Vrije Universiteit Amsterdam

Bachelor Thesis

Exploring the Efficacy of Different Machine Learning Techniques Across Diverse Classification Tasks

Author: R.P.M. Kras (2697196)

1st supervisor: Wan Fokkink
2nd reader: Natalia Silvis-Cividjian

A thesis submitted in fulfilment of the requirements of the VU Bachelor of Science degree in Computer Science

30 November 2023

ABSTRACT

This thesis aims to deliver an analysis of the efficacy of three popular machine learning techniques in classification tasks. The three techniques that will be compared are Random Forest, Support Vector Machine, and Neural Network. The aim of this thesis is to contribute to existing knowledge in the field in order to validate or challenge the findings of existing research on the topic. The implementations of these techniques will be described, discussed, and reviewed. Among the three methods that were analysed, the Random Forest and the Support Vector Machine model performed the best, partly due to being trained and tested on the most extensive dataset among the three. The models were evaluated based on accuracy, F1-score, recall, precision, their ROC curve, their AUC value, but also cross-evaluation metrics including mean accuracy and standard deviation. The test results imply that the Support Vector Machine vector machine had the best overall performance, whereas the Random Forest had the highest stability. The Neural Network its performance was severely lacking due to the nature of the task as well as the given dataset, thus resulting in no useful insight. Further research is recommended to explore enhanced optimization techniques for the Random Forest and Support Vector Machine. Using more extensive and less biased datasets when evaluating both Random Forest and Support Vector Machine vector machine would also grant deeper insight, while the Neural Network would benefit the most from different application altogether.

Keywords: *Comparison, Analysis, Random Forest, Support Vector Machine Vector Machine, Neural Network*

1 Introduction

In the modern world artificial intelligence is growing increasingly important as automation and access to AI has become higher than ever. With the introduction of quick-developing, easily accessible AI tools available to the general public such as ChatGPT, the popularity of these tools has skyrocketed[1]. The concept of training a model, known amongst field members as machine learning, is defined by IBM as teaching a model through imitating the way that humans learn, gradually improving its task accuracy[2]. One of the implementations of machine learning is classification, which is the concept of using artificial intelligence to predict an outcome, often as a label, through training the model on a dataset of features. Classification is a supervised machine learning technique, which means that user feedback is a necessity for it to operate correctly. Moreover, classification is widely applicable for various purposes in several fields. Some of its purposes include decision making, information retrieval, medical diagnosis, risk assessment, personalization of customer experience, among others. The importance of classification lies in its ability to reveal relationships and connections between data points that were previously unseen[3]. This underscores the undeniable value of creating high-end classification models.

This paper focuses on three machine learning algorithms used for classification. These techniques are Random Forest, Support Vector Machines, and Neural Networks. First, the Random Forest is a supervised machine learning algorithm that is used in both classification and regression. It is a support tool that uses a tree-like model of decisions and their consequences, including chance event outcomes. In Random Forest, a technique called bagging is used to counter trees being too similar and losing effectiveness, thereby reducing functionality of the tree. Through this technique the trees are grown independently on random samples of the features, allowing for comprehensive exploration of the model space. This technique works by using several smaller tree models to classify an input. Second, Support Vector Machine is a supervised machine learning algorithm that is used in classification, regression, and outlier detection. At its most abstract level, it works by deploying an algorithm that maximizes a particular mathematical function with respect to a given collection of data (Noble, 2006). It is able to classify an input through utilizing this mathematical function. Third, a Neural Network is a technical term for a model that is inspired by the sensory processing of the brain. According to Krogh's definition (2008), learning in a Neural

[1] https://nerdynav.com/chatgpt-statistics/.

[2] https://www.ibm.com/topics/machine-learning

[3] https://www.sciencelearn.org.nz/videos/2065-classifying-and-identifying

Network is replicated through simulating a network of model neurons in a computer. In practice, a Neural Network receives an input and assigns a weight to it. Depending on this assigned weight and a pre-set threshold, it will classify the input. The process can be both supervised and unsupervised depending on whether the desired output is already known[4].

The choice of these three classification algorithms was made because of three important reasons. Firstly, these algorithms exhibit high versatility for handling classification tasks: all these algorithms can manage multi-class classification problems, thus making them applicable to various domains. Secondly, the Random Forest and Support Vector Machine algorithms are inherently robust against overfitting. While this is not the case for Neural Networks, they can be regulated and controlled using techniques like dropout or early stopping, allowing them to be designed to mitigate overfitting. Lastly, Neural Networks and Random Forests are known for their potential to achieve high predictive accuracy, while Support Vector Machines are more effective when appropriate kernel functions are selected. A high predictive accuracy grants important insight of the data. As each of these models have different characteristics and underlying theoretical foundations, an analysis of these grants insight as to how they operate on different datasets. In real-world scenarios, it is easy to understand why scoring well on performance metrics is incredibly important.

[4] https://www.nnwj.de/supervised-unsupervised.html

2 Literature review

In the literature review the existing theoretical foundation of Random Forests, Support Vector Machines, and Neural Networks will be divided into key fundamental principles and explained.

2.1 Random Forest

As aforementioned, at its highest-level the Random Forest algorithm can be interpreted as combining multiple tree models that classify an input based on splits and chance and will then assign the input with an outcome label. The theoretical foundations of Random Forests can be understood through several key concepts and principles.

2.1.1 Decision trees

The most important concept of the Random Forest method is the decision tree. Decision trees are a series of sequential models that employ a logical combination of successive tests. Every test entails the comparison of either a numeric attribute with a threshold value or a nominal attribute with a set of potential values (Kotsiantis, 2013). They recursively split the dataset into subsets based on the most prominent features, aiming to create a set of decision rules that lead to accurate predictions. All nodes represent a question, test, or decision about a feature, and all nodes point to a child node based on the possible outcomes of the question, test, or decision (Kingsford et al., 2008). Subsequently, based on the outcome of the answers to the questions, the input is classified into any of the output classes. In Random Forest, multiple decision trees are generated where the combination of every result per decision tree results is one outcome.

2.1.2 Bootstrap Aggregating or Bagging

The second fundamental of the Random Forest method is called bootstrap aggregating. According to IBM, bootstrap aggregating is by definition a machine learning method that is used to reduce variance within a noisy dataset[5]. The algorithm was invented by Leo Breiman (1996), and it consists of three steps:

1. BOOTSTRAPPING: UTILIZING A TECHNIQUE THAT INVOLVES CREATING DIVERSE SAMPLES BY RANDOMLY SELECTING DATA POINTS FROM THE TRAINING DATASET WITH REPLACEMENT. THIS METHOD GENERATES MULTIPLE SUBSETS WHERE INSTANCES MIGHT BE REPEATED WITHIN EACH SAMPLE.

[5] https://www.ibm.com/topics/bagging

2. Parallel Training: The generated bootstrap samples are then independently and simultaneously trained using weak or base learners.
3. Aggregation: Finally, based on the task at hand (regression or classification), an aggregated prediction is derived by either averaging the predictions for regression or selecting the class with the highest number of votes in classification (known as majority voting).

This technique is a staple of the Random Forest model because it increases accuracy and stability while reducing overfitting, thus making the model more robust and able to perform well on new, unseen data.

2.1.3 Random Feature Selection

The third fundamental of the Random Forest method is called random feature selection. The random feature selection algorithm, also known as the random subspace method or feature bagging, is a process that helps prevent decision trees from overfitting. It does so by reducing the "weight" of dominant features that are limited in quantity. Random feature selection is a form of ensemble learning, where the primary objective is to enhance resulting accuracy by amalgamating multiple models rather than relying on a single model[6]. Bagging and random feature selection share similarities, particularly in their ability to both be applicability to small training sample sizes. In the random subspace method, the classifiers are constructed in random subspaces of the data feature space. Typically, these classifiers are ultimately combined by a simple majority voting mechanism in the final decision rule (Skurichina et al., 2002), thereby allowing the model to classify the input.

2.1.4 Ensemble Learning

The fourth fundamental of Random Forest is ensemble learning. Ensemble learning involves leveraging multiple machine learning algorithms to produce provisional, weak predictive results based on extracted features from several projections of data. These results are then fused with various voting mechanisms to produce better results than any constituent algorithm alone (Dong et al., 2020). In Random Forests, ensemble learning implies that multiple smaller decision trees based on subsets of features of the dataset are created, wherein the predictions of individual trees lead to a final decision through a majority vote. In the context of classification, ensemble learning offers higher accuracy, better performance, and reduces the risk of overfitting and underfitting.

[6] https://corporatefinanceinstitute.com/resources/data-science/ensemble-methods/

2.1.5 Out-of-Bag Error Estimation

The fifth and last fundamental of Random Forest is a performance metric called out-of-bag error estimation. Since every tree uses a bootstrap sample, some datapoints are left out of each training set. These datapoints are known as the out-of-bag samples and the out-of-bag error is determined by aggregating the predictions from these samples across all trees in the model. This measure estimates the model's accuracy, which serves as a validation measure.

2.2 Support Vector Machine

As aforementioned, the Support Vector Machine algorithm is an algorithm that, at its highest-level, works by deploying an algorithm that maximizes a particular mathematical function with respect to a given collection of data. Similar to the Random Forest algorithm, the Support Vector Machine algorithm can be grasped through several key concepts that form its underlying theoretical foundation.

2.2.1 Linear Separability

At the core of Support Vector Machine lies the concept of linear separability. This concept states that predicting a label of an unknown input is achievable by classifying data points as lying either below or above a line. From a mathematical perspective, this principle can be extended to higher dimensions, resulting in the formation of a line known as a hyperplane. This hyperplane essentially serves as the boundary that distinguishes between different classes (Suthaharan, S., 2016), suggesting binary classification if there are only two dimensions. For classification, Support Vector Machine can consist of a hyperplane if the plot has multiple dimensions, so multiple predictive labels.

2.2.2 Maximizing the Margin

The Support Vector Machine algorithm aims to find the hyperplane that maximizes the margin between the hyperplane and the two nearest datapoints, often referred to as support vectors. In theory, we assume that the larger the distance between the two groups of classes, the better the generalization error of the classifier will be (Bhavsar et al., 2012). Hence, maximizing this margin leads to building a more robust model capable of better classifying new inputs.

2.2.3 Hinge Loss

In Support Vector Machine, a performance metric called hinge loss is commonly used. The hinge loss function is a convex function that penalizes classification errors, and in turn encourages the model to find a hyperplane that maximizes the distance between the data point classes while minimizing the number of classification errors. Hinge loss is a very simple function, leading to reduced complexity and less computing being involved[7], which is always advantageous.

2.2.4 Soft Margin and Regularization

In theory, a straight line should be enough to separate data points in order to classify new data. However, many real-world datasets often cannot be separated cleanly and thus cannot be efficiently handled by the Support Vector Machine algorithm. To address this limitation, the concept of a soft margin is introduced. Soft margin allows for some misclassification in exchange for a wider margin. This allows for some data points to "push" their way through the margin of the separating hyperplane without messing up the final result (Noble, 2006). Although this somewhat fixes the problem, it would not be beneficial for the accuracy to allow too many datapoints to slip through. Hence, regularization of the number of data points that are allowed to slip through is necessary. In essence, the soft margin parameter specifies a trade-off between hyperplane violations and the size of the margin (Noble, 2006).

2.2.5 Kernel Trick

The kernel trick is one of the key concepts for the computation of classifying using Support Vector Machine. Essentially, it introduces another dimension that allows for classification of data points that would not be able to be linearly separated using a straight line. In practical terms, the kernel trick is a function that helps classify a two-dimensional input correctly by transforming it into a three-dimensional feature space. The concept of the kernel function is to facilitate operations in the input space rather than potentially dealing with high-dimensional spaces (Jakkula, 2006). Using this approach saves both time and space.

2.2.6 Statistical Learning Theory

Support Vector Machine is built on the statistical learning theory. The main goal of statistical learning theory is to provide a framework for studying the problem of inference, that is of gaining knowledge, making predictions, making decisions, or constructing models from a set

[7] https://www.baeldung.com/cs/hinge-loss-vs-logistic-loss

of data (Bousquet et al., 2004). At a high level, it can be interpreted as a machine learning framework that draws from statistics and is used to analyse and predict based on this data (Lei, 2017). It has many advantages, including scalability, accuracy, and flexibility.

2.3 Neural Network

The Neural Network is a method in artificial intelligence that learns through a way inspired by the human learning process. Of all three formerly discussed methods, Neural Networks are arguably the most complex. This is because Neural Networks are based on a combination of principles from mathematics, neuroscience, and computer science.

2.3.1 Artificial Neurons, Perceptron

The basic building block of Neural Networks are artificial neurons called perceptrons. A perceptron can be understood as a simple model of a biological neuron in an artificial Neural Network. In practice, the perceptron consists of an input layer, one or multiple hidden layers, and an output layer. Initially, the input values are multiplied by a weight. Then, in the hidden layer, the weight is altered through non-linear transformations of the input values[8]. Finally, this resulting weight will be entered in an activation function that will determine the output. Perceptrons can be multi-layered, which means that all nodes are connected to all previous and next nodes in the perceptron. The advantage of adding multiple hidden layers to the perceptron is that these help account for non-linear relations between the input and output. Through the careful choice of appropriate connecting weights and transfer functions, it has been demonstrated that a multilayer perceptron can approximate any continuous and measurable function that maps from input vectors to output vectors (Hornik et al., 1989). In essence, learning in a perceptron occurs by weighing the input values, undergoing transformations within hidden layers, and culminating in the application of the activation function to determine the output.

2.3.2 Feed-Forward Architecture

The feed-forward architecture is a simple concept: information will flow in a singular direction only. The network structure of a feed-forward Neural Network is not fixed and can be adjusted in various ways, which makes it very flexible and applicable to any classification task (Kisner et al., 2022). Considering perceptrons, this architecture implies that information flows from the input layer, passes through the hidden layers, and then in the end reaches the

[8] https://deepai.org/machine-learning-glossary-and-terms/hidden-layer-machine-learning

output layer. The advantage of this is that it allows for complex non-linear functions to be modelled, because this architecture in combination with hidden layers enables the structuring of intricate relationships.

2.3.3 Universal Approximation Theorem

The material that Neural Networks can learn is limited. The approximation theorem asserts that a sufficiently large shallow Neural Network, typically consisting of only two layers, can effectively approximate a continuous function within a bounded domain (Lu et al., 2020). This theorem underscores the expressive power of a Neural Network and suggests that a Neural Network with one or many hidden layers can approximate any continuous function.

2.3.4 Backpropagation

The backpropagation algorithm is employed for optimizing Neural Networks by calculating how small adjustments in the strength of each synapse would impact the network's error (Lillicrap et al., 2020). The backpropagation algorithm is rooted in calculus and the chain rule, allowing for the computing of gradients that guide the adjustment of the network's weights to minimize the error function. This forms the basis for supervised learning in Neural Networks.

2.3.5 Loss Functions

Optimization of machine learning algorithms is crucial as low accuracy leads to low value. Therefore, applying optimization techniques is essential. The optimization of Neural Networks happens through the minimization of the loss function, where loss quantifies the difference between the predicted label by the model and the true label. A common loss function for optimizing classification tasks is the cross-entropy function.

2.3.6 Activation Functions

Activation functions play a crucial role in the learning process of the Neural Network. As previously mentioned, Neural Networks are made up of several layers: the input layer, hidden layer(s), and the output layer. The output layer determines the output based on the result of the weight on the activation function. In the absence of an activation function, the output signal would manifest as a simple linear function, essentially a polynomial of degree 1 (Sharma, 2020). This is impractical, because real world problems often require the Neural Network to be able to handle non-linear inputs. Common examples of activation functions to introduce nonlinearity include ReLU, sigmoid, and tanh.

2.3.7 Stochastic Gradient Descent

The stochastic gradient descent algorithm optimizes machine learning models by following the negative gradient of the objective function. Operating on a small subset of training samples at each step, stochastic gradient descent efficiently navigates the parameter space. It addresses two main issues: calculating the gradient of an entire dataset is costly, and if datasets are too big to fit in main memory training can become very slow[9]. Stochastic gradient descent is crucial for weight updating during the training of a model, and therefore it is essential for its practical success.

2.3.8 Regularization Techniques

The benefit of incorporating regularization into Neural Networks is to counter overfitting and to improve generalization. Regularization techniques commonly used in Neural Networks include dropout, L1 and L2 regularization, early stopping, and batch normalization: dropout is when some hidden layer outputs are ignored or dropped at random, L1 regularization is when a penalty term is added to the loss function based on the L1 norm of the coefficient vector[10], L2 regularization is when a small percentage of weights is removed at each iteration, early stopping entails stopping the training before overfitting occurs, and batch normalization is normalizing of (often smaller) batches of data within the hidden layers.

2.3.9 Deep Learning

The concept of deep learning is the concept of Neural Networks consisting of multiple hidden layers. Deep learning enables computational models consisting of multiple layers of processing to acquire representations of data with varying levels of abstraction (LeCun et al., 2015). This results in the ability of models to be able to process data with more abstract features, leading to an improvement in the network's capacity to solve complex problems.

2.3.10 Convolutional and Recurrent Architectures

Convolutional and recurrent Neural Network architectures are two different specialized approaches for building Neural Networks. A convolutional network is a network specialized in image data, and a recurrent network is specialized in sequential data. The convolutional Neural Network is designed to leverage the inherent structure in data, such as spatial relationships for images, and temporal dependencies in sequences. The recurrent Neural

[9] http://deeplearning.stanford.edu/tutorial/supervised/OptimizationStochasticGradientDescent/

[10] https://www.collimator.ai/reference-guides/what-is-l1-regularization

Network uses recurrent layers with loops to capture information from previous steps, allowing for sequential dependencies to be modelled. They specialise in sequential data, meaning that these are used for handling time series forecasting, natural language processing, or any application where the order and context of data matters.

2.3.11 Universal Function Approximators

The universal approximation theorem states that Neural Networks can be used to approximate any continuous function to arbitrary accuracy if no constraint is placed on the width and depth of the hidden layers. Neural Networks are considered universal function approximators that are capable of approximating a wide range of functions. This concept denotes their versatility in handling diverse data types and sets and solving machine learning problems.

3 Data Collection and Pre-Processing

Not all machine learning models are equally a good fit for all data types. To perform a comparative analysis of the three selected models, all three models will each be tested with a dataset that is well-suited for them. These datasets were retrieved from Kaggle, which is a data science-oriented community website and a subsidiary of Google. All datasets are publicly available for learning, research, and application purposes, and are available in the appendix.

3.1 Dataset 1, heart disease

3.1.1 Background

According to the CDC[11], heart disease is a leading source of death for most people in the U.S. About half of all Americans (47%) have at least 1 of 3 major risk factors for heart disease: high blood pressure, high cholesterol, and smoking.

The first dataset originated from the CDC, which is part of a governmental organisation that monitors U.S. citizens' health. The data contains a total of seventeen distinctive features, together with an indicator whether these features correspond to heart disease, which allows for training of the model. The sheet is extensive, as it contains information of approximately 320.000 U.S. citizens aged sixty and older.

The sheet is well-organized, and it contains categorical data: smoking, alcohol, stroke, difficulty walking, sex, race, diabetic, physical health on a scale of 0 to 30, physical activity, mental health on a scale of 0 to 30, general health, age category, asthma, kidney disease, skin cancer, and numerical data: BMI and hours of sleep per night.

The first dataset is limited in the fact that this data was retrieved from Kaggle, therefore the dataset does not have a verifiable source and is not as trustworthy as it would have been if it were to be retrieved from the CDC directly. Although this is less relevant for this research as the main focus lies on the application and comparison of the machine learning techniques, it is still important to point out.

[11] https://www.cdc.gov/heartdisease/risk_factors.htm

3.1.2 Pre-processing

The first dataset data is well-maintained. Hence, the only step that had to be taken for the dataset to be suitable for the model was to encode the categorical labels. This is later explained in the methodology.

3.1.3 Data Exploration and Visualization

For the first dataset, visualization shows that the surplus of recorded stats is of citizens with a healthy BMI of approximately 22.5, that have good physical and mental health, and also get an average of 8.5 hours of sleep per night, as is displayed in figure 1.

In figure 2 it is shown that of the 320.000 U.S. citizens in this dataset, about 40% smoke, while the remaining 60% do not. Approximately 10% drink alcohol regularly, whereas 90% do not. Regarding history of stroke, only about 5% have experienced a stroke before. In total only about 15 to 20% of the citizens have difficulty walking.

The dataset is not equally divided with about 45% male, and 55% female. About 75% of recorded stats are of Caucasian, with about 10% Hispanic, 10% African American, 5% other, 5% Asian, and 5% American Indian or Alaskan Native.

Approximately 80% are not diabetic, where of the remaining 20%, 15% have diabetes, with 4% that are borderline diabetic, and only 1% have experienced it temporarily during pregnancy.

The vast majority are regularly active, and only about a quarter is inactive. Regarding health conditions, about 15% have asthma, 5% have kidney disease, and about 10% have or have had skin cancer.

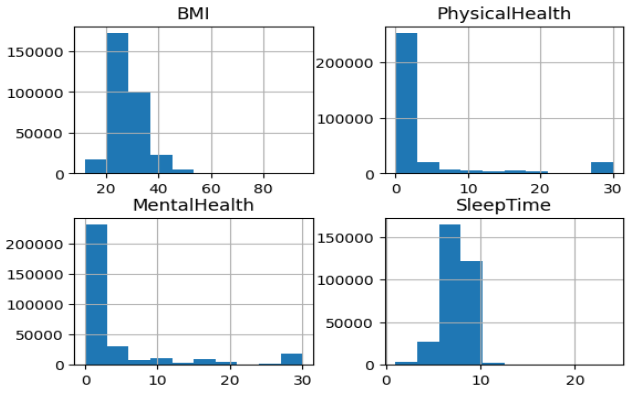


Figure 1, numerical data from “heart disease” dataset

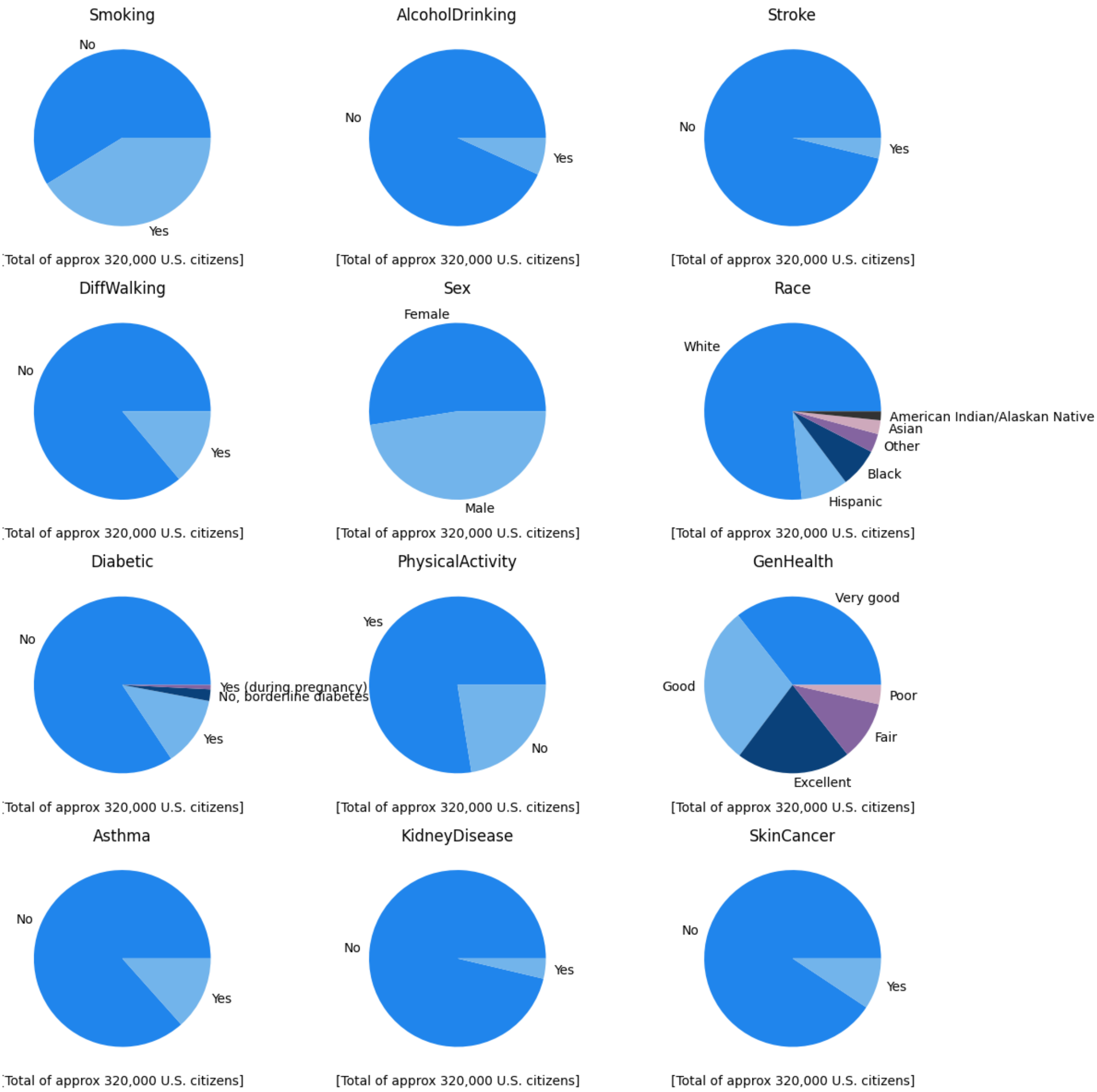


Figure 2, categorical data from “heart disease” dataset

3.1.4 Conclusion

To summarize, the first dataset was already in suitable condition upon extraction from Kaggle, therefore only minor adjustments had to be made to the dataset to prepare it for use in machine learning. The dataset contains records of approximately 320.000 U.S. senior-citizens and shows seventeen key indicators of health, such as general health, race, whether they smoke, whether they drink, and so forth. The dataset being in good condition serves as a beneficial starting point for the comparative analysis.

3.2 Dataset 2, the iris flower

3.2.1 Background

The second dataset called the iris flower dataset is a dataset introduced by the British statistician and biologist Ronald Fisher in his 1936 paper “The Use of Multiple Measurements in Taxonomic Problems”. The reasoning behind it is because Edgar Anderson collected the data to quantify morphologic variation of iris flowers of three related species. The dataset consists of 50 samples from each of three species of Iris (Iris Setosa, Iris Virginica, and Iris Versicolor). It contains 4 features: the length and the width of the sepals and petals, all in centimetres.

3.2.2 Pre-processing

The second dataset is well-documented, clean, and well-maintained. The only pre-processing that has to be performed is the encoding of the target column, ‘species’.

3.2.3 Data Exploration and Visualization

The second dataset is small, with only 150 samples. In figure 3, all recorded values are displayed for the 150 samples. The data of the length of the sepal is left-skewed to an average of 5 to 6 centimetres, whereas the data of the width of the sepal is recorded at around 3 centimetres. The data of the petal width and length are mostly close to 0, but also show an increase at around 4 to 5 centimetres for length and 1.5 to 2 for width.

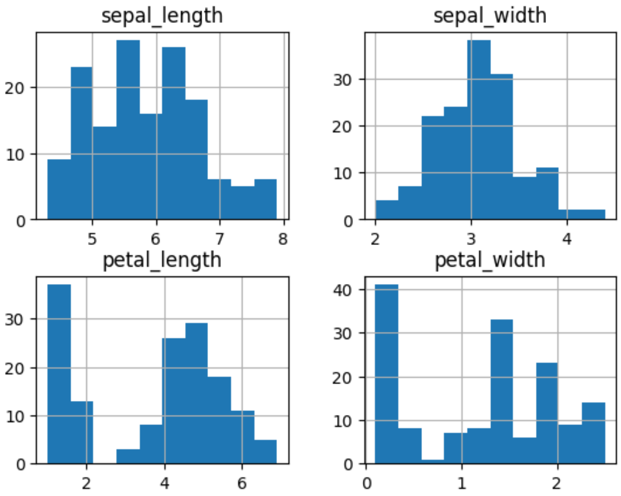


Figure 3, numerical data from Iris dataset

3.2.4 Conclusion

The second dataset contains 150 samples with four features corresponding to three species of Iris: Iris Setosa, Iris Virginica, and Iris Versicolor. These four features are length and width of the sepals and petals. The data was already clean; therefore, no data cleaning or pre-processing was necessary except for the encoding of the categorical target label.

3.3 Dataset 3, historic AAPL stock data

3.3.1 Background

The third and final dataset is tricky. This dataset contains historic stock data from all US-based stocks on the NYSE, NASDAQ, and NYSE MKT. For this thesis, the Apple stock has been selected. The AAPL stock dataset contains historic data from between 1980 and 2022, where historic data implies the values of the stock's low, open, high, close, and adjusted close values, as well as its traded volume over a period of time. This dataset is important because predicting the volatile stock market has always been an appealing challenge in the machine learning industry as it is very lucrative.

3.3.2 Pre-processing

The last dataset is well-documented, containing data of over 10.000 days from 1980 to 2022. Although there was a substantial amount of data missing of multiple dates, it still suffices for its intended use in this practical analysis. It is however regretful and important to point out that data from the stock market is often time inaccessible or hard to retrieve, leading to the inability to retrieve data from the missing dates. For the stock data to be suited for classification, a new column is added to the sheet based on the adjusted close. This new column is created by comparing the adjusted close of the next date to the adjusted close of the previous date: if the new adjusted close is higher, then the new column will display a one. Similarly, if it is lower, then the new column will display a zero.

3.3.3 Data Exploration and Visualization

The third dataset shows the historical closing price of the US-based AAPL stock. As is visible in figure 4, it started with a value close to zero, then started increasing in 2005 until 2020 to 175 USD, but as of 2022 has a value of approximately 150 USD.

3.3.4 Conclusion

The third dataset contains the historic stock data of AAPL ranging from 1980 to 2022. Although a substantial portion of data is missing, the dataset still follows the general direction of the stock price, therefore making it suitable for use in machine learning. The dataset shows that the AAPL stock value was close to zero in 1980, then it started increasing from 2005 until 2020, where it remained roughly the same from 2020 to 2022.

3.4 Motivation

Picking a ‘well-suited’ dataset for each model is challenging as most models are able to handle various types of data quite easily. Therefore, the rationale behind each dataset choice for each model is as follows:

- The Random Forest benefits from a dataset with mixed data types, as Random Forests are known to be exceptional at handling both categoric and numeric data. Furthermore, Random Forests are also efficient at handling complex relationships in data. Hence, a dataset containing mixed data types is used.
- For the Support Vector Machine, SVMs benefit from a dataset with numeric data as SVMs focus on finding the most optimal hyperplane in a numeric space. Moreover, SVMs are also very efficient high-dimensional space. Hence, a dataset containing numeric data is suitable.
- As for the Neural Network, Neural Networks are exceptional at capturing temporal dependencies within sequential data. In addition, Neural Networks excel at learning complex patterns in datasets. Therefore, a dataset containing historic stock data is suited for this model.

4 Methodology

The purpose of the methodology is to provide a detailed account of the implementation of the machine learning models for Random Forest, SVM, and Neural Network, all implemented using Scikit-Learn. All three models were tested using a testing size of 0.2 or 20%, in order to minimize the risk of overfitting during evaluation.

4.1 Random Forest

For the Random Forest model, the first step is importing necessary libraries:

- In order to handle the heart disease dataset, the model utilizes the Pandas library.
- ‘LabelEncoder’ from ‘sklearn.processing’ encodes categorical features.
- ‘train_test_split’ from ‘sklearn.model_selection’ divides the data into training and testing data. The parameter of 0.2 implies that 20% of the data is allocated for testing purposes.
- ‘accuracy_score’, ‘f1_score’, and ‘confusion_matrix’ from ‘sklearn.metrics’ are all evaluation metrics for assessing machine learning models.
- ‘RandomForestClassifier’ from ‘sklearn.ensemble’ serves as the classifier for the Random Forest model.

The data was read through Pandas’ ‘read_csv’ function. Next, in order to prepare the model for training, the feature columns and the result column had to be defined as variables. All independent variables are assigned to the variable ‘X’, whereas the prediction value is assigned to the variable ‘y’.

Given that the majority of the dataset is categorical data, a label encoder was used to encode the categorical labels into numerical values, suitable for the model’s use. The label encoder transforms categorical labels such as ‘happy’, ‘sad’, and ‘neutral’, into numerical values of 0s, 1s, and 2s.

Finally, a train test split was used to divide the dataset into training and testing data. The test size of 0.2 or 20% was used for all three models regardless of data size, and the split was executed with random state set to 1. After this split, the data was fitted to the model, and the model was trained.

4.2 Support Vector Machine

For the Support Vector Machine, the iris flower dataset was deemed the best fit. Similar to the Random Forest model implementation, libraries have to be imported first:

- In order to handle the iris flower dataset, the model utilizes the Pandas library.
- ‘LabelEncoder’, ‘train_test_split’, ‘accuracy_score’, and ‘svm’ were all imported from ‘sklearn’ and serve the same purpose as in the Random Forest model.

First, the data is read through Pandas. Then, the columns are defined by assigning all independent numerical values of petal and sepal width and length to ‘X’, and the predicted species type to ‘y’. The train test split is again of 0.2 or 20%, with random state of 1. Finally, the data was fitted to the model, and the model was trained.

4.3 Neural Network

For the Neural Network, the historical stock value dataset of AAPL (Apple Inc.) was used. Imported libraries:

- In order to handle and manipulate the data, the model utilizes the Pandas library.
- ‘train_test_split’ and ‘accuracy_score’ are used similarly to previous models.
- ‘LogisticRegression’ from ‘sklearn.linear_model’ is used for logistic regression.

Initially, the data is read through Pandas. Then, feature columns and a prediction column are defined as ‘X’ and ‘y’ respectively. The feature columns are used to make the prediction of an increase or decrease of the stock its value. Again, the train test split is set to 0.2, with random state set to 1, where 20% of the data will be used for testing and the remaining 80% for training. In this case, that means the data of roughly 8.000 dates are used for training purposes.

The model also makes use of a lagged variable, specifically the adjusted closing price was lagged. The rows containing NaN-values as a consequence of the lagged variable are removed, and the corresponding indices are updated with the cleaned values. Logistic regression is a common choice for classification using a Neural Network model.

5 Results and analysis

The three different classification models are each trained using the datasets described in chapter 3, “Data Collection and Pre-processing”. The models were implemented using the Scikit-Learn library, which allows for swift instantiating of the models. The Random Forest model was trained using a dataset containing both categorical and numerical features, the Support Vector Machine was trained using a dataset containing only numerical features with three different target labels, and the Neural Network was trained using numeric historic stock data.

All models are evaluated using common performance metrics such as accuracy, f1-score, recall, and precision. In order to evaluate performance across the three models, cross-validation scores such as mean accuracy and standard deviation are also computed.

Accuracy, calculated by dividing the number of correct predictions by the number of total predictions, is a crucial metric in classification tasks. However, in real-world scenarios this score is often less relevant than recall because it is highly important for a model to correctly classify negative instances instead. As an example, think of a classification model that classifies heart disease in patients.

F1-score, akin to accuracy, assesses the performance of a model but is more reliable than accuracy when data is unbalanced. This is particularly relevant in the Neural Network trained on historical stock data.

As mentioned earlier, recall is very important for the Random Forest model. The recall is calculated by dividing the correctly classified positive labels by the sum of the correctly classified positives and falsely classified negatives.

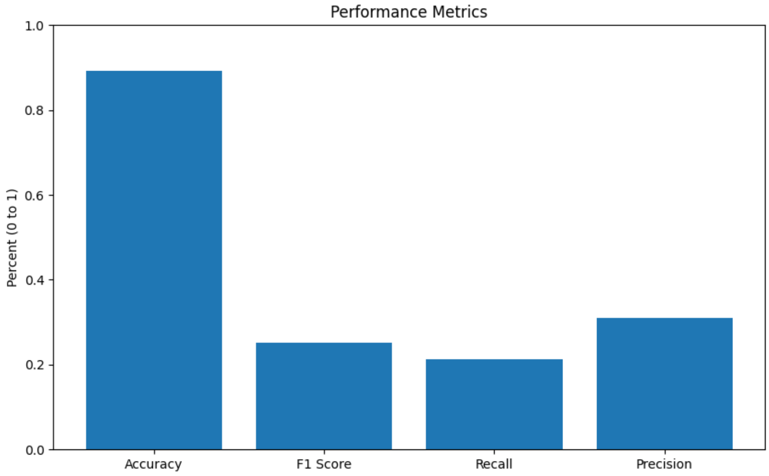
On the other hand, the precision of a model is the number of times the model has correctly positively classified an input. This is particularly relevant to the Neural Network, as a high precision implies that the model predicts stock directions accurately.

The cross-validation scores are used to quantify performance of all three models despite being trained on different datasets. The mean accuracy is by definition a measure of overall performance which is calculated as the average of accuracy scores obtained from multiple rounds of cross-validation. In this case, cross-validation has been applied 5 times on each model. The standard deviation is a measure of spread or variability in model performance. This value denotes the consistency or stability of a model’s predictions.

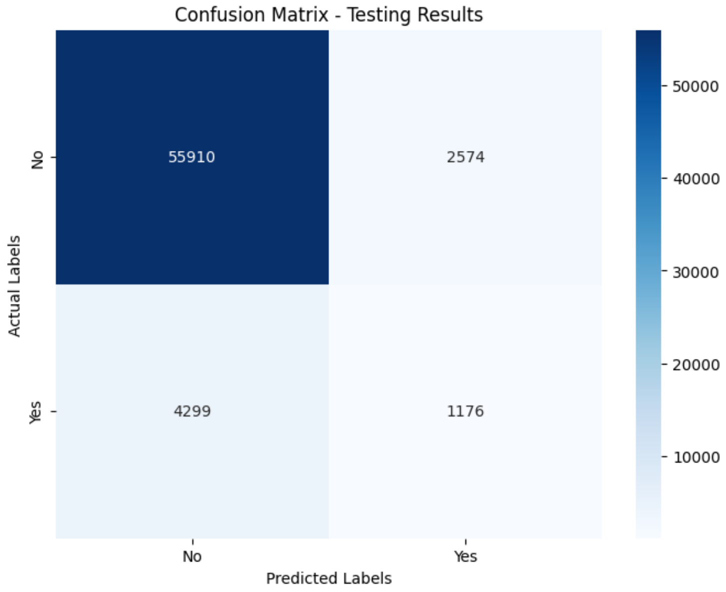
The Random Forest model’s performance can also be evaluated using the Received Operating Characteristic (ROC) curve and the Area Under the Curve (AUC). The ROC curve provides deeper insight into the model its discriminatory ability, particularly in balancing true positives against false positives across various thresholds, and the AUC quantifies overall model performance.

5.1 Random Forest

The Random Forest model tested with input of the heart disease dataset, implemented as described in Methodology results in:



Model 1, performance metrics Random Forest



Model 1.1, confusion matrix Random Forest

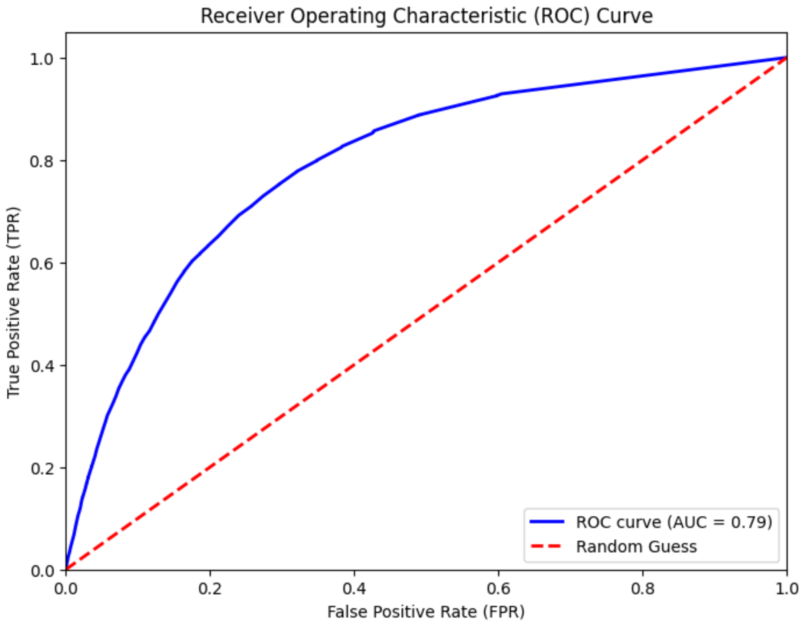
The metrics are displayed in the histogram of model 1:

- The accuracy reaches a value of approximately 89.21%, which suggests that the majority of the predictions made by the SVM are correct.
- The precision of the model is approximately 31.01%, which implies relatively bad performance.
- The recall achieved a score of 21.2%, which implies that the model is ineffective at capturing the positive instances out of all people with heart disease.
- The f1-score achieved a score of approximately 25.19%, which indicates bad performance overall.

From the results of the testing that are displayed in model 1.2, we can infer that it is likely that the model has underfitted the dataset, meaning that it was unable to find the actual underlying patterns in the data. The reason for this is because the model has a very easy time correctly classifying people as not having heart disease, while the reverse, classifying people with heart disease correctly, is not true. Moreover, from the confusion matrix we can infer that the dataset or testing set is heavily skewed towards people that do not have heart disease, meaning that the classes are very imbalanced. This leads to bias, from which we can deduce that this has likely impacted these results.

The mean accuracy and standard deviation, both computed through cross evaluation, stand at 0.9064 and 0.0006. A high mean accuracy of 0.9064 implies that the Random Forest model performs well on unseen data. The low standard deviation implies that the Random Forest model its performance is stable and robust. It will not produce highly variable results.

With a recall score of 21%, it is implied that this model has very bad performance in correctly identifying patients with heart disease. Contextually, this score on its own already suggests that this model is unsuitable for real-world use. Moreover, the accuracy is unreliable, as the confusion matrix suggests that the majority of correct predictions are made by classifying patients without heart disease, of which there are more present in the dataset.

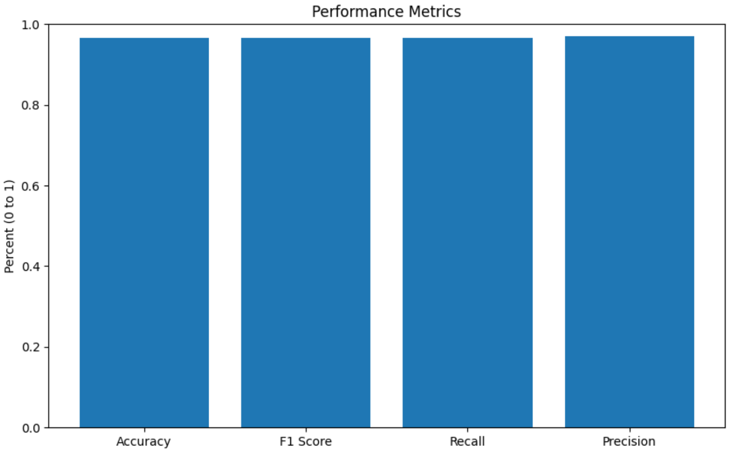


Model 1.2, ROC Curve of Random Forest

The bell-shaped ROC curve suggests that the model has good discrimination at different classification thresholds. To further support this observation, the AUC value of 0.79 implies that the model has reasonably effective performance in distinguishing between positive and negative instances of heart disease. Contextually, the ROC curve showcases the trade-off between correctly identifying patients with heart disease and minimizes false diagnoses.

5.2 Support Vector Machine

The Support Vector Machine tested with input of the iris flower dataset, implemented as described in Methodology results in:



Model 2, performance metrics Support Vector Machine

Due to the low number of samples in the dataset, the values of accuracy, f1-score, recall, and precision are quite similar:

- The accuracy reaches a value of approximately 96.67%, which suggests that the majority of the predictions made by the SVM are correct.
- The precision of the model is approximately 96.97%, which shows a high ratio of correctly predicted positive classifications to the total positive predictions.
- The recall achieved a score of 96.67%, which implies that the model is effective at capturing the positive instances.
- The f1-score achieved a score of approximately 96.65%, which also indicates good performance.

The Support Vector Machine model showed remarkable results across all chosen performance metrics. Hence, it is implied that the model is highly reliable for the task at hand. However, due to the size of the dataset, it is recommended to validate its performance on a bigger dataset for future research.

A 3x3 confusion matrix is interpreted similarly to a 2x2 matrix. Namely, in a 3x3 matrix all correctly classified instances are along the diagonal, where all other instances can be generalized as misclassification. In this case, the singular "1" value in the third row implies that it has been wrongly classified. The matrix is in 3x3 form due to the iris flower dataset having 3 different classes, thus making it a multilabel classification problem.

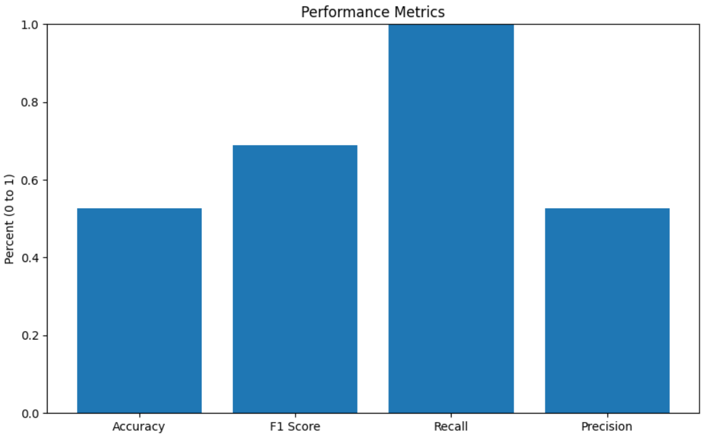
Model 2.1, confusion matrix Support Vector Machine

10	0	0
0	10	0
0	1	9

The mean accuracy and standard deviation of the Support Vector Machine model were 0.9583 and 0.0264 respectively. These values imply that the model is robust and reliable. However, it should be noted that the relatively small size of the dataset likely influenced this result.

5.3 Neural Network

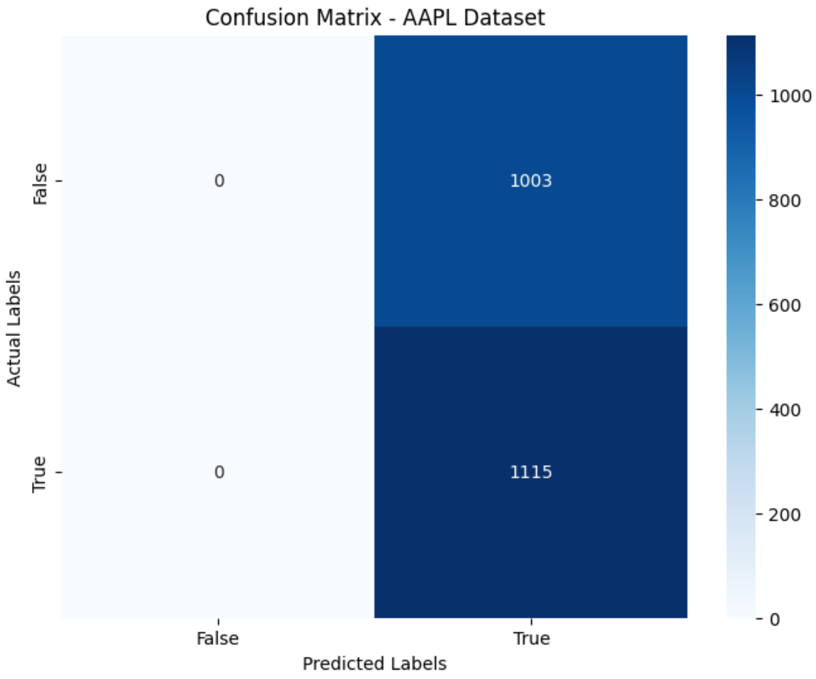
The Neural Network based on logistic regression, tested with the input of the historical stock data of AAPL, implemented as described in Methodology results in:



Model 3, performance metrics Neural Network

The metrics are displayed in the histogram of model 3:

- The accuracy of the model is approximately 53%, which suggests that the number of good predictions made by the Neural Network is very low, it can be considered equal to random guessing.
- The precision of the model is approximately 53%, which shows a very low ratio of correctly predicted positive classifications to the total positive predictions.
- The recall achieved a score of 100%, which is explained through the confusion matrix' its results. That is why contextually this score is unreliable.
- The f1-score achieved a score of approximately 69%, which indicates relatively good performance.



Model 3.1, confusion matrix Neural Network

This confusion matrix implies that the model has difficulty predicting downward movements of the AAPL stock value. Additionally, the model has been overly optimistic, with 1003 instances of wrongfully predicting an upward movement. In real-world context, the model is of low reliability. As is shown in figure 4, the dataset is highly imbalanced because the majority of the plot shows an upwards trend. Consequently, the model exhibits low accuracy and precision but a high, worthless recall score.

The mean accuracy and standard deviation were calculated at 0.4979 and 0.0109 respectively. These values indicate that the model performs poor on average in terms of accuracy. Despite the relatively low standard deviation, due to the mean accuracy we can still say that the model its performance is unreliable.

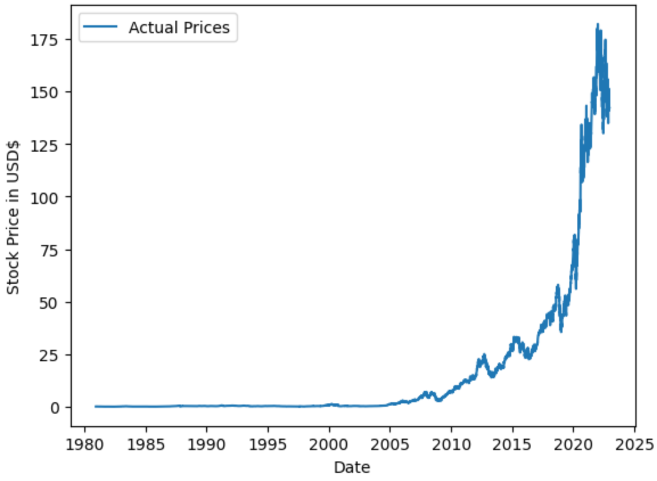
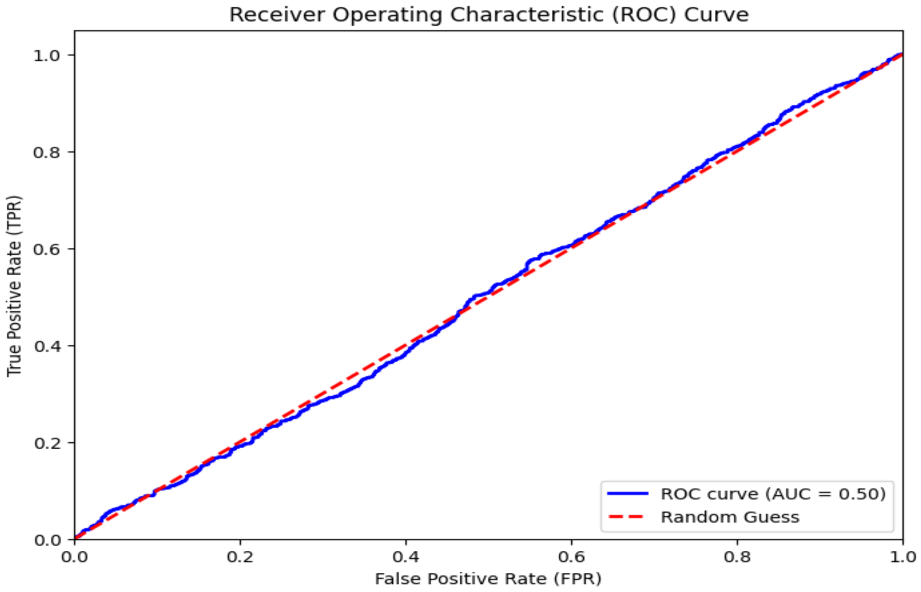


Figure 4, AAPL stock price



Model 3.2, ROC Curve of Neural Network

The ROC curve of the Neural Network highlights the inconsistency of the historic AAPL stock data as well as the low performance of the model. From the shape of the plot and the value of the AUC, it is implied that the Neural Network ranks a random positive example higher than a random negative example 50% of the time, thus indicating that the predictive ability of the model is worthless[12].

[12] https://developers.google.com/machine-learning/crash-course/classification/check-your-understanding-roc-and-auc

5.4 Analysis

To conduct a comparative analysis between the performance of the three models, the following 4 factors are important:

5.4.1 Overall performance

Based on the results for each model, it is evident that all models had relatively bad performance except for the Support Vector Machine, which is favoured due to its low number of samples during testing. None of the models, except for the SVM, excelled in any specific areas. Hence, from the performance metrics, it can be concluded that the SVM outperforms all other models.

5.4.2 Stability

The Support Vector Machine and the Random Forest model both exhibit remarkable stability. The Support Vector Machine reached an impressively high mean accuracy, while the Random Forest model had a similar mean accuracy with a notably higher standard deviation. Even though all models displayed a relatively low standard deviation, it is clear that the Random Forest model has the highest stability thanks to scoring achieving the best scores across both metrics.

5.4.3 Interpretability

Interpretability is a factor regarding gained insights from the models. Using the Neural Network, it is possible to forecast the future direction of the stock value. Such predictive insight is hard to gain from Random Forests and are even harder to obtain from Support Vector Machines. Consequently, the Neural Network stands out as the winner in this regard.

5.4.4 Domain Relevance

The domain relevance is a factor regarding how well the model aligns with the requirements and constraints of the domain. The Random Forest model scores relatively low in the tests but could be effective in real-life context with correct preparations and further perfecting of the model. The Support Vector Machine excels here by virtue of its simple task in classifying Iris species. On the contrary, the Neural Network had too many issues to be considered relevant given its domain.

6 Discussion

The Random Forest and the Support Vector Machine both have sufficient results for testing purposes. The Support Vector Machine achieved high accuracy scores with a good balance between precision and recall, whereas the Random Forest achieved high accuracy but bad scores for precision and recall. Their cross-evaluation scores are both relatively good. These scores imply that the methods are both effective for their tasks based on the chosen metrics. On the contrary, the Neural Network had relatively bad performance. While it has high recall, it falls short in terms of overall accuracy and precision. The results suggest that the Neural Network model using logistic regression may not have been the most suitable choice for the task at hand, and the Random Forest model requires further tuning for better results.

It is crucial to highlight that, as previously mentioned, the Random Forest model may be hard to use in the real world due to it having bad scores for precision and recall. The reason for this is because the data contains very limited samples of patients with heart disease, thus implying serious bias during model training. The Random Forest model does, theoretically, show high promise for its chosen classification task, as it was able to handle both categorical and numerical data easily. Therefore, for future research it is recommended to tune the parameters of the model or to balance the dataset to achieve better results. Similarly, the Support Vector Machine would also strongly benefit from a bigger dataset. Although its general performance is good, it is recommended to use a bigger dataset, as the size of the current dataset is relatively small and therefore causes the model to be less reliable. Regarding the Neural Network, it is advisable to add more variables during training or to employ a different machine learning technique altogether. The use of the Neural Network based on its chosen classification task is not recommended, due to the dataset's strong bias towards an increase in stock price, rendering the test results impractical and unreliable.

In practice, good overall performance and stability hold significant importance. The results of the analysis of the models suggests that all three selected techniques are able to handle their classification tasks to some extent. Considering the underlying framework as well as the performance metrics of the Random Forest model, it suggests that this model should adeptly handle both numerical and categorical data. Although the Support Vector Machine had overall better performance, the results it produced are less reliable due to the small dataset it has been trained on. Neural Networks are known for being efficient in detecting relations between predictor variables, however in this case, identifying stock movements proves exceptionally challenging due to lacking data as well as heavy bias in the dataset. Considering these limitations, future research is recommended due to the constraints of the AAPL dataset as per 2023, while the Random Forest and Support Vector Machine would benefit from enhanced optimization techniques as well as data manipulation in order to alleviate bias.

7 Conclusion

In conclusion, in the modern world artificial intelligence has gained a lot of positive traction. Due to this, it is important to highlight different machine learning techniques and compare them to each other in different applications and contexts. The aim of this thesis is to conduct an analysis of three different machine learning techniques on classification tasks in order to challenge or reaffirm existing research on the topic. The techniques discussed are Random Forest, Support Vector Machine, and Neural Network.

The Random Forest was trained and tested using a dataset of numerical and categorical features with a topic of heart disease. The Support Vector Machine was trained and tested using a dataset with numerical data with a target column of Iris species, of which three classes exist. The Neural Network was trained and tested using a dataset of historic stock data of the AAPL stock, and the purpose of this classification task was to predict future stock movements.

The Support Vector Machine exhibited superior overall performance, yet its reliability might be lower due to the small size of its training and testing set. The Random Forest demonstrated remarkable stability, reflected in its high cross-evaluation scores, even though this may have been influenced by the highly biased dataset it was trained and tested on. Both models delivered acceptable results for their respective classification task, proving their competence in their given domain. Notably, the Neural Network stands out in regard of interpretability, considering its ability to generate forecast plots, which offer a convenient way of conveying information. However, the Neural Network struggled with a heavily biased dataset, highlighting its limitation in its given domain. The Random Forest’s ROC curve and AUC suggest robust discriminative performance, although this may also have been influenced by its dataset, while the Neural Network’s ROC curve suggests predictions akin to random guessing, thereby indicating worthless predictive ability.

The findings of this research are inconclusive due to the training and testing of the models being done on limited and biased datasets, therefore producing unreliable results. For future research, it is recommended to use more expansive or balanced datasets, minimizing bias. This approach will likely contribute to producing more dependable and robust results during experimentation and analysis, especially for the Random Forest model and the Support Vector Machine. For the Neural Network it is recommended to use a different technique or to wait until more data becomes available in the future. Alternatively, changing the context in which the task was given to the Neural Network could also prove useful.

References

Noble, W. (2006). What is a Support Vector Machine? *Nature Biotechnology*, *24*, 1565–1567. https://doi.org/10.1038/nbt1206-1565

Krogh, A. (2008). What are artificial Neural Networks? *Nature Biotechnology*, *26*, 195–197. https://doi.org/10.1038/nbt1386

Kotsiantis, S.B. (2013). Decision trees: a recent overview. *Artificial Intelligence Review*, *39*, 261–283. https://doi.org/10.1007/s10462-011-9272-4

Kingsford, C., & Salzberg, S. (2008). What are decision trees? *Nature Biotechnology*, *26*, 1011–1013. https://doi.org/10.1038/nbt0908-1011

Breiman, L. (1996). Bagging Predictors. *Machine Learning*, *24*, 123-140. University of California. [PDF]. https://link.springer.com/content/pdf/10.1007/BF00058655.pdf

Skurichina, M., & Duin, R. (2002). Bagging, Boosting and the Random Subspace Method for Linear Classifiers. *Pattern Analysis and Applications*, *5*, 121–135. https://doi.org/10.1007/s100440200011

Dong, X., Yu, Z., Cao, W., et al. (2020). A survey on ensemble learning. *Frontiers of Computer Science*, *14*, 241–258. https://doi.org/10.1007/s11704-019-8208-z

Suthaharan, S. (2016). Support Vector Machine. In *Machine Learning Models and Algorithms for Big Data Classification* (Integrated Series in Information Systems, Vol. 36). Springer. https://doi.org/10.1007/978-1-4899-7641-3_9

Bhasavar, H., & Panchal, M.H. (2012). A Review on Support Vector Machine for Data Classification. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, *1*(10). https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=d683a971524a0d76382ce335321b4b8189bc8299

Jakkula, V. (2006). Tutorial on Support Vector Machine (SVM), p. 37. *School of Electrical Engineering and Computer Science, Washington State University.*

Bousquet, O., Boucheron, S., & Lugosi, G. (2004). Introduction to Statistical Learning Theory. In *Advanced Lectures on Machine Learning* (ML 2003). *Lecture Notes in Computer Science, 3176*. Springer. https://doi.org/10.1007/978-3-540-28650-9_8

Lei, Y. (2017). Individual Intelligent method-based fault diagnosis. *Intelligent Fault Diagnosis and Remaining Useful Life Prediction of Rotating Machinery*, *2017*, 67–174. https://doi.org/10.1016/b978-0-12-811534-3.00003-2

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*, 359–366.

Kisner, H., Ding, Y., & Thomas, U. (2022). Capacitive material detection with machine learning for robotic grasping applications. *Tactile Sensing, Skill Learning, and Robotic Dexterous Manipulation, 2022*, 59–79. https://doi.org/10.1016/b978-0-32-390445-2.00011-8

Lu, Y., & Lu, J. (2020). A universal approximation theorem of deep Neural Networks for expressing probability distributions. [Abstract]. In *NeurIPS 2020 Proceedings*. Retrieved from https://proceedings.neurips.cc/paper/2020/hash/2000f6325dfc4fc3201fc45ed01c7a5d-Abstract.html

Lillicrap, T.P., Santoro, A., Marris, L., et al. (2020). Backpropagation and the brain. *Nature Reviews Neuroscience*, *21*, 335–346. https://doi.org/10.1038/s41583-020-0277-3

Sharma, S. (Author1), Sharma, S. (Author2), & Athaiya, A. (2020). Activation Functions In Neural Networks. [PDF]. In *International Journal of Engineering and Applied Sciences Technology* (IJEASt), 10(2), 310-316. Retrieved from https://www.ijeast.com/papers/310-316,Tesma412,IJEAST.pdf

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436–444. https://doi.org/10.1038/nature14539

Appendix

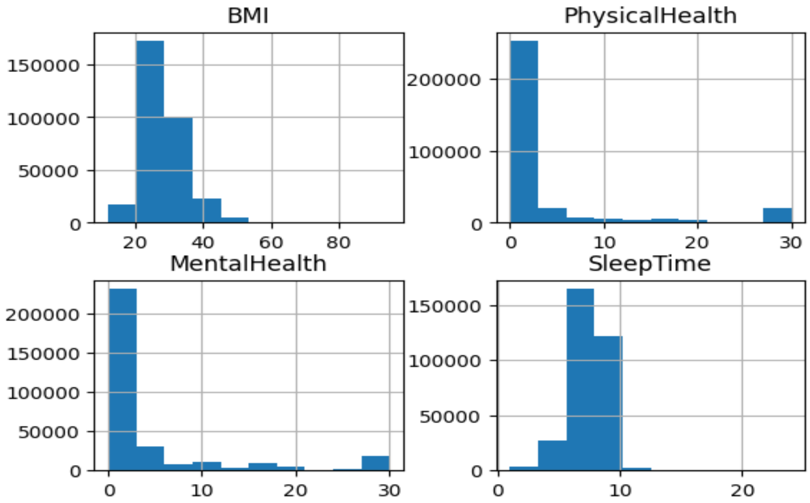


Figure 1, numerical data from "heart disease" dataset

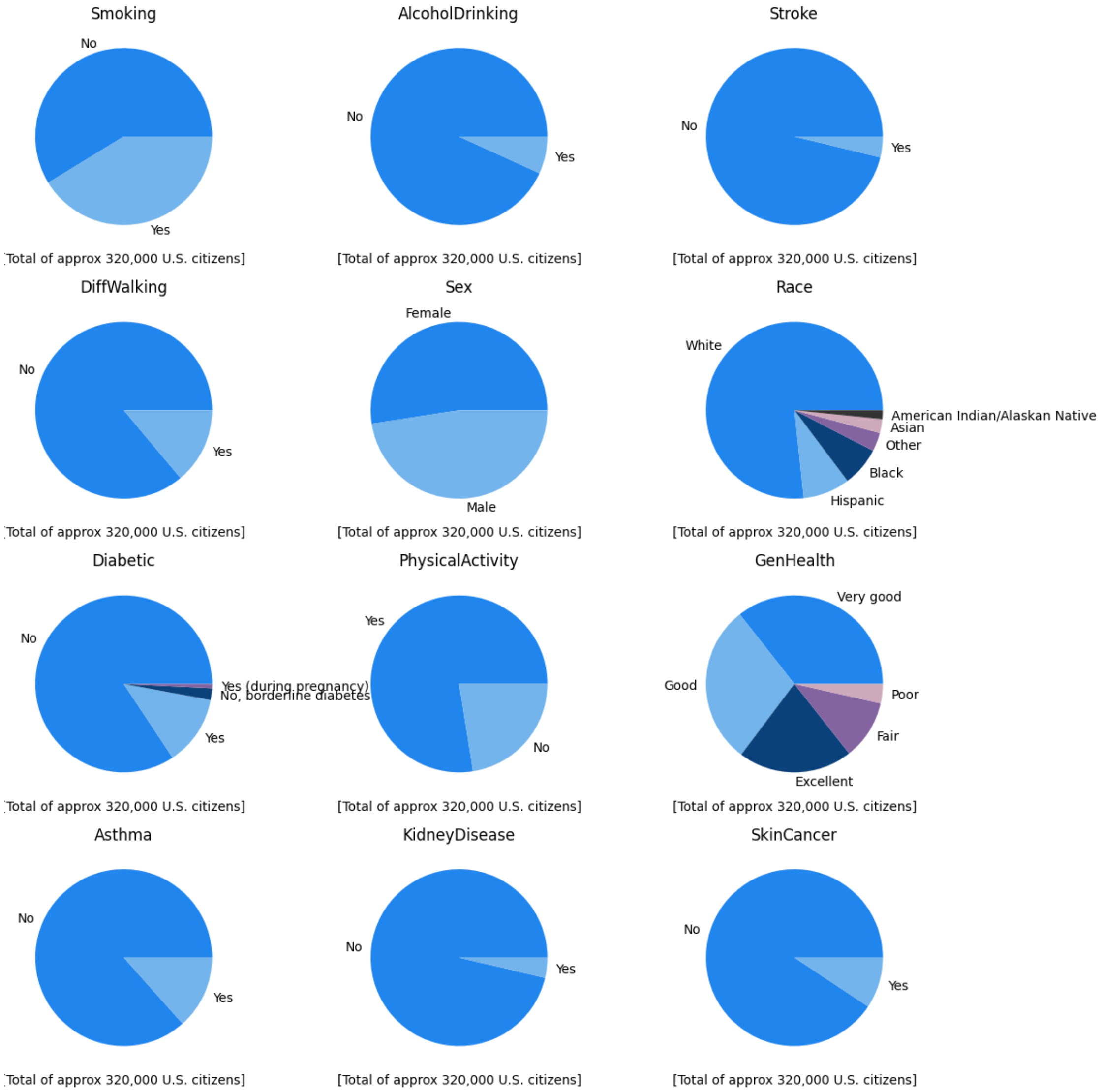


Figure 2, categorical data from "heart disease" dataset

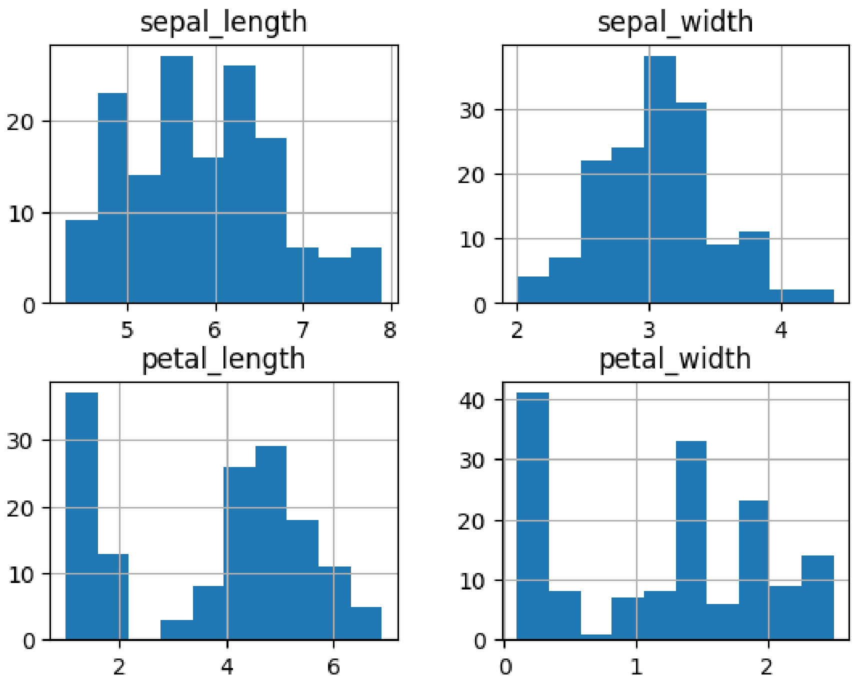


Figure 3, numerical data from Iris dataset

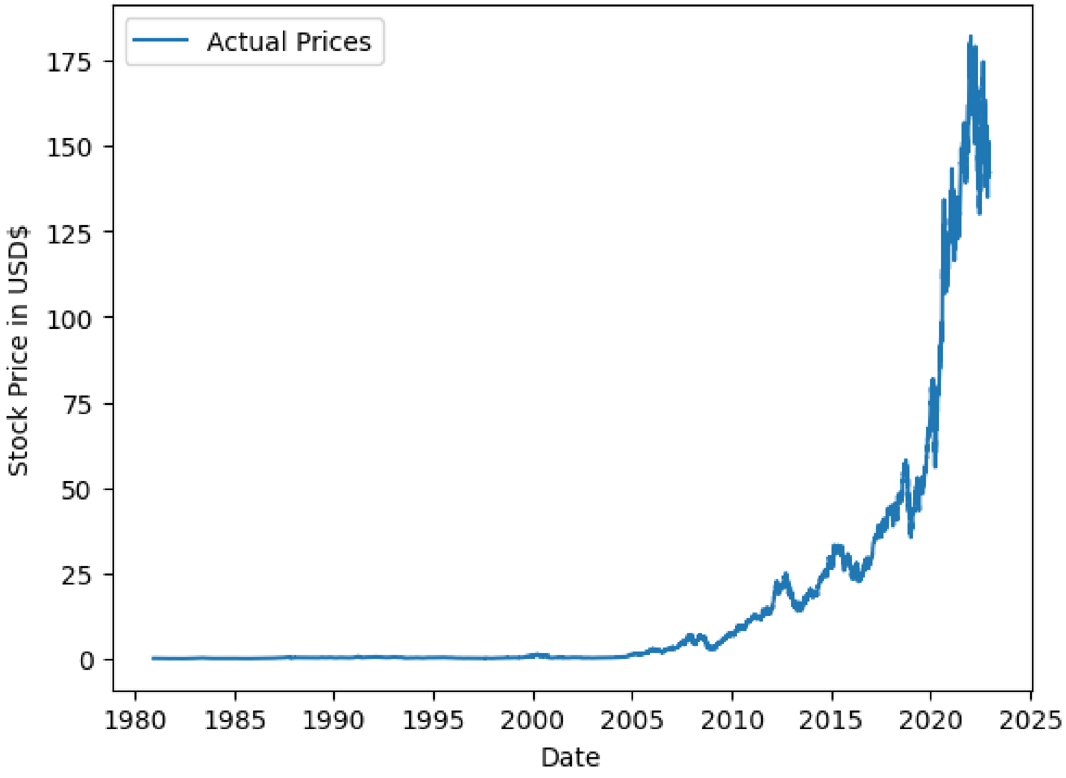


Figure 4, AAPL stock price

Datasets:

1. https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease
2. https://www.kaggle.com/datasets/arshid/iris-flower-dataset
3. https://www.kaggle.com/datasets/borismarjanovic/price-volume-data-for-all-us-stocks-etfs