Pearson	MRMR	Chi-square	ANOVA	Borda	Arithmetic Mean	Copeland
Mean Accuracy	0.9542 (9)	0.9560 (10)	0.9489 (12)	0.9542 (9)	0.9542 (4)	0.9560 (11)	**0.9595 (4)**
Mean Precision	0.9056 (9)	0.9104 (11)	0.8916 (14)	0.9056 (9)	0.9056 (13)	**0.9152 (11)**	0.9102 (4)
Mean Recall	0.9759 (10)	0.9806 (4)	0.9719 (12)	0.9759 (10)	0.9811 (5)	0.9764 (5)	**0.9813 (4)**
Mean F1 score	0.9353 (9)	0.9374 (10)	0.9270 (12)	0.9353 (9)	0.9346 (4)	0.9385 (11)	**0.9427 (4)**

As can be observed from Table 12, the ensemble methods slightly outperformed the individual methos in terms of the maximum value of each metric. Copeland score was able to grasp the maximum values already with just 4 features, while other methods in general seemed to need more features selected in the subset. However, the differences were very subtle with all the evaluation metrics and between individual and ensemble methods.

When the number of selected features increases, there is still a very subtle difference in the results between single and ensemble methods – the latter providing comparative results and even slightly better when it comes to accuracy for example. As the results indicate, the ensemble methods slightly outperform the individual methods – but what is noteworthy is that the individual methods still present very strong classification performance, and the difference is very subtle.

8 Conclusions

In this thesis, we examined the performance of ensemble methods and single feature ranking methods on a range of datasets to assess their relative effectiveness in feature selection and predictive modelling. In this section the conclusions are presented while answering the original research questions.

The objective of this research was to investigate whether ensemble feature selection methods are superior to individual ranking methods in terms of classification performance. The method was to create feature subsets with individual feature ranking methods and train a classifier with those subsets. Then the subsets were combined to an ensemble and the same classifier was trained with the new subset. After that, the trained classifiers were tested and evaluated with different performance metrics to see whether the ensemble subset would present better results than the subsets from individual methods. The question was also whether the construction of such ensembles is justifiable so that there is indeed an improvement in performance and how the selection of the ensemble method would impact the results. Our findings indicate that while ensemble methods were slightly better overall, the difference in performance compared to single feature ranking methods was not overwhelming with the chosen methods and datasets.

- Do feature subsets generated by ensemble feature ranking provide higher classification performance statistics compared to a single feature selection method?

When addressing the first research question, the analysis suggests that while ensemble methods offer a slight advantage over single feature ranking methods, the difference is not that visible with the chosen methods and datasets. The choice of method should be carefully considered based on the dataset's specific characteristics and the problem context. The decision to use ensemble methods or single feature ranking methods should be driven by the nature of the dataset and the computational resources available. In scenarios where datasets are high-dimensional but only a few features are relevant, single feature ranking methods may be more advantageous.

It can be stated that based on the results the ensemble methods do not always provide superior classification accuracy results or higher performance statistics as a whole when comparing to a single feature selection method. It must be noted though that ensemble methods were slightly better in providing better results with the chosen classification method.

- How does the selected aggregation method impact classification performance?

The ensemble aggregation methods examined (Borda count, arithmetic mean, Copeland's method) exhibited notable stability, making them a reliable choice in scenarios where consistency across different datasets is critical. It does seem that the selected aggregation method, or in other words the chosen ensemble method influences the classification performance. All the three methods were in the end quite simple and straightforward, but there was still a difference across the datasets when comparing the classification results of the different ensemble methods as well. Furthermore, the implication is that the single methods chosen for the ensemble need to be carefully thought as well.

Ultimately, this thesis highlights that there is no one-size-fits-all approach to feature selection. Both ensemble methods and single feature ranking methods have their strengths and limitations. The key is to understand these trade-offs and select the method that best aligns with the specific needs and constraints of the problem at hand.

Ensemble methods generally demonstrated a slight edge in prediction accuracy over single feature ranking methods. However, this improvement was not substantial across all datasets, indicating that the benefits of ensemble techniques are context-dependent. The effectiveness of ensemble methods appeared to be highly dependent on the specific characteristics of the dataset. For example, in datasets with high dimensionality but sparse relevant features, single feature ranking methods performed comparably well, sometimes even surpassing ensemble methods.

Single feature ranking methods consistently demonstrated lower computational requirements. In resource-constrained environments or when rapid feature selection is necessary, these methods provided a significant advantage by reducing computational overhead while maintaining satisfactory performance. Single feature ranking methods offer greater interpretability, which is crucial in fields where understanding the contribution of individual features is important. This interpretability can facilitate better insights and decision-making, particularly in domains such as healthcare and finance.

Although ensemble methods typically enhance robustness by aggregating multiple models, the added complexity did not always translate to significantly improved performance. However, ensemble methods generally showed greater stability across different datasets, which could be attributed to the aggregation of multiple models or ranking methods. The ensemble methods researched in this study, such as Borda count, arithmetic mean, and Copeland's method, were relatively simple yet still demonstrated considerable stability. Despite the marginal improvement in prediction accuracy, ensemble methods were generally more stable across different datasets. The ability to aggregate results from multiple models or ranking methods mitigated the impact of any single model's inferior performance, leading to more consistent outcomes.

Earlier research strongly suggests and indicates that ensemble methods would be superior to single feature selection methods in a variety of cases. Creating an efficient ensemble then again possibly requires a more extensive approach in order to choose the most appropriate methods that would be included in the ensemble. For example, Drotár, Gazga & Gazda (2017) included more base feature selection methods in order to create the ensemble and also the aggregation methods were slightly different. This emphasizes the fact that the ensemble creation phase is vital when choosing the single methods, because as their research also showed, the ensemble method was not always outperforming the single methods in all the datasets. Furthermore Seijo-Pardo, Porto-Díaz, Bolón-Canedo & Alonso-Betanzos (2016) approached the feature selection problem in a slightly different way although there were many similarities as well. The individual methods were a bit different and so was the combination method, and the aim was to also improve the computation load of using those methods. Classification results were very comparable and in some cases the ensemble methods did in fact outperform the individual methods.

The classification evaluation metrics presented here are not the only factor that should guide the selection process but also the computation load of the methods and the need for simplicity over complexity. For simple problems the individual methods still work efficiently, and one should not start to develop complex ensembles just for the sake of developing those, but in order to improve the classification phase for example.

This research contributes to the theoretical understanding behind individual methods and ensemble methods and to the actual effective usage of both methods. Moreover, the research showcases the situations where individual methods are more suitable to utilize whereas ensemble methods, while still having their place, require usually more complex problems. Building also on earlier remarks, the research helps clarify the strengths and limitations of different approaches and provides guidance on their use in various contexts.

Future research then could focus on how to choose the best individual methods for which problem and which combination method to utilize – in overall ensemble methods and the whole idea behind ensembles has shown great promise in solving different problems, such as classification within this research.

References

Azar, A. T. & El-Metwally, S. M. 2013. Decision Tree Classifiers for Automated Medical Diagnosis. Neural computing & applications 23.7–8: 2387–2403. Web.

Al-Sharrah, G. 2010. Ranking Using the Copeland Score: A Comparison with the Hasse Diagram. Journal of chemical information and modeling. [Online] 50 (5), 785–791.

Biswas, S., Bordoloi, M. & Purkayastha, B. 2016. Review on Feature Selection and Classification using Neuro-Fuzzy Approaches. International Journal of Applied Evolutionary Computation, vol. 7, no. 4, pp. 28-44.

Bolón-Canedo, V. & Alonso-Betanzos, A. 2018. Recent Advances in Ensembles for Feature Selection. Intelligent Systems Reference Library, Volume 147.

Bolón-Canedo, V. & Alonso-Betanzos, A. 2019 Ensembles for feature selection: A review and future trends. Information fusion. [Online] 521–12.

Bolón-Canedo, V., Sánchez-Maroño, N. & Alonso-Betanzos, A. 2012. A review of feature selection methods on synthetic data. Knowledge and Information Systems. 34.

Borandağ, E., Özçift, A. & Kaygusuz, Y. 2021. Development of majority vote ensemble feature selection algorithm augmented with rank allocation to enhance Turkish text categorization. Turkish Journal of Electrical Engineering and Computer Sciences: Volume 29, No. 2, Article 3.

Brown, G. 2011. Ensemble Learning. In: Sammut, C., Webb, G.I. (eds) Encyclopedia of Machine Learning.

Chandrashekar, G. & Sahin, F. 2014. A survey on feature selection methods. Computers & Electrical Engineering. Volume 40, Issue 1. 16-28.

Chen, R., Dewi, C., Huang, S. & Caraka, R. 2020. Selecting critical features for data classification based on machine learning methods. Journal of Big Data 7, 52.

de Ville, B. 2013. Decision trees. WIREs Computational Statistics, 5, pp. 448-455.

Ding, C. & Peng, H. 2005. Minimum redundancy feature selection from microarray gene expression data. Journal of Bioinformatics and Computational Biology. Vol. 3, Number 2, pp. 185–205.

Drotár, P, Gazda, M & Gazda, J. 2017. Heterogeneous ensemble feature selection based on weighted Borda count. 2017 9th International Conference on Information Technology and Electrical Engineering (ICITEE), Phuket, Thailand, 2017, pp. 1-4.

Dumitrescu, E., Sullivan, H., Hurlin, C. & Tokpavi, S. 2022. European journal of operational research, Vol.297 (3), p.1178-1192.

Effrosynidis, D. & Avi Arampatzis, A. 2021. An evaluation of feature selection methods for environmental data. Ecological Informatics, Volume 61.

El-Hajj, W. & Hazem Hajj, H. 2022. An optimal approach for text feature selection, Computer Speech & Language, Volume 74.

Foxwell, H. 2020. Creating Good Data: A Guide to Dataset Structure and Data Representation.

Guan, D., Yuan, W., Lee, Y. K., Najeebullah, K., & Rasel, M. K. 2014. A Review of Ensemble Learning Based Feature Selection. IETE Technical Review, 31(3), 190–198.

Guyon, I. & Elisseeff, A. 2003. An Introduction of Variable and Feature Selection. Machine Learning Research Special Issue on Variable and Feature Selection 3. 1157 - 1182.

He, Y., Fataliyev, K., Wang, L. 2013. Feature Selection for Stock Market Analysis. In: Lee, M., Hirose, A., Hou, ZG., Kil, R.M. (eds) Neural Information Processing. ICONIP 2013. Lecture Notes in Computer Science, vol 8227.

Hoque, N., Singh, M. & Bhattacharyya, D.K. 2018. EFS-MI: an ensemble feature selection method for classification. Complex & Intelligent Systems, Volume 4, 105–118.

Hua Han, S., Xiu Lu, S & C.H. Leung, S. 2012. Segmentation of telecom customers based on customer value by decision tree model. Expert Systems with Applications, Volume 39, Issue 4, Pages 3964-3973.

Huang, D., Liu, Z. & Wu, D. 2023. Research on Ensemble Learning-Based Feature Selection Method for Time-Series Prediction. Applied Sciences, Volume 14.

Jabbar, H.K., & Khan, R.Z. 2014. Methods to Avoid Over-Fitting and Under-Fitting in Supervised Machine Learning (Comparative Study).

Kataria, A. & Singh, M. 2013. A Review of Data Classification Using K-Nearest Neighbour Algorithm. International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 6.

Khoder, A. & F. Dornaika, F. 2021. Ensemble learning via feature selection and multiple transformed subsets: Application to image classification. Applied Soft Computing, Volume 113, Part B.

Kohavi, R. & John, G. (1997). Wrappers for Feature Subset Selection. Artificial Intelligence. 97. 273-324.

Kumar, L. & Bharti, K.K. 2019. A novel hybrid BPSO–SCA approach for feature selection. Natural Computing 20. 39–61.

Kwon, H., Park, J. & Lee, Y. 2019. Stacking Ensemble Technique for Classifying Breast Cancer. Healthcare Informatics Research 2019, 25, pp. 283-288.

Lessmann, S., Voß, S. 2009. Feature Selection in Marketing Applications. In: Huang, R., Yang, Q., Pei, J., Gama, J., Meng, X., Li, X. (eds) Advanced Data Mining and Applications. ADMA 2009. Lecture Notes in Computer Science, vol 5678.

Li, H. et al. (2024). Machine Learning Methods. 1st ed. 2024. [Online]. Singapore: Springer Nature Singapore.

Li, HY., Dong, L., Zhou, WD. et al. 2023. Development and validation of medical record-based logistic regression and machine learning models to diagnose diabetic retinopathy. Graefes Arch Clin Exp Ophthalmol 261, 681–689.

Li, J., Chang, K., Wang, S., Morstatter, F., Trevino, R., Tang, J. and Liu, H. 2017. Feature Selection: A Data Perspective. ACM Comput. Surv. 50, 6, Article 94, 45 pages.

Mera-Gaona, M., López, D.M. &Vargas-Canas, R. 2021. An Ensemble Feature Selection Approach to Identify Relevant Features from EEG Signals. Applied Sciences, 11, 6983.

Mushroom. 1987. UCI Machine Learning Repository. https://doi.org/10.24432/C5959T.

Noureldien, N. A, & Mahmoud, S. (2021). The Efficiency of Aggregation Methods in Ensemble Filter Feature Selection Models. Transactions on Machine Learning and Artificial Intelligence, 9(4). 39-51.

Owusu-Adjei, M., Ben Hayfron-Acquah, J., Frimpong, T. & Abdul-Salaam, G. 2023. Imbalanced class distribution and performance evaluation metrics: A systematic review of prediction accuracy for determining model performance in healthcare systems. PLOS Digit Health 2.

Panda, D., Ray, R., Dash, S.R. 2020. Feature Selection: Role in Designing Smart Healthcare Models. In: Pattnaik, P., Mohanty, S., Mohanty, S. (eds) Smart Healthcare Analytics in IoT Enabled Environment. Intelligent Systems Reference Library, vol 178.

Pes, B. 2020. Ensemble feature selection for high-dimensional data: a stability analysis across multiple domains. Neural Computing & Applications 32, pp. 5951–5973.

Pudjihartono, N., Fadason, T., Kempa-Liehr, A. & O'Sullivan, J. 2022. A Review of Feature Selection Methods for Machine Learning-Based Disease Risk Prediction. Frontiers in Bioinformatics.

Punch, William & Goodman, Erik & Pei, Min & Chia-Shun, Lai & Hovland, Paul & Enbody, R.J. 1993. Further Research on Feature Selection and Classification Using Genetic Algorithms. 557-564.

Re, M. & Valentini, G. 2012. Ensemble methods: A review. Advances in Machine Learning and Data Mining for Astronomy.

Reby, D., Lek, S., Dimopoulos, I., Joachim, J., Lauga, J. & Stéphane Aulagnier, S. 1997. Artificial neural networks as a classification method in the behavioural sciences. Behavioural Processes, Volume 40, Issue 1, pp. 35-43.

Rokach, L. 2009. Pattern Classification Using Ensemble Methods. Series In Machine Perception and Artificial Intelligence, Volume 75.

Rong, M., Gong, D. & Gao, X. 2019. Feature Selection and Its Use in Big Data: Challenges, Methods, and Trends. IEEE Access, vol. 7, pp. 19709-19725.

Rutherford, A. 2001. Introducing Anova and Ancova : A GLM Approach, SAGE Publications, Limited, London. Available from: ProQuest Ebook Central.

Saeys, Y., Abeel, T. & Van de Peer, Y. 2008. Robust Feature Selection Using Ensemble Feature Selection Techniques. In: Daelemans, W., Goethals, B., Morik, K. (eds) Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2008. Lecture Notes in Computer Science (), vol 5212. Springer, Berlin, Heidelberg.

Sahin, Y., Bulkan, S. & Duman, E. 2013. A cost-sensitive decision tree approach for fraud detection. Expert Systems with Applications, Volume 40, Issue 15, Pages 5916-5923.

Seijo-Pardo, B., Bolón-Canedo, V., Porto-Díaz, I., & Alonso-Betanzos, A. 2015. Ensemble Feature Selection for Rankings of Features. 29-42.

Shao, C., Paynabar, K., Kim, T.H., Jin, J., Hu, S.J., Spicer, J.P., Wang, H. & Abell, J.A. 2013. Feature selection for manufacturing process monitoring using cross-validation. Journal of Manufacturing Systems, Volume 32, Issue 4. Pages 550-555.

Subbiah, S. & Chinnappan, J. 2021. Opportunities and challenges of feature selection methods for high dimensional data: A review. Ingénierie des Systèmes d'Information, Vol. 26, No. 1, pp. 67-77.

Synthetic Minority Over-sampling Technique (SMOTE) method for MATLAB. https://se.mathworks.com/matlabcentral/fileexchange/75401-synthetic-minority-over-sampling-technique-smote

Tadist, K., Najah, S., Nikolov, N., Mrabti, F. & Zahi, A. 2019. Feature selection methods and genomic big data: a systematic review. Journal of Big Data 6, Article number 79.

Taiwanese Bankruptcy Prediction. 2020. UCI Machine Learning Repository. https://doi.org/10.24432/C5004D.

Tang, J., Alelyani, S., & Liu, H. 2014. Feature selection for classification: A review. In Data Classification: Algorithms and Applications, pp. 37–64.

Tsymbal, A., Pechenizkiy, M. & Cunningham, P. 2004. Diversity in search strategies for ensemble feature selection. Information Fusion, Volume 6, Issue 1, pages 83-98.

Venkatesh, B. & Anuradha, J. (2019). A Review of Feature Selection and Its Methods. Cybernetics and Information Technologies. 3-26.

Wolberg, W., Mangasarian, O., Street, N. & Street,W. 1995. Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository. https://doi.org/10.24432/C5DW2B.

Yan, Y., Shen, H., Liu, G., Ma, Z., Gao, C. & Sebe, N. 2014. GLocal tells you more: Coupling GLocal structural for feature selection with sparsity for image and video classification. Computer Vision and Image Understanding, Volume 124, Pages 99-109.

Zhou Y, Kantarcioglu M. 2020. On Transparency of Machine Learning Models: A Position Paper, in AI for Social Good Workshop.

Zhou, Z. 2021. Machine Learning. Springer Nature.