CONL708: Applied Machine Learning Project - Report

Word Count:1,972

# Abstract

In the recent decades, Machine Learning (ML) has gained much popularity as a subfield of Artificial Intelligence. This report evaluates three types of Machine Learning Models against a sample dataset to examine the outcome. In addition, the development of code provided in conjunction with literature reviews from disparate sources determining such possibilities. Current findings indicate that K-Nearest Neighbours amongst the three performed the strongest, caveating that the other models only contained default configuration. Furthermore, more testing should operate against both configuration of the ML models, and against other datasets; enabling to gain more insight into scenarios these ML models would perform most effectively within.

**Keywords—Machine Learning, K-Nearest neighbours, Logistic Regression, Support Vector Machines, model, code, python.**

# Introduction

Machine Learning (ML) develops algorithms to identify and extract patterns from data [1], to learn without additional programming, autonomously [2,4]. Thus, this report’s purpose aims to capture how Machine Learning can run against a particular dataset with a goal of applying three distinct variants of ML models to help answer a question and see how these models compare with one another. Firstly, examine the selected dataset. Secondly, elaborate on the problem or question to answer. Thirdly, document pre-processing and feature extraction steps against dataset before ML model initialisation. Additionally, create three ML models against the dataset and evaluate its performance. Finally, compare all three models against one another before providing a summary and recommendations going forward.

# Dataset to use

The dataset used for this report, derives from Kaggle, known as the Titanic dataset [3]. The Justification for this, derives from the popularity within Kaggle, particularly with competitions to apply different Machine Learning Models to achieve the highest accuracy, as well as much documentation available ways dataset is useable. The files used from this will be the train.csv and test.csv. The train.csv will drive training all three models, whereas the test.csv file will see the predictions the models make on new data if applicable.

# Problem to solve

With the dataset mentioned, the aim of this report is to classify which passengers survived the Titanic disaster of 1912 [9] or perished. Therefore, the report emphasis will focus on ML concepts of classification – a form of supervised learning where observations are labelled belonging to a class out of n classes [5-8], as the goal here is to establish how many passengers survived (1) or passed away (0) at such event.

# Three Machine Learning Models to apply on dataset

All three models deploy classification onto the titanic dataset to measure the prediction of how many passengers survived. For all three models chosen, all code produced and executed within python.

## k-Nearest Neighbours (KNN)

knn models work by measuring similarity or proximity between data points [10] and can use this method to place new data points into relevant categories based on k nearest data points found in the training data [11-12]. This form of model uses full training data saved in memory to perform predictions [13]. The rationality being that it is easy to interpret and process [14].

## Logistic Regression

Logistic Regression is method for performing binary classification, and outputs the probability or likelihood between classes using the sigmoid function [15]. Logistic Regression is useful, as can accept both discrete and continuous values as inputs, can provide a qualitative output, splits data via a hyperplane to form the classes [16-17]. Reasoning here is that it is interpretable and achieves reliable performance on straightforward datasets.

## Support Vector Machines (SVM)

SVM’s are like Logistic Regression for use in classification but uses a different boundary line to classify the data points [18]. The rationale for using this model as can analyse complex data well, and less onus of outliers disrupting the data [19]. Additionally, SVM’s are like the methods above but done differently.

# Data Pre-processing and Feature Extraction steps

The following screenshots and information taken from **Appendix One** show steps taken to load the data, perform pre-processing activities, and ensure a dataset enriched with features, before applying all three ML models to display the results.

## Import relevant libraries

The following screenshot shows the packages used in the python code found in **Appendix One** for both titanic dataset and all three ML Models.

Graphical user interface, text

Description automatically generated

Figure 1 - code showing packages used in python code

## Load in data

The Titanic Dataset downloaded from Kaggle [3], onto the local machine before loading into python.

Table

Description automatically generated

Figure 2 - Code showing dataset loaded into python

## Exploratory Data Analysis (EDA) Report

After the titanic dataset loads into python, an EDA report generated using the package pandas profiling [20] to assist examining data in further detail to advise what pre-processing required before conducted in later steps; as shown in **Appendix Two.**

## Split Training data

First, separation of the target variable (Survived) into a new variable (Y) incorporated later for use in the modelling phase; away from the independent variables, which will go into another variable (titan\_train).

Graphical user interface, text, application

Description automatically generated

Figure 3 code showing splitting of training data.

## Removal of columns

From an initial glance, there appear to show columns in both datasets that are unfeasible, particularly Name, Ticket, and PassengerID. An argument potentially that Name could show indicators of survival, such as Doctor or Reverend, but may include bias at this stage, so will remove.

A picture containing rectangle

Description automatically generated

Figure 4 - Code showing removal of columns

## Conversion of Sex with one-hot encoding

The next attribute to deal is with Sex; appearing to be a potential important feature for our modelling purposes. However, unusable in its current format. Therefore, we can employ the use of one-hot encoding to convert into a numeric binary value. The new values will show as 1 (male) and 0 (female.)

Text

Description automatically generated with medium confidence

Figure 5 - Code showing one-hot encoding of Sex column

## Conversion of Cabin into a binary output

Now to focus on the Cabin attribute, we have seen from the EDA report in **Appendix Two** that there appears to be a mixture of passengers assigned or recorded to have a cabin, and those who did not. For this reason, it seems inadvisable to remove the data, as it could prove to eb an important feature the models may consider as to who would survive the disaster. With that in mind, instead of removing any data here, the aim is to convert it to a binary value with having a cabin (1) or not (0). The first step would be to convert any NULL values into a zero, and observations with cabin numbers assigned to one.

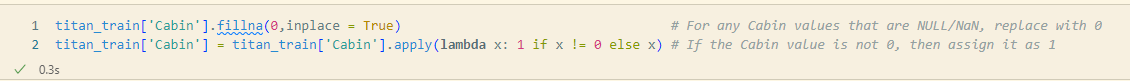
****

Figure 6 - Conversion of Cabin Column

## Managing the missing values within Age

As seen from the data, there are values missing from the Age column within the dataset. Therefore, in the first instance, rather than removing the rows with missing values - instead, data imputation can provide the mean value across the values to fill in these gaps. This ensures as much of the available data is useable, without losing information that could model upon.



Figure 7 - Code dealing with Age values.

## SibSp and Parch Columns

Based on two attributes found in the dataset, they are similar in nature. SibSp - represents the number of siblings or spouses aboard with the passenger; whilst Parch describes the number of parents or children aboard with the passenger. Therefore, these two attributes can reform into one feature known as family size, to reduce cardinality slightly. One has added to result, to represent a passenger traveling alone otherwise.

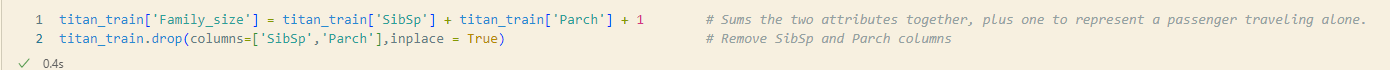


Figure 8 - Feature Engineering Family\_size

## One-hot encoding the Embarked Column

The second to final pre-processing step to take before the modelling phase can begin. With the Embarked column - that represents the port in which the passenger boarded the titanic from- either Southampton in the UK (S), Cherbourg in Normandy (C), or Queenstown - now known as Cobh in Ireland (Q). However, in its current form, most models would not be able to take these categorical values as they are, but through one-hot encoding once again, this is achievable.

Text

Description automatically generated

Figure 9 - One-hot encoding of Embarked.

## Scaling

The Last step in pre-processing, which involves scaling all input values to standardise them [21].

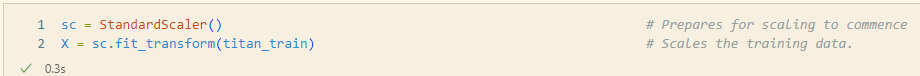


Figure 10 - Standardising training data.

## Prepare Train test split for the titanic train dataset

Now all the pre-processing is complete, the data from the train.csv file splits into training and testing datasets to run the three ML models against. Note that data from test.csv will perform predictions where model(s) have not seen the data previously. Train.csv shall split into 67% used for training the model, and 33% to test model performance.



Figure 11 - Train test split on Titanic dataset.

# K-NN Performance Evaluation

Based on a method conducted by Goel [22], a for loop iterates between values 1-20, to see what number is most optimal to provide as k – number of nearest neighbours to apply in the model.

. Graphical user interface, text, application, email

Description automatically generated

Figure 12 - Code to find the most optimal model based on k.

Based on the above executed code, results store into a dataframe, showing k used, including accuracy score and confusion matrix against it.

Table

Description automatically generatedTable

Description automatically generated

Figure 13a and b - Dataframe storing results through KNN model iterations, and chosen model

Based on iterations above, it appears that k = 6 with an accuracy score of circa 84.40% is the most optimal number to provide as part of the model build. Considering this, configuration can now apply to make the model formally.

# Logistic Regression Performance Evaluation

Now to build a model showing performance on training data using Logistic Regression.

Table

Description automatically generated

Figure 14 - Model creation and performance of Logistic Regression.

Based on predictions of the logistic regression model, the accuracy score comes in at approximately 79.66%, with AUC score of 77.5% as seen in the ROC curve chart below [24].

Chart

Description automatically generated

Figure 15 - ROC Curve of Logistic Regression Model

# SVM Performance Evaluation

Finally, to create a model on the training data, via the use of SVM.

Table

Description automatically generated

Figure 16 - Model creation and performance of SVM

Based on the predictions of the SVM model, the accuracy score comes in at approximately 83.38%, with an AUC score of 81.42% as shown below [24].

Chart

Description automatically generated

Figure 17 - ROC Curve for SVM Model

# Comparison between all Model methods

Based on pure use of the default configuration across the three models, SVM performed more favourably with an accuracy score of 83.38%, with the lowest being Logistic Regression at 79.66%.. Even with KNN being ran only once on the default number of neighbours being five [23], this still would have brought an accuracy rating of 82.37%. Overall, KNN performed the best on the titanic dataset. However, KNN is different to compare against the others, compared with Logistic Regression vs SVM, due to the comparison of ROC curves. However, bear in mind that the KNN model had the optimal configuration selected for accuracy, whereas the other models where used based on their default settings only.

# Conclusion

In this report, an examination undertaken of using a famous dataset - the Titanic dataset of passengers who survived the event in 1912; and to apply a set of three different Machine learning models against it, to determine which model would be best at predicting who survived or not. After evaluating all three, this report concluded – given the scenario presented – that K-Nearest Neighbours favoured better in terms of accuracy based on the configuration used; followed by Support Vector machines and Logistic Regression in that order. However, this would not determine the type of ML model to use given other scenarios, and remains a trial and error approach of the right model, given the problem to solve- for instance where K-Nearest Neighbours has fared well here, it may not so in future if against a large dataset due to the model storing the training data in memory – whereas the other two methods may have performed better. Additionally, recommendation would be to find the optimum parameters to use for all three models to see if this provided a different image converse to the current.

# Appendices

## Appendix One – Code to Build ML Models and evaluate performance



## Appendix Two – Pandas Profiler EDA Report



# References

[1] J. D. Kelleher and B. Tierney, Data science. Cambridge, Massachusetts ; London, England: The Mit Press, 2018, p. 97.

[2] O. Theobald, Machine learning for absolute beginners : a plain English introduction. United States: The Author, 2017, p. 8.

[3] Kaggle, “Titanic - Machine Learning from Disaster,” kaggle.com. https://www.kaggle.com/competitions/titanic/overview (accessed Sep. 28, 2022).

[4] S. Sedkaoui, Data analytics and big data. London: Iste Ltd / John Wiley & Sons, Inc, 2018, p. 109.

[5] Kirill Eremenko, Confident data skills : master the fundamentals of working with data and supercharge your career. London: Kogan Page, 2018, p. 105.

[6] S. Sedkaoui, Data analytics and big data. London: Iste Ltd / John Wiley & Sons, Inc, 2018, p. 133.

[7] J. Bell, Machine learning : hands-on for developers and technical professionals. Indianapolis: John Wiley & Sons, 2015, pp. 140–141.

[8] A. C. Müller and S. Guido, Introduction to machine learning with Python : a guide for data scientists. Beijing: O’reilly, 2017, pp. 27–28.

[9] M. Crooks, “Apr 15, 1912 CE: Titanic Sinks | National Geographic Society,” education.nationalgeographic.org, May 20, 2022. https://education.nationalgeographic.org/resource/titanic-sinks (accessed Sep. 28, 2022).

[10] N. Abraham et al., Coding all-in-one. Hoboken, Nj: John Wiley & Sons, Inc, 2017, pp. 522–523.

[11] Kirill Eremenko, Confident data skills : master the fundamentals of working with data and supercharge your career. London: Kogan Page, 2018, pp. 112–113.

[12] S. Sedkaoui, Data analytics and big data. London: Iste Ltd / John Wiley & Sons, Inc, 2018, pp. 134–135.

[13] O. Theobald, Machine learning with Python : a practical beginners’ guide. Erscheinungsort Nicht Ermittelbar] Scatterplot Press, 2019, pp. 118–119.

[14] A. C. Müller and S. Guido, Introduction to machine learning with Python : a guide for data scientists. Beijing: O’reilly, 2017, p. 21.

[15] S. Sedkaoui, Data analytics and big data. London: Iste Ltd / John Wiley & Sons, Inc, 2018, p. 136.

[16] O. Theobald, Machine learning with Python : a practical beginners’ guide. Erscheinungsort Nicht Ermittelbar] Scatterplot Press, 2019, pp. 97–98.

[17] Kirill Eremenko, Confident data skills : master the fundamentals of working with data and supercharge your career. London: Kogan Page, 2018, pp. 126–135.

[18] O. Theobald, Machine learning for absolute beginners : a plain English introduction. United States: The Author, 2017, p. 79.

[19] O. Theobald, Machine learning with Python : a practical beginners’ guide. Erscheinungsort Nicht Ermittelbar] Scatterplot Press, 2019, p. 110.

[20] pandas-profiling, “pandas\_profiling API documentation,” pandas-profiling.ydata.ai, 2022. https://pandas-profiling.ydata.ai/docs/master/index.html (accessed Sep. 30, 2022).

[21] O. Theobald, Machine learning with Python : a practical beginners’ guide. Erscheinungsort Nicht Ermittelbar] Scatterplot Press, 2019, p. 121.

[22] E. Goel, “Applying 7 Classification Algorithms on the Titanic Dataset,” Medium, Jul. 01, 2021. https://medium.com/geekculture/applying-7-classification-algorithms-on-the-titanic-dataset-278ef222b53c#id\_token=eyJhbGciOiJSUzI1NiIsImtpZCI6ImJhMDc5YjQyMDI2NDFlNTRhYmNlZDhmYjEzNTRjZTAzOTE5ZmIyOTQiLCJ0eXAiOiJKV1QifQ..WPMb029EbGI5tk\_gATlEzauV765ECzdGUviSjscCMIJBQFAkT1lo\_-tuSGIPz3rsBJ1UqZJQQHTHG3fXTfy6ujnSwJwpNTz0Tgj0MJ-H2KDTVj8YV\_6TFoG0osUTVE32RNfN-VRWot8T8-n6Gox5tkaq\_8P4Upsr4VCqtZ2JR72YWUp1KU3Ef5OdsPledZVZdCUfb\_a09D5yLPdh-EbxBU4E5c0P59B7FfxyGdlq\_hDYGAqB33q1P3oygb8lorJ5CLEmNWPeJIU4QCLL9LeU-RTGGfFDvJhueV\_MRtVks1nRh3fNBNjr947T-nLze4tNuOyYk8GWfPupty4mT7nnuQ (accessed Sep. 30, 2022).

[23] scikit-learn developers, “sklearn.neighbors.KNeighborsClassifier — scikit-learn 0.22.1 documentation,” Scikit-learn.org, 2019. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html (accessed Sep. 30, 2022).

[24] plotly, “Roc,” plotly.com, 2022. https://plotly.com/python/roc-and-pr-curves/ (accessed Sep. 30, 2022).