L-Università ta' Malta
Faculty of Information & Communication Technology
Department of Computer Information Systems

GAPT Assignment, Titanic - Machine Learning from Disaster

Alana Busuttil (0363502L), Edward Sciberras (274402L), Andrew Darmanin (95602L)
B.Sc. (Hons) Information Technology (AI)

Study-unit: **Group Assigned Practical Task**
Code: **ICS2000**
Lecturer: **Dr Kristian Guillamier**

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Declaration

Plagiarism is defined as “the unacknowledged use, as one's own, of work of another person, whether or not such work has been published, and as may be further elaborated in Faculty or University guidelines" (University Assessment Regulations, 2009, Regulation 39 (b)(i), University of Malta).

We, the undersigned, declare that the assignment submitted is my / our* work, except where acknowledged and referenced.

We understand that the penalties for committing a breach of the regulations include loss of marks; cancellation of examination results; enforced suspension of studies; or expulsion from the degree programme.

Work submitted without this signed declaration will not be corrected and will be given zero marks.

(N. B. If the assignment is meant to be submitted anonymously, please sign this form and submit it to the Departmental Officer separately from the assignment).

Alana Busuttil

Student Name | Signature

Edward Sciberras

Student Name | Signature

Andrew Darmanin

Student Name | Signature

ICS2000	Titanic - Machine Learning from Disaster
Course Code	Title of work submitted

20/05/2022

Date

Contents

1. Introduction

1.1 What is Machine Learning?

Artificial intelligence may be defined as the replication of a human intelligence, it well simulates human actions and thought. AI is a growing sector integrated into our everyday lives, from the improvement of social media platforms such as Twitter and Instagram to self driving and parking vehicles. Machine learning is a branch of AI in which algorithms improve themselves over a large amount of data. As defined by Tom Mitchell, founder of the world's first Machine Learning department, "Machine learning is the study of computer algorithms that allow computer programs to automatically improve through experience [1,2]."

1.2 Machine Learning and the RMS Titanic

The RMS Titanic closed its final stages of building and construction in 1912 and set sail on its maiden voyage during that year. The ship departed from Southampton, UK and planned to arrive at New York City. Unfortunately, during its trip, the ship hit an iceberg that caused it to sink. Out of the estimated 2,224 passengers and crew on-board, over 1,500 people lost their lives due to the ship not having enough lifeboats and the water being unbearably cold. Years later, Machine learning techniques can be used to give educated predictions on survival. A dataset containing details of all registered passengers was fed into a Machine Learning model. The data was then processed accordingly and the final output was which passengers were likely to have survived and which were not.

1.3 Our Approach

The objective of this project is to train a Machine Learning model to make educated predictions on who did and did not survive the well known Titanic tragedy. Each team member selected two Machine Learning models to build using python. The models were trained and the final results were analysed and compared. Finally, the most accurate model was selected as the optimal one. The models selected to tackle this task were the following:

1. Artificial Neural Network
2. Random Forest Classifier
3. Logistic Regression Model
4. Support Vector Machines

5. Naive Bayes Model
6. K-Nearest Neighbours

The scope of this project is mainly on using different machine learning techniques, noticing the differences between them and drawing conclusions. In this research an in depth evaluation of each technique will be produced. Finally with the help of the results a comparison of the algorithms efficiencies and accuracies will be made. This way one can note the advantages and disadvantages of making use of one of these machine learning models to solve a similar problem.

It would be expected that some machine learning models will outperform others. However this does not mean that they are necessarily better, all it shows is that for this kind of data it managed to generalise better. For example, Artificial Neural Networks tend to overfit and hence learn the detail and noise in the training data, which results in a poor performance. While, models like random forest which makes use of random samples from the dataset hence avoiding overfitting resulting in better results on the test set. The performance of each machine learning model will be based on the score it achieves when submitted to the kaggle challenge.

It is important to note that for this scenario a better score does not necessarily mean that it is a better technique. It is impossible to make a statistical model to achieve a near 100% accuracy. This is due to uncertainty and lack of information. Mainly, in the dataset there are no variables such as medical conditions, athletic background and others which might have a role in determining whether a person survives or not. Moreover the situation was a total chaos and there was certainly an element of luck which the machine learning models cannot account for. All in all, care was taken to not overfit the training data so that the model can generalise better.

1.3 Key Words Used

ML	Machine Learning
AI	Artificial Intelligence
ANN	Artificial Neural Networks
RF	Random Forest
SVM	Support Vector Machines

2. Literature Review

Recently, machine learning techniques have been implemented to handle many applications [3]. Their ability to analyse data at a large scale, adapt and be able to conclude logical results makes them attractive when dealing with datasets [4].

2.1 Artificial Neural Networks

Artificial Neural Networks (ANNs) have a relatively high success rate when used in real world situations, an example being making loan decisions at banks. [5].
An ANN was constructed to assist with evaluating credit applications. This was achieved by training the ANN on a dataset of 140 personal loan applications. Each applicant was abstracted to 11 influential variables.
The variables chosen were those which were relevant to the loan decision, for example 'Age', 'Income' and 'Job'. Variables such as 'Gender' and 'Name' were omitted. Hyperparameters like network parameters, the number of hidden neurons, the learning rate and the momentum applied all had an effect on the accuracy of the network. The final output of this ANN was 0 or 1, a bad candidate or a good candidate respectively. The constructed ANN was able to classify 95% of the cases in the testing set.
This may be compared to the titanic shipwreck application, although different in nature, there are many underlying similarities with regards to constructing the ANN that predicts the Titanic shipwreck survivors. In both cases, the input is a person record having multiple variables and the output is 0 or 1. Moreover, in both applications the ANN parameters require keen tuning. Given that the ANN performed well in the load decision process, it is expected to be able to handle the Titanic's data with satisfactory results. Given the flexible nature of an ANN it will be interesting to observe how it can handle data and the results it can achieve.

2.2 Support Vector Machines

Support Vector Machines (SVMs) are used in speech recognition, a feature implemented in most phones nowadays as virtual assistants. [6].
The model was trained on speech data, it then analysed the spectrogram, a graph representing sound waves, and learned that certain patterns represent certain sounds. Once the sound data was gathered, words and sentences started to form.
In addition to speech recognition, SVMs are also used in facial recognition. [7].

The model created data points relating to a person's facial features and compared it to past photos of the person. Finally, a true or false (0 or 1) output was considered to check if the person is the same or not.
Similarly, SVMs would be to detect facial expressions and relate them to emotion. The model would have to be trained on how data points of a person's face would relate to each emotion, it would then be able to compare these results to the provided test set.

SVM could be useful in predicting the shipwreck survivors scenario as it is not only capable of producing a binary output, but is also able to take in multiple data points to be processed and adjusted accordingly. Therefore, the SVM is expected to have good performance and produce satisfactory results.

2.3 Random Forest Classifier

Random Forest Classifier (RF) has a number of advantages over other ML techniques. When compared to Decision Trees, RF performs significantly better since it averages all the random built trees and results in a lower variance and bias. This may be observed in a study comparing both approaches, it was concluded that RF gives better results for the same number of attributes and dataset size. [8]
As for Support Vector Machines, Random Forests perform relatively similarly. This is seen in the study made by M. Pal [9].
Hence it is expected that the RF model will perform relatively similarly to the SVM model.

RF adaption is widely used in a number of applications, one example being Health Monitoring [10]. RF performed well in these applications, hence it is expected to be feasible when implementing RF for the titanic shipwreck survivors scenario.

2.4 Logistic Regression

Logistic Regression (LR) can be seen being used in credit scoring [11].
A financial company uses more than 15 data features and feeds them into a logistical regression model that will then predict credit scores of individuals. This technique is used as it is relatively easy to find out which variables are redundant. With this feature elimination, the attributes that matter and affect the result could be focused on more. Considering these advantages, logistic regression has seen excellent results in this sector of finance.

Logistic Regression can also be used in tumour prediction [12].

The model takes into account multiple variables such as: tumour size, body area and acceleration of growth. The model then produces a binary outcome to predict whether the tumour is malignant.
Given that LR is useful in deciding which variables favour the best outcome as well as provides a binary output, that it would fit well in the titanic shipwreck survivors scenario.

2.5 Naive Bayes

Naïve Bayes is best known for its excellent performance when filtering emails into spam [13]. This is achieved through assigning a higher spam probability to spam-related words, for instance 'free' and 'win'. Yet words such as relative or friends' names have a low spam probability. This process is essentially simple, the data is filtered and preprocessed (excluding noise, word frequency calculation), the required features are selected (in this case, the words to be processed, based on the frequency calculations) and finally the classifier is applied. In the context of the given study, Naïve Bayes was used to filter through spam in two datasets, the highest accuracy score being 91.13%.
Although the study is machine learning in NLP, a similar approach can be taken in our scenario. A likelihood table may be produced by calculating the independent probabilities on selected features, meaning probabilities related to attributes such as sex, age, gender and so on. The classifier will then be applied. This application is obviously much more streamline than the given Titanic scenario. The Titanic had much more conflicting and inconsistent data. However, if the Titanic's dataset is well fitted and smoothing is applied, Naïve Bayes could prove simple and effective [14].

2.6 Nearest Neighbours

Using K Nearest Neighbour, the most likely score for one's survival can be predicted. Through well-chosen features, the classifier would produce a binary true or false output. Although a simple, lazy learning algorithm; a well-tuned distance function, neighbourhood size and class probability estimation may lead to promising results [15].
A main fault of KNN would be the Euclidean distance's 'curse of dimensionality', referring to the restrictions that come along with linear distance. A way to overcome this, as shown in the study improving KNN's pitfalls, would be by weighting each attribute.

KNN is used to detect breast cancer in its early stages [16], to do so, Euclidean and Manhattan distance were deemed the most suitable out of 6 other experimental

distances, including city block difference, cosine difference, correlation distance, and so on. K nearest neighbour was used to compare an element to other elements using a similarity measure before classifying it. The neighbours are weighted as seen in the previous study. KNN worked well in the given study, with a 98. 70% accuracy when using Euclidean distance and a well-tuned distance function. It is therefore predicted that the algorithm will be successful in the Titanic scenario if close attention is played to feature selection and relation functions. Weighting the neighbours will be attempted.

3. Preprocessing Data and Visualisation

3.1 Data Cleaning

Before fitting into the models, the data required cleaning, this refers to detecting and arranging corrupt or invalid data records. Two similar datasets ‘train.csv’ and ‘test.csv’ were used. These two datasets contain passenger information, the only difference being that ‘train.csv’ contains a feature named Survived, which reveals if that person survived or not. This Survived column will be used as the labels for the ML models. In fact ‘train.csv’ will be used in the training phase, while ‘test.csv’ will be used in the evaluation phase. More on this later. The first task was to preprocess the data.

Data preprocessing is a crucial part of any machine learning task. Since most of the time data contains noise and missing values. These negatively affect the performance of the modals. Hence to handle noise, certain features were deemed useless or unimportant and removed. The ‘train.csv’ dataset contain the following features: PassengerID, Survived, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin and Embarked. These variables should be influential attributes that determine whether a passenger survived the Titanic shipwreck or not. However it is self-evident that a person's passenger identification number, name and ticket number do not play a role in determining whether the individual survived or not. Therefore these features were removed from the datasets. On the other hand, the feature Embarked is a bit ambiguous, in this research the importance of this variable will be analysed. Another feature which was removed was Cabin. Although the passenger's cabin might have an effect on survival rate, the reason it was removed was because around 77% of the cabin fields are missing in the training set. It is useless to try and compute the missing values due to the lack of viable options available. Also it is important to note that passengerId is just an identifier and from now on the models will use index as an identifier instead. To further reduce the dimensions of the dataset, SibSp and Parch will be merged into one feature FamilySize. The rationale here was that family size is an accurate replacement for the number of siblings/spouses and of parents/children aboard the Titanic. The FamilySize attribute is computing by adding up these two features.

Another part of preprocessing data is handling missing values. In the datasets feature Age has a few missing values. These missing values must be computed in a way that noise and bias is minimised. For this reason k-Nearest Neighbour was used. When used to find missing age values, this algorithm will identify k samples in the dataset that are similar to the missing data point, then estimate the age value by using the

mean value of the k samples (where k is an integer). K-NN provides an efficient way of estimating the missing age values.

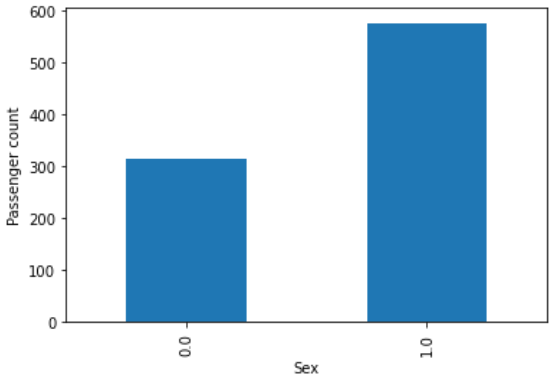
One-hot encoding is the process of encoding categorical features as a one-hot numeric array. This in return makes the data more readable to the ML models and results in better performance [17]. Here features Sex, Pclass and Embarked are encoded. For feature Sex, the same column was used but 'male' was changed to 1 and 'female' to 0. For feature Pclass, this column was converted to three new columns 'A_Class', 'B_Class' and 'C_Class'. After this step, the datasets are converted to numeric (or binary) values only.

Data normalisation is a data transformation process that aligns data values to a common scale. In this case min-max normalisation was applied on Age, Fare and FamilySize transforming values to [0,1]. This ensures that variables with large magnitudes do not exert disproportionate influence over variables with small magnitudes. This is especially important when dealing with ML models such as Neural Networks, where initial weight bias is significant if not normalised [18].

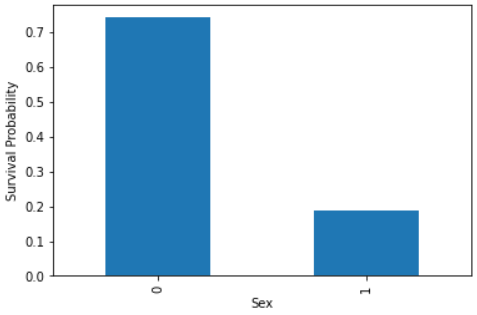
The dataset post cleaning provided more compact and streamline input for the ML models.

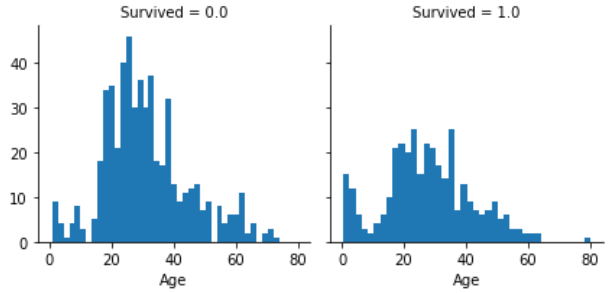
3.2 Data Analysis

When analysing the data, charts and graphs were created for better visualisation. Below are some instances of such followed by any observations made.

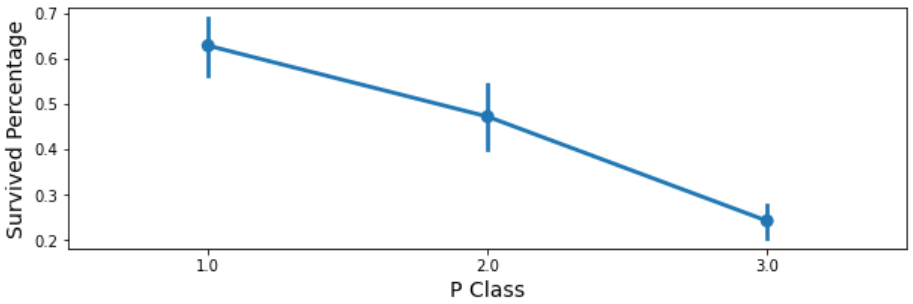


According to the dataset, the number of males aboard the Titanic is much larger than the number of females. Considering the number of females and the fact that they were given first priority, it is likely that more females survived, this speculation may be confirmed in the next figure below.

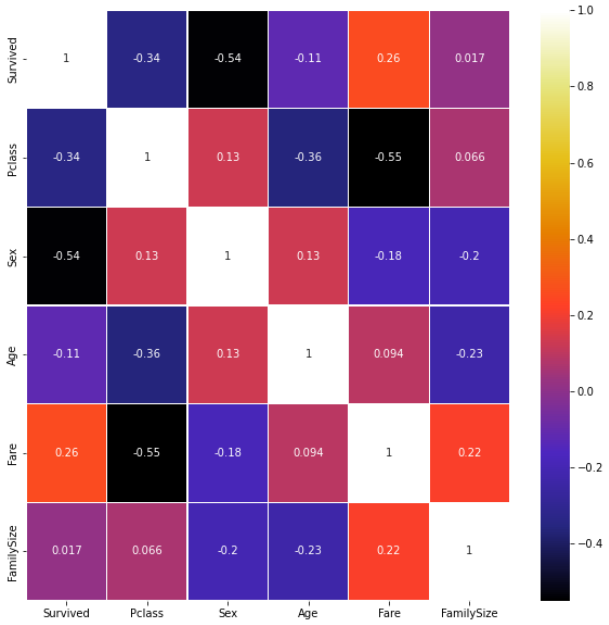




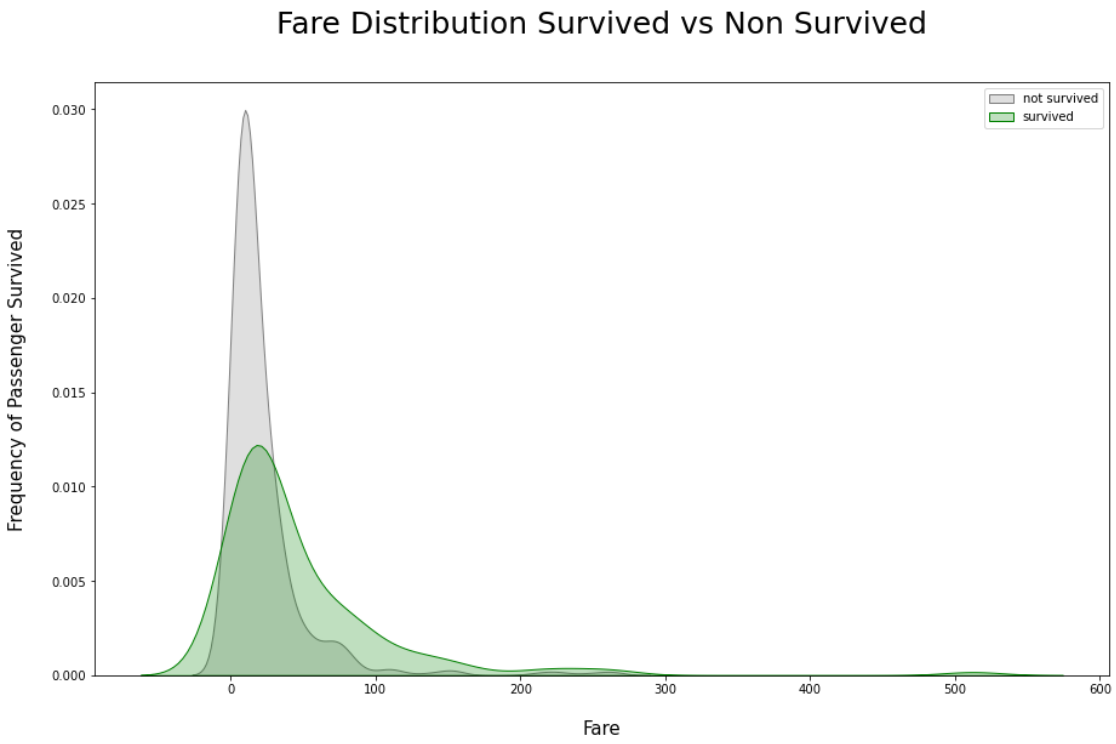
The figures above show survival with reference to age demographic. The majority of survivors fit into the 20-40 year old range, this is relevant to both male and female survivors. Younger males were more likely to survive than elder ones, this is as children were made a priority irrespective of gender.



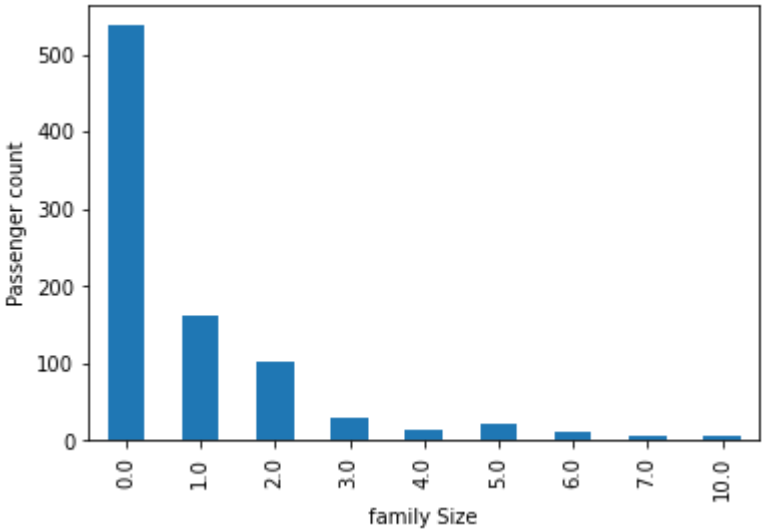
The figures above depict survival with relation to the passenger class, which is a reflection of the socio-economic class. It is evident from the negative slope that the majority of survivors were of a higher class. This means that those of a higher economic demographic were made priority.

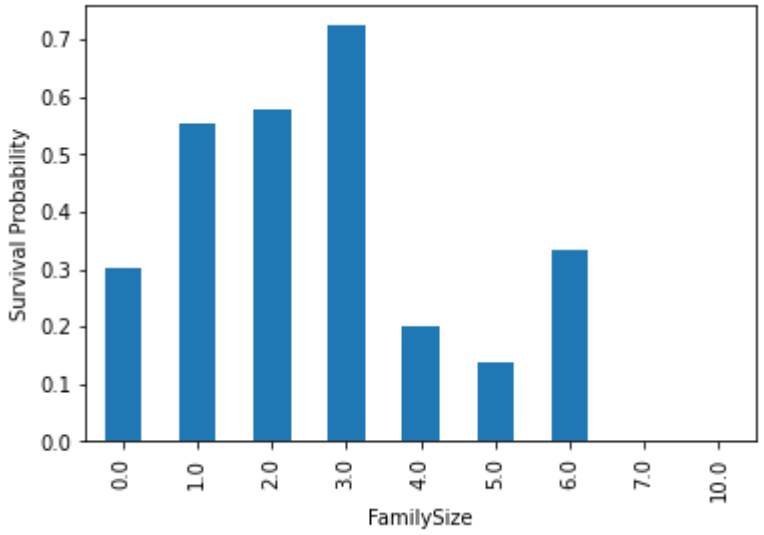


The figure above shows correlation between the variables. Important to know which variables are most influential to surviving. From the plot one can conclude that the most impactful variables are Sex, Pclass and Fare in that order.



The plots below show the effect of family size on survival rates.





4. Implementation, Model and Model Design

In this section a deeper understanding of the techniques will be discussed.

4.1 Artificial Neural Network

Artificial Neural Network (ANN) is inspired from how the human brain neural network is designed. An ANN is a collection of perceptrons and activation functions that map input to output. By making use of an input layer, hidden layer/s and the output layer an accurate representation of a function is captured. Moreover it is capable of performing pattern recognition. The pattern is learnt by having every input paired with a weight, then in training mode, by making use of an error function and an optimizer, these weights are tweaked to accurately represent the given data. That is, the model makes adjustments to reduce the error of the current representation. Here, since the output (survived) is a binary variable, the loss function will be binary cross entropy. The optimizer used will be Adam, since it uses gradient descent with momentum hence can escape local minimums. To deal with overfitting dropout regularisation is used. Finally the number of epochs, the number of perceptrons per layer, the number of dense layers were chosen by trial and error, always trying to maximise binary_accuracy on both the training set and the test set.

4.2 Random Forest Classifier

A decision tree is a model that predicts the value of a target vector by learning simple decision rules inferred from the data features. Making a decision tree an effective decision-making diagram, however it tends to overfit when handling complex datasets.
The main idea of Random Forest (RF) Classifier is to construct decision trees on different samples from the training set and take their majority vote for classification. This predictive model is effective in handling datasets similar to the Titanic's. Furthermore, since Random Forests use random subspace methods and bagging, they are much less sensitive to the data compared to a decision tree. This prevents overfitting and hence provides a higher accuracy.

The RandomForestClassifier takes a number of hyperparameters. The best max_depth (=6), min_samples_leaf (=1) and n_estimators (=23) parameters were found using GridSearchCV.

4.3 Support Vector Machines

Support Vector Machines (SMVs) involve separating the dataset into two categories that are separable by a hyperplane. A hyperplane is an equation in a graph that would graphically separate the two categories. For instance, in a 1-dimensional plane it would be a dot, in a 2-dimensional plane it would be a line and in a 3-dimensional plane it would be a plane.
Certain assumptions need to be made about the data before this technique can be applied, The first of said assumptions being that there can be no outliers. If there are outliers in the data then a variable can be used when making the hyperplane. The larger the value of this variable then the narrower the hyperplane and thus less tolerant towards the outliers. On the other hand, if the variable is smaller, this will result in a wider hyperplane at the cost of a few misclassifications. The second assumption made is that the data can be linearly separable. If not, then a kernel trick can be used to transform the data from one space to another.
A key point in SVMs is to find the optimal hyperplane that separates the data in the most efficient way possible. This is done by finding the extremes from both categories and placing the hyperplane between them. These are called the Support Vectors, which are the nodes closest to the hyperplane.
Unlike other techniques in which all data points influence the final optimization, in SVMs, only the extreme cases (support vectors) do. So much so, that moving the support vectors will impact the decision boundary whilst moving any other data points will have no impact. Important to note that here SVC() was trained on the whole training dataset.

4.4 Logistic Regression

Logistic Regression is used to model the probability or chances of an event happening. The model could also be manipulated in such a way that depending on the probability (below or over 50%), the output would simply be a “true” or “false” for every data entry as opposed to a continuous value. When a logistical curve (S shaped line) is formed from the model's output, it can be used to predict the probability of the event happening or not for that node.
When using multiple variables as input for the LR model, it must be ensured that the variables in use are valid in estimating the probability of the event taking place for

each individual node. To choose the variables that correctly influence the outcome, Wald's Test will be performed on each potential variable before executing Logistic Regression.
The way Logistic Regression works is that it uses the coefficients that are produced and estimated from the training data. These values are formed through the Maximum-Likelihood estimation. The most efficient and accurate coefficients would result in a model that predicts values very close to 1 for one state and very close to 0 for the other.
In this case we will be using Multivariate Binary Logistic Regression, which essentially means that we will be using Logistic Regression with multiple variables and the outcome for each node would be a boolean "true" or "false". The LogisticRegression function takes a couple of parameters, here penalty term was set to L2 and Stochastic Average Gradient descent was used as the solver. The whole training dataset was used to train this classifier efficiently.

4.5 Naive Bayes

The Naïve Bayes algorithm finds the class with the highest likelihood under Bayes law. Bayes law may be defined as the following:

The conditional probability of an event, based on the occurrence of another event, is equal to the likelihood of the second event given the first event multiplied by the probability of the first event.

Or mathematically represented as the following:

$$P(E1|E2) \propto P(E2|E1)P(E1)$$

The 'Naïve' in Naïve Bayes refers to the naïve assumption that all events (input variables) are independent.

In the given context, the data set would first be converted into a frequency table (the frequency of each event happening), from these frequencies the probabilities will be found, and finally the posterior probabilities will be found using Naïve Bayes. The posterior probability being the chance of survival.

Naïve Bayes, in comparison to other probabilistic algorithms, is quick to train and classify, is robust and has a low variance, meaning it requires little changes to the data set to fit the target function. It is ideal in low data situations and is optimal if the independence assumption holds true. However, in the context of the Titanic, and many other real-life applications, the independence assumption is rarely ever true. Furthermore, smoothing is required for unforeseen classes and combinations due to

the zero frequency, meaning such events are automatically given a probability of 0, giving skewed results. Here, Naïve Bayes was training using GaussianNB() and on a training set of 70% of the ‘train.csv’.

4.6 K Nearest Neighbour

k-means clustering, also referred to as K-nearest neighbours, is a method of vector quantization that aims to partition n observations into k 'groups', or clusters. The ‘K’ in K-means therefore represents the number of clusters. Using Euclidean distance, the KNN algorithm will assign the most common class of variable x’s k nearest neighbours to x. KNN is generally used in classification problems, meaning it produces a fixed output, for instance true or false.

Other clustering techniques include:

- Density-based clustering
- Gaussian Mixture models

K-means clustering is the most popular for reasons of robustness and easy, simple implementation.

In the context of the Titanic, the data must be optimised, and a relation feature must be found (the two axis). Although KNN itself doesn’t require any training, the relation feature does. KNN is then used to predict the final scores, this being a binary true or false in terms of survival. Here, KNN was computed using KNeighborsClassifier() and on a training set of 80% of the ‘train.csv’. For the choice of k, different integers were considered, the best performing k was 5.

5. Model analysis and Discussion

In this section, the final results of accuracy and performance will be shown and compared to one another, as well as different configurations of the dataset to see how the manipulation of data can have an impact on the final results and how big that impact can be. The table below shows the accuracy of each model when predicting the test dataset with each configuration of the data sets.

	SVM	**LR**	**ANN**	**RF**	**NB**	**kNN**	**Avg**
Default	0.780	0.758	0.791	0.789	0.720	0.761	0.767
Without Data Normalisation	0.665	0.672	0.782	0.794	0.713	0.658	0.714
Without FamilySize	0.778	0.761	0.785	0.780	0.713	0.746	0.761
Using kNN = 1 for Age Replacement	0.780	0.758	0.778	0.782	0.715	0.761	0.762
Using kNN = 5 for Age Replacement	0.780	0.758	0.787	0.784	0.718	0.785	0.769
With Embarked Attribute	0.775	0.761	0.792	0.780	0.684	0.787	0.763
Average	0.760	0.745	0.786	0.785	0.711	0.750	

Table 1

The default configuration for all models in this case was:

- Using FamilySize as a feature
- Normalising the age, fare and FamilySize
- When using kNN to fill empty values of age use 3 neighbours
- The embarked attribute was not used

Starting off with the FamilySize feature addition, this attribute was constructed from the addition of the “SibSp” and “Parch” features in the original dataset. With those two representing the number of siblings on board and the number of parents and/or children on board with each passenger respectively. When FamilySize is taken out of the dataset and “SibSp” and “Parch” are put back instead, almost all models perform slightly worse. The difference is not great but it is a notable difference that can and will affect final results. The only model that actually improved slightly with the removal of this feature was the Logistical Regression model. This can be due to the way the model works under the hood with the extra dimension of accuracy, it could also be pure coincidence as the improvement is only 0.3% and the training and test set are not big enough to justify this tiny difference. A possible reason as to why most models had a tendency to perform worse when both features were not combined into one attribute could be due to the Curse of Dimensionality.

In machine learning, since in most cases, a new feature means a new dimension on the model that will be used, too many features and thus dimensions are not always a good thing. When more dimensions are in play for a model then more data would be required for the machine learning algorithm to reach a good performance. In addition, there would need to be a greater spread of samples and training data to vary from all data features and attributes. For instance, if there is a model with ten binary features, one would need a minimum of 2^{10} (1024) values to fill out every single possible type of data point.

With that being said, this is a very plausible reason as to why combining the sibling count and the parents and/or child count slightly improved on the results of all models except for one. Overall, it is safe to say that combining these features was a plus for the accuracy.

Normalising the data is important as it ensures that the data can be used and processed in the same way across all models in an equal way and means that no single variable will be weighted more than the other. As if one feature is normalised and the others are not, if the unnormalized values are larger in nature, then the

models might tend to be a bit more biassed towards those types of values as opposed to the normalised ones. With all that being said, in short, it improves the accuracy of the model. In this case, three features were normalised, the age of the passenger, the fare that the passenger paid, and the previously explained FamilySize feature.

To see the extent of how important it is to normalise data as opposed to leaving the raw data as is with pure and possibly large numbers, all the models were run without any data normalisation. In general, all models performed worse in accuracy. Even taking the base stat of average accuracy when compared to the other data set types, the accuracy goes down by just over five percent. Some models took much bigger hits than the others, for instance Support Vector Machines and k-Nearest Neighbours both lost out on more than ten percent whilst Logistic Regression took a hit of almost eight percent. These models suffering the most could be attributed to the way they work being dependent on the data being normalised and processed before being inputted to said model. Other models such as the Neural Network and Naïve Bayes only took a small hit. Maybe these models do not depend on the normalisation of data as much as the previously mentioned models that took the biggest hits when compared to the other models. Interestingly, Random Forest actually improved on its accuracy. This is due to the fact that scaling is not necessary for Random Forests. RF uses each variable independently to form multiple decision trees, hence the nodes of these decision trees are just trying to split a variable range. Overall, as seen from the results, most would say that for most machine learning models data normalisation is not only important but imperative to get accurate results that can be depended on.

Since age is such a vital feature when it comes to predicting the outcome of the passenger, when it was seen that there were missing values in the provided datasets, something had to be done to fill in these holes of data. In total, there were one hundred and seventy-seven (177) missing age values out of the total eight hundred and ninety-one (891) passenger data points, this being a percentage of almost twenty percent. It was definitely not plausible to just delete almost twenty percent of the data. Thus, in order to replace the empty and missing data that can be found in both training and test sets of the given data, it was decided that a k-Nearest Neighbour model would be used to try and estimate or predict the age of the passenger with the other data that was given. All the other features such as fare, FamilySize, sex and class of room were used. In addition, this k-Nearest Neighbour model was testing with three different values of k to see what the difference in performance would be.

By default, the k value was set to three. When the k value was set to one, meaning that the age only takes into consideration the node absolutely closest to it without any consideration of any more, it is clear to see that although it was not a big dip in performance and accuracy, the dip was there nonetheless. Simply looking at the average accuracy, it is lowered by 0.5%. Although it is not a big difference it is

important to see that it makes a difference even if it is affecting just under twenty percent of the total data from the datasets. Looking specifically at the individual models, all models either did not get affected by this change, such as Support Vector Machines, Logistic Regression and k-Nearest Neighbour, or got a small dip in accuracy like the other models, which are, Neural Networks, Random Forest and Naïve Bayes.

On the other side of the spectrum, when the k-Nearest Neighbour age prediction model has a k value of five, different results will start to appear when compared to the previous attempt. Four out of the six models improved on their results as opposed to the default configuration. The other two models out of the six equaled the performance when compared to the default configuration. These two models being the Support Vector Machine and the Logistic Regression. With that being said, such a small change in selecting five neighbours to take into consideration as opposed to three can still make a difference. Albeit not a big difference, it is still there. There is almost a one percent difference on average between using k value of five and a k value of one in the favour of k value five.

Finally, the last configuration that was experimented with was if the embarked feature in the original dataset should be used to calculate the final prediction. There was a slight discussion if this should be used or not as it failed to realise if there was any correlation to the final outcome of the passenger and if the extra dimension would be worth it in the grand scheme of things. Ultimately, looking at the results, it is a split decision as three of the models (Support Vector Machines, Random Forest and Naïve Bayes) performed worse whilst the others (Logistic Regression, Neural Networks, k-Nearest Neighbour) had an increase in accuracy. However, when an average of each algorithm is taken, there was a slight 0.4% decrease in accuracy. As previously mentioned, since there is not a definitive answer to whether this feature helped in the predictions or not, this specific feature could just increase the damage of the Curse of Dimensionality whilst not providing enough training data to support the extra dimension. Another reason could be that this data simply does not have a strong, if any, correlation to the final outcome of each passenger. There was no reason found as to why someone boarding from Queenstown had more chance of surviving or dying than someone from Southampton or Cherbourg. Therefore, this feature did not seem to outweigh the disadvantages it brings from the Curse of Dimensionality.

To conclude, these results showcase the importance of selecting correct features, data cleaning and normalisation of the data for any machine learning model. Moreover, these results can be used to analyse the difference between the models when coming to solve similar problems.

6. Conclusion

All assignment objectives were achieved. The Machine Learning models were successful in execution and predictions, all scores on kaggle were relatively high. As seen in table 1 above, the highest scoring models were the Artificial Neural Network and Random Forest, producing average scores of 0.786 and 0.785 respectively. Support Vector Machines, K- Nearest Neighbour and Logistic Regression had fairly similar scores, all having a 0.01 difference or less between one another. The worst scoring model being Naive Bayes, scoring 0.711, this is likely due to the assumption required. It was ensured that the data was well pre-processed, this meaning the disclusion of redundant data, the filling in of missing values, the joining of columns when necessary, and in depth analysis of visual data representations for a better understanding of the demographic. Accounting for empty values was done through KNN prediction, this was decided on as best suited after attempting various other algorithms. Variable selection was based on logical assumptions made through data visualisation, factors which were assumed to be important, supported by statistics, such as ticket fare and gender were included. Hyperparameters for the different models were also tuned accordingly, different models required different configurations, the final results were achieved through a series of tests and trial and error. Further improvements to the models are specific to each, for instance using weighted variables would apply to KNN whilst more or less layers would apply to ANN, but all are subject to further tuning of the hyperparameters or better variable selection. The most suitable model for survival prediction, according to average scoring as displayed in table 1, would be the Artificial Neural Network. Table 2 below displays the models and their overall ranking.

Model	Average Overall Score	Ranking
Artificial Neural Network	0.786	1
Random Forest	0.785	2
Support Vector Machines	0.760	3
K-Nearest Neighbour	0.750	4
Logistic Regression	0.745	5
Naive Bayes	0.711	6

Table 2

Citations and references:

[1] I. Mihajlovic, "How Artificial Intelligence Is Impacting Our Everyday Lives", *Medium*, 2022. [Online]. Available: https://towardsdatascience.com/how-artificial-intelligence-is-impacting-our-everyday-lives-eae3b63379e1.

[2]R. Iriondo, "What is Machine Learning (ML)?", *Towards AI*, 2022. [Online]. Available: https://towardsai.net/p/machine-learning/what-is-machine-learning-ml-b58162f97ec7.

[3]Sarker, I.H. Machine Learning: Algorithms, Real-World Applications and Research Directions. SN COMPUT. SCI. 2, 160 (2021). Available: https://doi.org/10.1007/s42979-021-00592-x

[4]Caiafa, C.F.; Sun, Z.; Tanaka, T.; Marti-Puig, P.; Solé-Casals, Machine Learning Methods with Noisy, Incomplete or Small Datasets. Appl. Sci. 2021, 11, 4132. https://doi.org/10.3390/app11094132

[5] Shorouq Fathi Eletter, "Neuro-Based Artificial Intelligence Model for Loan Decisions", American Journal of Economics and Business Administration, vol. 2, no. 1, pp. 27-34, 2010.

[6]Team, T. (2021, June 24). SVM Applications – Top 10 astonishing real life applications of SVM. TechVidvan. https://techvidvan.com/tutorials/svm-applications/

[7]TY - BOOK SN - 978-3-319-01603-0 Feature Selection for Enhanced 3D Facial Expression Recognition Based on Varying Feature Point Distances Available: https://www.researchgate.net/figure/SVM-Classifier-system-used-for-facial-expression-recognition_fig1_258565562

[8] Ali, Jehad & Khan, Rehanullah & Ahmad, Nasir & Maqsood, Imran. (2012). Random Forests and Decision Trees. International Journal of Computer Science Issues(IJCSI). 9. Available: https://www.researchgate.net/publication/259235118_Random_Forests_and_Decision_Trees

[9] M. Pal (2005) Random forest classifier for remote sensing classification, International Journal of Remote Sensing, 26:1, 217-222, DOI: 10.1080/01431160412331269698

[10] Ricordeau, Julien & Lacaille, Jérôme. (2010). Application of Random Forests to Engine health Monitoring.

[11] 5 Real-world Examples of Logistic Regression Application. (2021). ActiveWizards: Data Science and Engineering Lab. Available: https://activewizards.com/blog/5-real-world-examples-of-logistic-regression-application

[12]Goel, A. (2018, May 21). *4 Logistic Regressions Examples to Help You Understand*. Magoosh Data Science Blog. https://magoosh.com/data-science/4-logistic-regressions-examples/

[13] Nurul Fitriah Rusland et al 2017 IOP Conf. Ser.: Mater. Sci. Eng. 226 012091

[14]Lewis, D.D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. In: Nédellec, C., Rouveirol, C. (eds) Machine Learning: ECML-98. ECML 1998. Lecture Notes in Computer Science, vol 1398. Springer, Berlin, Heidelberg. https://doi.org/10.1007/BFb0026666

[15] *Survey of Improving K-Nearest-Neighbor for Classification*. (2007, August 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/4406010/;jsessionid=0IDeVO3DdEK3JB94eoqz8tDyzMWLJeufGUvxKBRaNzJwYFvrqMkZ!-1666311061?tp=&arnumber=4406010

[16] European Journal of Molecular & Clinical Medicine. ISSN 2515-8260 Volume 07, Issue 03, 2020 Available: https://ejmcm.com/pdf_5172_967a0564c5efcd9ca4d9ca59189e807a.html

[17] Hancock, J.T., Khoshgoftaar, T.M. Survey on categorical data for neural networks. J Big Data 7, 28 (2020). https://doi.org/10.1186/s40537-020-00305-w

[18]Sola, J. & Sevilla, Joaquin. (1997). Importance of input data normalization for the application of neural networks to complex industrial problems. Nuclear Science, IEEE Transactions on. 44. 1464 - 1468. 10.1109/23.589532.

Kaggle Titacin Challenge can be found here: https://www.kaggle.com/c/titanic

Distribution of Work

Edward

- SVM & Logistic Regression descriptions, literature review and code
- Pre-processing and cleaning of data
- Data Exploration
- Evaluation and discussion of report
- Speaking on video and presentation of video

Andrew

- Ann & Random Forest descriptions, literature review and code
- Pre-processing and cleaning of data
- Implementation section of report

Alana

- Naive Bayes & KNN descriptions, literature review and code
- Pre-processing and cleaning of data
- Introduction, conclusion and formatting of report