# TITANIC KAGGLE COMPETITION REPORT

ADVANCED DATA ANALYTICS ASSIGNMENT 1

NAUDÉ CONRADIE – 19673418

# INTRODUCTION

A dataset of the passengers on board the Titanic was provided, with the intention of creating a machine learning model capable of predicting whether or not individual passengers survived its sinking. The dataset contains various attributes which may be used as predictors for the model. The goal is to obtain a model that predicts the survival rate of passengers the most accurately.

# DATA

## ATTRIBUTES

The attributes are listed in Table 1 below, along with a brief description and discussion on their relevance and use, and an example value.

**TABLE 1 - ATTRIBUTES**

| Attribute | Description | Relevance | Use | Example |
|---|---|---|---|---|
| `PassengerId` | Unique identifier for each passenger, starting at 1 and incrementing by 1 until the last passenger | Not relevant for training | Used for identification of each passenger | `1` |
| `Survived` | Binary value of 0 or 1, representing the survival of the passenger, with 0 meaning they did not survive and 1 they did | The key attribute, as the model will be tested to determine its accuracy in predicting this value correctly | Used to train the model against | `0` |
| `Pclass` | Value ranging from 1 to 3, representing the class of the passenger | Higher class passengers, i.e. 1st class, were more likely to survive than lower class passengers, i.e. 3rd class | Scaled and used to train the model | `3` |
| `Name` | A string containing the passenger's full name and title | Titles may indicate higher classes, sex, or age, which have effects as discussed in those respective attributes | Searches were performed to determine the occurrence of relevant titles and return categorical attributes used to train the model | `Braund, Mr. Owen Harris` |
| `Sex` | A string indicating the passenger's sex, i.e. male or female | Female passengers were more likely to survive than male passengers | Converted to binary predictor and used to train the model | `male` |
| `Age` | A value indicating the passenger's age | Younger passengers were more likely to | Scaled and used to train the model | `22.0` |

| | | | | |
|---|---|---|---|---|
| | | survive than older passengers | | |
| SibSp | An integer indicating the amount of siblings and/or spouses the passenger had on board | People with close relationships to other passengers were more likely to survive than lone passengers, as they would likely attempt to ensure that the entire group survived | Combined, scaled and used to train the model | 1 |
| Parch | An integer indicating the amount of parents and/or children the passenger had on board | | | 0 |
| Ticket | A string with the passenger's unique ticket number | Potentially relevant, as they indicate the office of the ticket issued, and passenger cabin placements. Passengers buying tickets together and/or staying together may have had increased chances of survival. | Not used | A/5 21171 |
| Fare | A value indicating the fare the passenger paid | Similar to the Pclass attribute | Scaled and used to train the model | 7.2500 |
| Cabin | A string indicating the passengers cabin number | Potentially relevant, as the location of a passenger's cabin is related to their class and/or may place them closer to lifeboats. However, no information about the layout of the Titanic is provided, and, upon inspection, the majority of the attributes values are empty, and the rest are often erroneous. | Not used | NaN |
| Embarked | One of three characters indicating where the passenger embarked the ship, i.e. C for Cherbourg, Q for Queenstown and S for Southampton | Relevance unclear but easy to incorporate into the model | Converted to three binary predictors and used to train the model | S |

## DATA PREPROCESSING

The `Cabin` attribute was immediately discarded, as the data was too complex and too much was lost or incorrect to gain any use from it.

The `Ticket` attribute was inspected. The text prefixes in Table 2 were found to be attached to some ticket numbers (often followed and/or separated by "\" and " . " characters). These may indicate ticket sales offices, and passengers purchasing tickets at the same locations may have known each other. The ticket numbers indicate the room and/or bedding, and close numbers may similarly indicate neighbours, acquaintances, traveling partners, etc. However, the `Ticket` attribute was not used in the model, as its incorporation was found to be too complex.

**TABLE 2 - TICKET PREFIXES**

| **Ticket Prefixes** | |
|---|---|
| A | PC |
| AH | PP |
| Basle | Paris |
| C | Q |
| CA | S |
| E | SC |
| F | SCO |
| Fa | SO |
| LINE | SOTON |
| O | STON |
| OQ | SW |
| P | W |
| PARIS | WE |

The numerical attributes `Pclass`, `Age`, `SibSp`, `Parch`, and `Fare` were preprocessed as follows. Any missing values were replaced by the median of the respective attribute by a simple imputer. `SibSp` and `Parch` were then added to form a new attribute, as they essentially represent the same thing. Finally, all attributes were standardly scaled.

The categorical attributes `Sex` and `Embarked` were preprocessed as follows. Any missing values were replaced by the most frequent value (mode) of that attribute by a categorical imputer. A one-hot encoder was then used to convert the attributes into one-hot vectors.

Finally, the `Name` attribute was inspected to obtain three new categorical attributes (by nature one-hot vectors) explained in

Table 3 below. Full stops were included after most strings to prevent the string from being found within a person's name. The double quotation marks as seen in the `FancyTitle` category were included as titles or nicknames were occasionally enclosed within them. Additionally, only the male title for children (`Master`) was included, as the female title for children (`Miss`) was also used for unmarried women of any age.

**TABLE 3 - NEW CATEGORICAL ATTRIBUTES FROM NAME**

| New Attribute | Description | Relevance | Relevant Strings |
|---|---|---|---|
| `FancyTitle` | Titles indicating a higher social class | People in higher social classes were more likely to survive | `Sir.`<br>`Lady.`<br>`Count.`<br>`Countess.`<br>`Duke.`<br>`Duchess.`<br>`M.`<br>`Mlle.`<br>`“` |
| `FemaleTitle` | Titles indicating the passenger is female | Female passengers were more likely to survive | `Miss.`<br>`Mrs.`<br>`All female strings from previous attribute` |
| `ChildTitle` | Titles indicating the passenger is a child | Children were more likely to survive | `Master.` |

Once the preprocessing pipeline was completed, the training data and test data were put through the pipeline and various models were tested.

# MODEL TRAINING

Various different models were tested in order to find one that fit the data best. Figure 1 and Table 4 below shows the accuracy of the respective models as obtained by 5-fold cross-validation for various models. The four best models were all submitted to Kaggle for the competition, and are discussed further on.

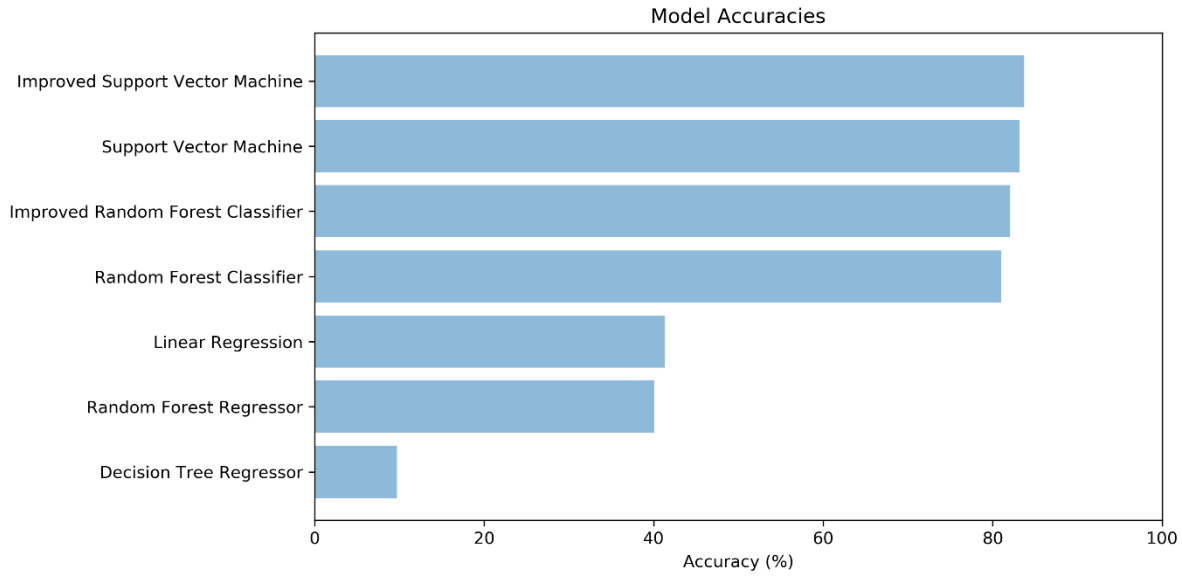


FIGURE 1 - MODEL ACCURACIES

TABLE 4 - MODEL ACCURACIES

| Model | Accuracy (%) |
|---|---|
| Improved Support Vector Machine | 83.73 |
| Support Vector Machine | 83.17 |
| Improved Random Forest Classifier | 82.05 |
| Random Forest Classifier | 81.04 |
| Linear Regression | 41.35 |
| Random Forest Regressor | 40.06 |
| Decision Tree Regressor | 9.72 |

### RANDOM FOREST CLASSIFIER

The random forest classifier was found to perform well. In order to improve the model, a grid search was used to find better hyperparameters. However, the grid search was found to be slow and inefficient at finding optimal hyperparameters. Thus, a randomised search was used instead. The results from the randomised search were consistently found to be better than those from the grid search. The final hyperparameters as used by the model are shown in Table 5 below.

**TABLE 5 - RFC HYPERPARAMETERS**

| Hyperparameter | Value |
|---|---|
| max_features | 7 |
| n_estimators | 436 |

### SUPPORT VECTOR MACHINE

The support vector machine was found to perform even better than the random forest classifier. In order to improve the model, a grid search was used to find better hyperparameters, as the grid search was found to be efficient at finding optimal hyperparameters. The final hyperparameters as used by the model are shown in Table 6 below.

**TABLE 6 - SVM HYPERPARAMETERS**

| Hyperparameter | Value |
|---|---|
| C | 15 |
| gamma | 0.02 |

## RESULTS

The final results of the best models as submitted to Kaggle are found in  below. As of the time of writing, the best model has placed in 2167$^{th}$ position, which is in the top 21%.

**TABLE 7 - KAGGLE RESULTS**

| Model | Kaggle Score |
|---|---|
| Improved Support Vector Machine | 0.79425 |
| Support Vector Machine | 0.78947 |
| Improved Random Forest Classifier | 0.73684 |
| Random Forest Classifier | 0.74641 |

## CODE

The code for this project may be found at the following link:

https://github.com/NaudeConradie/ADA874/blob/master/Kaggle%20Competition/Titanic/TitanicMLCompetition.py