# Group ID - MSc in Data Analytics

Author: G.Davis

e-mail: sba22311@student.cct.ie

Student ID: sba22311

**Title: An analysis of DCC travel time data to obtain predictive models for travel times in Dublin City using a machine learning approach**

**Abstract**

Travel time across the city is of great interest when understanding the day to day running of city roads. How we can properly plan roadworks and maintenance to prevent disruption to the flow of traffic, increasing travel times on city routes where travel times may already be excessively high, is of great importance. Past data provided by the Dublin City Council on travel times across the city from the intelligent TRIPS system can provide valuable insight into travel times on particular routes in Dublin City. The TRIPS data was harnessed to generate a model which could accurately predict the travel times expected in particular areas in order to prevent traffic diversions, causing increased time in traffic and congestion in the city centre, leading to a vast array of other city wide issues. Thus this study has harnessed travel times from the TRIPS system dataset from 2019 provided by the Dublin City Council with the aim of generating machine learning models to predict travel times at particular routes in order to correctly plan roadworks, parades, marathons or city events which could otherwise disrupt the flow of the population into the city centre rendering alternate routes congested, travel times unacceptable, and disrupt the flow of the population through Dublin City. The data was explored and critically analysed, before processing and implementation of a number of machine learning models. The model which yielded the highest performance was Random Forest Regressor yielding R2 values for training, 0.98, and testing ,0.88 , along with the lowest RMSE, 67.79 and MAPE, 21.94.

**Abbreviations**- AccSTT (accumulated smoothed travel time), DCC ( Dublin city council), RMSE (Root mean squared error), MAPE (mean absolute percentage error), ML (Machine learning), EDA (exploratory data analysis), RF (random forest), KNN (K-nearest neighbors), DT (Decision Tree)

**Contents**

1: Introduction

2: Methodology

2.1: Approach CRISP-DM

2.2: Programming tools

2.3: Statistics

2.4: Data preparation and visualisation

2.5: Machine learning

3: Results

3.1: Description of Trips datatset

3.2: Statistical analysis

3.3: Data visualisation

3.4: Data preparation for machine learning

3.5: Machine learning

3.5.1: Reverse feature selection

3.5.2: Alternative machine learning model comparisons

4.Discussion

5.Conclusion

6.Bibliography

**1.Introduction**

The prediction of travel times is an area of great interest, especially in light of the development of intelligent machine learning (ML) algorithms which can be applied to predict future events. The requirement to better understand and accurately predict travel times has a number of benefits for the planning and functioning of a big city. In light of this, many publications have focused on the application of machine learning models on past traffic data with the aim of predicting areas with the highest travel times, traffic levels, potential for congestion and even the likelihood of accidents (Zhang *et al.*, 2013; Liu and Wu, 2017; Jiber *et al.*, 2020; Qiu and Fan, 2021; Balamurugan *et al.*, 2022). These studies have harnessed an array of ML models such as K-nearest neighbour (KNN), linear regression (LR) and random forest (RF), to name a few. With these intelligent prediction models in place the planning of roadworks and city events can be organised with a lesser effect on perturbing the normal flow of traffic in a city, leading to increase travel times. In this study, ML models have been applied to the Trips dataset published in 2019 from the Dublin City Council (DCC) with the aim of generating accurate predictive models for journey times across the city centre at a number of specified routes in Dublin City. The appropriate exploratory data analysis (EDA) on the dataset was performed in order to understand the data in question, followed by data preparation for insertion into different ML models. Prior to applying ML models, statistical analysis was performed on the dataset to better understand the spread of the data and understand how the target variable chosen, AccSTT, was distributed. The focus of this study was to prepare, visualise and understand the Trips dataset provided by the DCC and apply suitable ML models.

Within this study, the Trips datasets supplied by the DCC has been used which contains travel times across Dublin city (trips.csv). In depth analysis was performed, and where appropriate suitable ML models have been applied in order to optimize and develop methods to predict various parameters from the available data. As will be highlighted, some ML models are more suited to the dataset than others, and some parameters of the dataset are more or less important in these prediction models. It is also clear from this study that the data collected by the DCC leaves much to be desired in terms of granularity and depth when trying to harness this dataset to generate ML models.

To understand and prepare the data for ML models, statistical analysis and EDA was carried out to develop an approach on how best to harness this dataset for prediction, and which parameters were most suited to move forward with. This study follows the CRISP-DM approach to data mining which has been widely used as an industry standard approach to data mining, evaluation, prediction and deployment (Schröer, Kruse and Gómez, 2021). Although this study is not developed from a business stand point the benefit to business, infrastructure and transport is heavily weighed, and thus CRISP-DM was chosen as the most thorough approach, considering the DCC as the business.

**Data source:** <https://data.gov.ie/organization/dublin-city-council?tags=Transport+and+Infrastructure>

**2.Methodology**

**2.1.Approach-CRISP-DM**

The CRISP-DM approach is widely used as the industry standard set of criteria for carrying out a data mining project. It is the most widely used and complete methodology to carrying out a data mining project in comparison to SEMMA or KDD, and was thus selected for this study (Schröer, Kruse and Gómez, 2021).

**Diagram

Description automatically generated**

**Figure 1: A schematic representation of the approach taken for this study using the CRISP-DM methodology.** 1-Business understanding describes getting an overview of the business and/or organisation in question, describing the projected project goals and expected outcome. 2-Data understanding describes collecting the data, exploring it and describing it using statistical analysis and visualisation. 3- Data preparation describes preparing the data for modelling by cleaning and feature engineering making the available data suitable to be used in ML models. 4-Modeling describes choosing an appropriate model that that fits the initial question and the gained understanding of the data explaining the choice and the parameters set. 5-Evaluation describes evaluating the results and discussing them in line with the objectives of the projected outcome. 6-Deployment presented as a final report or software component, including the plan for deployment and how to monitor and maintain (Schröer, Kruse and Gómez, 2021).

**2.2.Programming tools**

For statistics, data preparation/visualisation, and machine learning anaconda navigator was used along with Jupyter notebook as a coding interface. The language of choice for the project was python. All code files and outputs are provided alongside this report. File names: TripsDataAnalysis.ipynb

**2.3.Statistics**

As previously stated, statistical analysis was performed in Juypter notebook using python. Python libraries such as scipy.stats and math were used in the analysis of datasets while also harnessing the inbuilt functions which python provides to ascertain statistical measures such as mean, max, min, mode, variance and standard deviation. Poisson and Normal distributions were used to gain statistical understanding of certain parts of the data and highlight the distribution of the chosen target variable, AccSTT.

**2.4.Data preparation and visualisation**

For data preparation and visualisation python libraries pandas, numpy, matplotlib.pyplot and seaborn were used along with in-built python functions to visualise and prepare data in order to gain an understanding of the data to generate appropriate questions for this study and to prepare visualisations such as bar plots and heatmaps to ascertain any trends within the data that could be seen.

**2.5.Machine learning**

Given the data provided in datasets was labelled a supervised learning approach was used. Supervised machine learning is used for labelled datasets, given the Trips dataset had labelled features, the supervised ML approach was deemed to be suitable. The models harnessed in this study were chosen according to the available literature on traffic prediction. In light of this KNN, RF, LR, along with DTRegressor, KNRegressor and RFRegressor were used to develop predictive models from the DCC datasets. From a comparison standpoint, Naïve bayes classifier and Support Vector Machine were also applied.

The KNN model is considered one of the more simple models and isa non-parametric technique developed using a training dataset. The model then makes predictions on the test element by calculating the closest neighbours to that point and making a classification based on the assumption that the closest neighbours to the test element are the same, hence labelling the test element as the closest neighbours (Zhang *et al.*, 2013).

Random Forest (RF) is another widely used classification model for supervised machine learning. It is based on decision trees and was developed as a stronger model to reduce overfitting of decision trees. The RF model works by building many trees and averaging the results (Müller and Guido, 2016). This model is preferred in many cases to reduce overfitting often seen with Decision Tree algorithms.

Linear regression is another highly implemented supervised ML model. It is based on linear regression and the equation of a line. Variables which have some correlation can be used for prediction along with other independent variables. In simple linear regression one variable is correlated with another and prediction of one can be made from the other (Müller and Guido, 2016) . Multiple regression given a number of independent variables is often the case in machine learning implementation, using a number of features to generate multidimensional regression analysis to predict the target variable, with a number of independent features. There are a number of types of linear regression models namely, simple Linear regression, multiple linear regression, Lasso regression, Ridge regression. Lasso and Ridge regression impose a penalty factor on the coefficients which reduces the model complexity and the advent of multicollinearity. Lasso and Ridge regression differ as Lasso takes the magnitude of the coefficients and Ridge takes the square. There are also regression based versions of Decision Tree, Random Forest and K-nearest neighbors ML models which apply the same methodology as described about to regression models (Müller and Guido, 2016).

**3.Results**

**3.1.Description of Trips dataset**

The Trips dataset (“trips.csv” ) is a dataset provided by the DCC from the DCC TRIPS system. The journey times are recorded from a number of routes across Dublin city. The routes consist of a number of links which is a georeferenced traffic control site provided in “sites.csv”. The original trips.csv data set consists of 7 columns which are detailed in the centralised document provided with the dataset. These columns are: # Route, Link, Direction, STT, AccSTT, TCS1 and TCS2. # Route denotes the route, Link denotes the link between sites, Direction denotes the direction of movement, STT is the smoothed travel time, AccSTT is the accumulated travel time, TCS1/2 are the traffic control sites. The total size of the dataset is 902 rows with the 7 columns listed below.

Table

Description automatically generated

**Table 1: Showing the description of the column headings in the Trips dataset**

Preliminary data exploration highlighted that all the data was numeric of integer type. There were no null or duplicate values present in the dataset meaning removing null values or duplicates was not necessary for this particular dataset. AccSTT as the accumulated travel time was taken as the dependent or target variable for the study with the aim of identifying potential travel times using ML models for particular routes and links also harnessing the power of the other independent variables, Direction, STT, TCS1 and TCS2.

**3.2.Statistical analysis**

***Statistical overview*:** The data set was statistically examined using in-built python functions such as .describe shown in Table 2, along with scipy.stats and matplotlib.pyplot for distributions and statistical visualisations. As shown in Table 2 all columns consisted of 902 values with a minimum value of > 1. There were 50 unique values for routes and 33 unique values for Link. Direction had a value of either 1 or 2. There was substantial standard deviations for STT and AccSTT of 45.47 and 235.47 given the mean values of 39.26 and 276.22 respectively. This indicates a large spread in the data for smoothed and accumulated travel times also confirmed by the min and max values (STT min= 4.0, max =731.0, AccSTT min = 7.0, max = 2222.0).

**Table

Description automatically generated**

**Table 2: Summary statistics of the Trips Dataset**

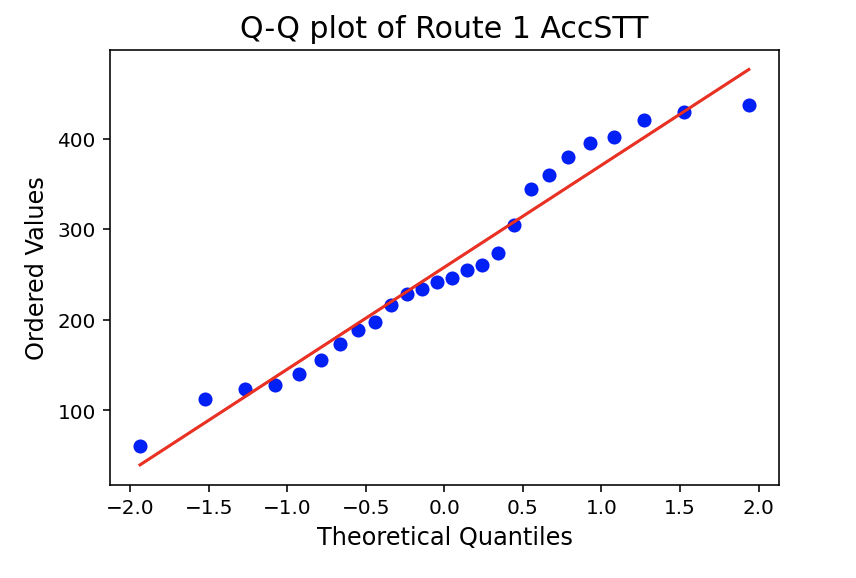
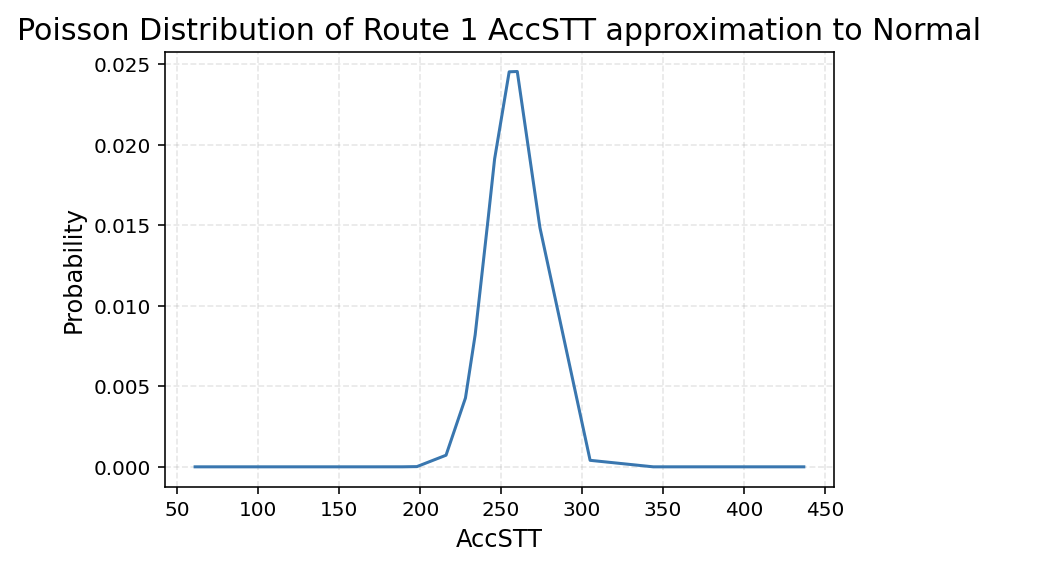
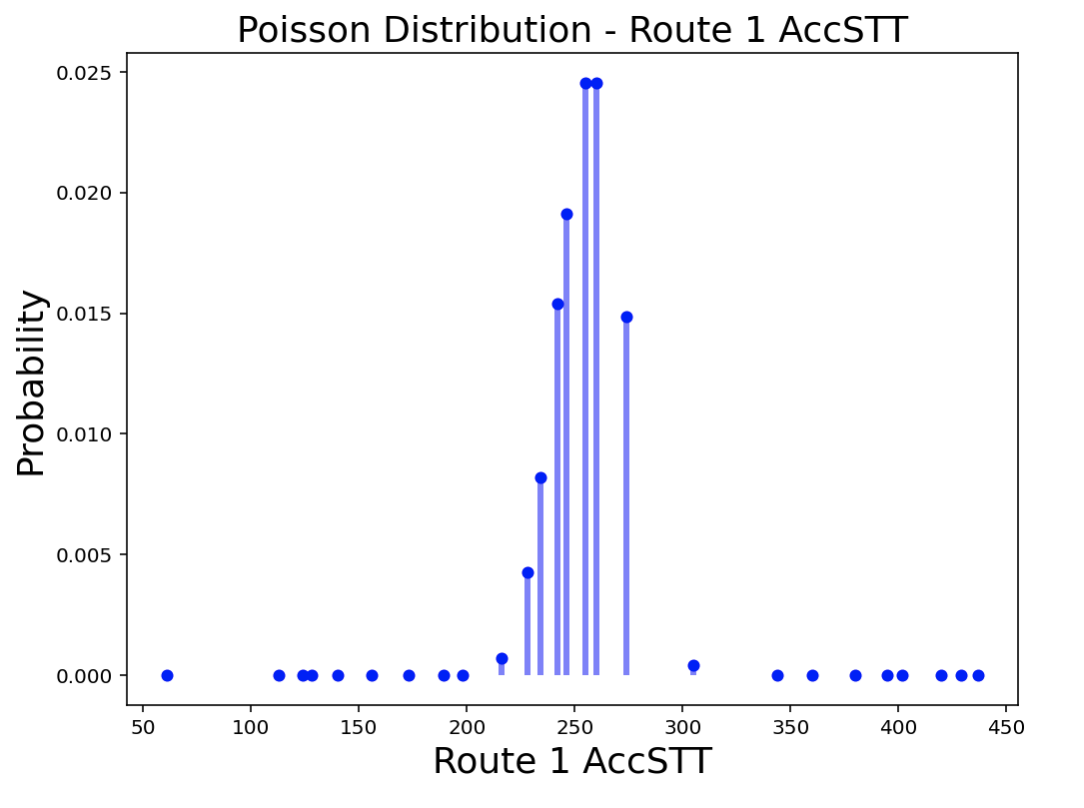
***Poisson distribution:*** To better understand the spread and weight of the data for AccSTT from the 50 routes they were plotted as a histogram shown in Figure 2. As AccSTT is a discrete random variable and the accumulated travel time can be any value with no defined limit, a poisson distribution was chosen to understand the probabilistic outcome of certain events at different routes. For example, it was determined what the probability was if the AccSTT exceeded 100 on Route 1. To do this a poisson distribution was employed and the AccSTT for Route 1 was isolated with the corresponding probability mass function (pmf) value calculated for each value of AccSTT for Route 1, of which there were 26, shown in Figure 3A.

Chart, histogram

Description automatically generated

**Figure 2: Histogram showing the distribution of AccSTT**

As shown in Figure 2 AccSTT values are heavily skewed right meaning the highest amount of values are approximately below 500. To gain a stronger statistical understanding of the AccSTT data a poisson distribution was applied, shown in Figure 3A. We can see in Figure 3A the poisson distribution of the AccSTT variable generated by calculating the probability mass function (pmf) of the AccSTT values at Route 1, which is the probability that a discrete random variable is equal to a value. The AccSTT variable was fitted to a poisson distribution as the values for AccSTT could be of any value despite knowing that the values themselves must fall within a certain range for travel times of vehicles on a particular route and link.



A

B

C

**Figure 3: Poisson distribution, , Normal approximation and Q-Q plot of Route 1 AccSTT.**

As shown in Figure 3A a poisson distribution can be generated for the AccSTT values by calculating the pmf of each value, ordering the data, generating the resulting distribution. At first glance the distribution looks to be normal, this will be discussed in the next section. Taking Route 1, as an example of the AccSTT variable, not forgetting the other 49 routes, a statistical understanding of the probability of certain AccSTT values occurring at Route 1. The overall mean of AccSTT is 276.22 and so the probability of an AccSTT occuring over this time as calculated using poisson with the Route 1 mean (or lambda) of 258.04. This probability was 0.126 meaning there was a 12.6% chance that AccSTT would exceed the mean at Route 1. In the context of the overall data this could mean that Route 1 has below average accumulated travel times. Appling the same logic and poisson the chance of an AccSTT below the overall AccSTT mean for the 50 routes at Route 1 was 0.874, meaning 87.4% of the values at Route 1 are below the overall mean for AccSTT. Taking the example of Route 1, it indicated that some Routes may have higher values, further indicating that accumulated travel times may be longer at one or more of the 50 Routes.

***Normal Distribution:*** As seen the poisson distribution approximates the normal distribution of AccSTT at Route 1. It is said as the mean increases the poisson distribution approximates a normal distribution (Downey, 2014). Thus by fitting a curve to the Poisson discrete probability distribution in Figure 3B we can see that the distribution of AccSTT at Route 1 in this example can approximate the normal distribution given its shape and the high value for the mean of AccSTT at Route 1. To further validate the normal distribution approximation a Q-Q plot was generated showing the variable is in fact normal as it follows theoretical quantiles at a 45 degree angle highlighting the data is normally distributed. The stats.probplot() generates theoretical quantiles for the variable thus highlighting its normality as it follows the 45 degree angle shown in Figure 3C.

**3.3.Data visualisation**

**Graphical user interface, application

Description automatically generated**

**Figure 4: Heatmap of feature correlations from Trips dataset.**

Chart, scatter chart

Description automatically generated

**Figure 5: Scatter plot with a line of best fit for AccSTT and Link highlighting positive correlation.** Red box marks extreme outliers from the positive correlation between AccSTT and Link.

Chart, bar chart

Description automatically generated

**Figure 6: Bar plot of AccSTT values for each Route**. It can be seen as that the extreme outliers present in Figure 5 are likely to be located at Route 44. The location of these outliers was confirmed in the TripsDataAnalysis.ipynb file Section

**3.4.Data preparation for machine learning**

After initial data exploration and visualisation, AccSTT was chosen as the target variable. The aim was to prepare the data in such a way that it could be implemented into a suitable ML model for the prediction of accumulated travel times for different Routes/Links. As stated previously the data was numerical of type integer, with no missing values or null values.

In section 3.3 using the correlation heatmap there is a significant correlation between AccSTT and Link. In general we can say that AccSTT values are positively correlated with the Link value with a correlation coefficient of 0.74. In light of this a linear regression approach was chosen to attempt to generate a predictive model for AccSTT from the other labelled features such as Route, Link, STT, TCS1/2. In doing so, this study aims to generate a model with the potential of predicting accumulated travel times for particular routes and links for traffic moving in a particular direction.

In order to prepare the data for machine learning the appropriate EDA to explore and understand the data was performed. Following this, by the use of boxplots, the spread of the data was visualised to examine for outliers for the independent features (Section 1.4 ipynb file). As shown in Figure 6 there is quite a variance in the data for AccSTT for different Routes. For AccSTT, STT, TCS1 and TCS2 winorization was applied to reduce the effects of potential outliers such as those that were identified at Route 44. After winorization, given the large variance in values between each of the features from movement values of 1 or 2 to TCS1/2 values of 2 to 6032, the data was scaled using the standard scaler function from sklearn, thus reducing the effects of large differences of value scales between different features. The standard scaler function does not change the underlying trend in the data but reduces the magnitude of numbers by subtracting the mean and dividing by the variance. The scaled features has a mean of 0 and a variance of 1. An example of the scaled TCS1 values is shown as a histogram in Figure 7.

Chart, histogram

Description automatically generated

**Figure 7: Histogram of TCS1 values after standardization.**

**3.5.Machine learning**

To apply the refined data to a machine learning model, X and Y variables were allocated with Y being defined as the AccSTT target variable and the other independent features allocated to X ( Route, Link, STT, Direction, TCS1, TCS2). After defining X and Y, train test split from sklearn was used to split the data into training and testing data with a test size of 0.3 meaning 30% of the data was used for testing the model and an initial random state of 80. Random state is a hyperparameter which controls the shuffling across different executions of the training and testing data (Müller and Guido, 2016). Given 30% of the data was used for testing, 70% was used for training and fitting the model. Sklearn Linear regression was used to apply a Linear, Ridge and lasso regression to the training and testing sets. All iterations produced similar R squared values, root mean squared errors (RMSE) and mean absolute percentage error (MAPE), as shown in Table 3, despite the imposing of penalty factors with Ridge and Lasso regression. These outputs indicate the strength of the model and its accuracy in prediction harnessing the supplied training and testing sets. The R squared value indicates how well the model is fitted to the training and testing sets, while RMSE measures the error which is the distance of the actual versus predicted values, where MAPE is a the percentage error in prediction(Müller and Guido, 2016). There are also regression models for K-nearest neighbors (KNRegressor), Decision Trees (DTRegressor) and Random Forest (RFRegressor). These hybrid models apply the regression and following this the algorithm is performed as it is in each respective model, for example, in the case of KNRegressor, multiple regression is applied and then Euclidean distance is used to classify the point along with the number of nearest neighbours, thereby predicting the value. Table 3 shows a summary of R squared, RMSE and MAPE for the different regression models. Highlighted in green in Table 3 shows that the RFRegressor produced the best model out of all the regression models used, followed by DTRegressor and KNRegressor. RFRegressor displayed the highest R squared value for train (0.98) and test sets (0.88), along with the lowest RMSE (67.79) and MAPE (21.66). These accuracy parameters for the RFRegressor on the Trips dataset indicated that it was the best regression model for the data as indicated by the R2 values for the training and testing data, along with the RMSE and MAPE. The DTRegressor also displayed high R2 values for training (1.0) and testing (0.75), however an accuracy of 1 suggests that the model is being overfitted with the training data, therefore RFRegressor was chosen as the most accurate.



**Table 3: Summary of R squared, RMSE and MAPE for different regression models.**

Given the relatively high accuracy of the linear models, particularly the RF and DTregressors these models were chosen as the most suitable models for the AccSTT variable prediction. To further confirm the accuracies of these models K-fold cross validation was performed on each model. K-fold cross validation is a method used to test how different splits of the data and features effect the accuracy of the model (Müller and Guido, 2016). The cross validation results from the two best performing models are shown in Figure 8 along with the linear regression cross validation results for comparison. The results in the K-fold cross validation using 5 spli-

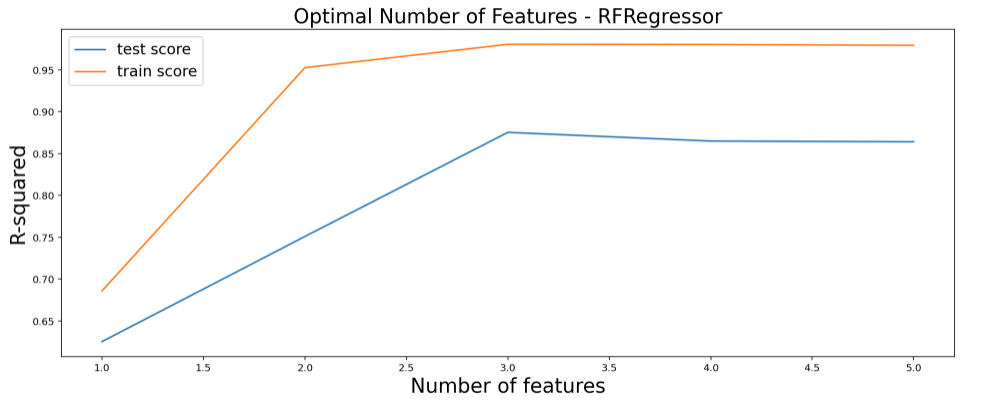
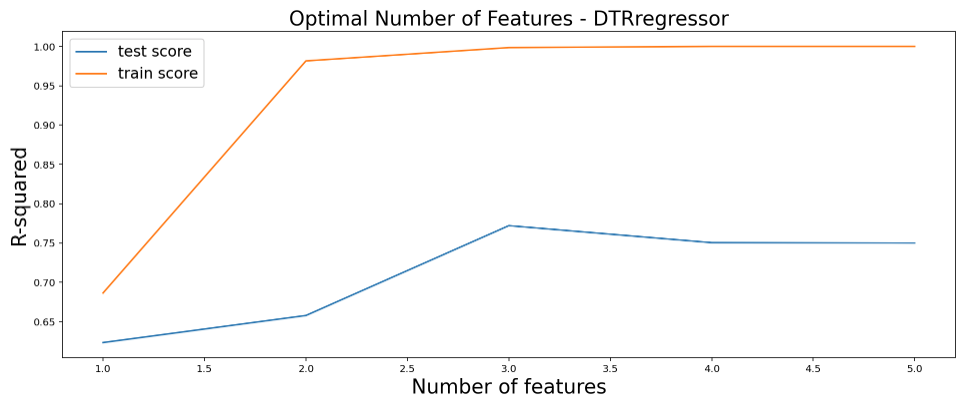
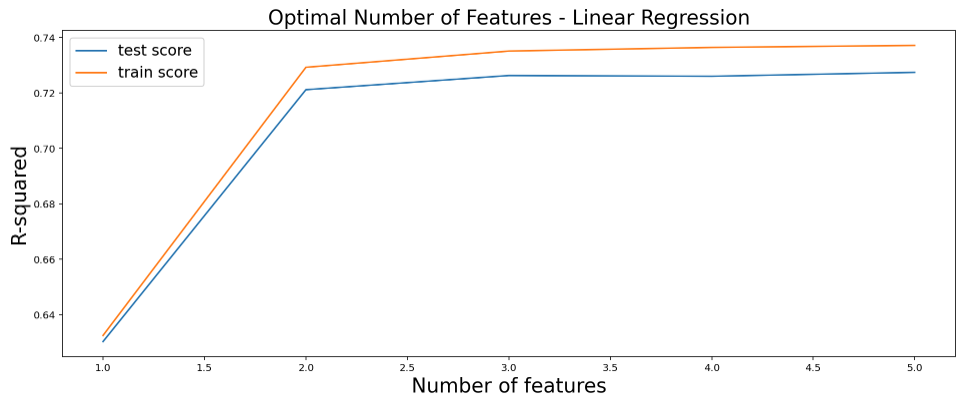
ts indicate good reproducible accuracy across different splits in the model which is promising in terms of its accuracy. K-fold cross validation confirmed that the high accuracies of the RF and DTRegressors were not a random occurance due to a particularly split of the model and variations in the splits of the data still produce well fitted models with similarly high accuracies. Using 5 splits for K-fold cross validation it can be seen that the RFRegressor Train and Test R2 values remain relatively stable indicating the model performs well across different splits of the data. The optimal number of features was determined to be 3 for both the RFRegressor (R2-0.88) and DTRegressor models (R2-0.73) after performing cross validation and generating the final models. The optimal number for features as shown by the

accuracy visualisation in Figure 8 was chosen to be 3.

A

B

C



**Figure 8: K-fold cross validation of Linear Regression, DTRegressor and RFRegressor.**

**3.5.1 Reverse Feature Selection**

Reverse feature selection is a useful way of understanding the importance of certain independent features and their contributions to a models accuracy. In light of this, independent features: # Route, Link, Direction, STT, TCS1 and TCS2 were sequentially removed and the different models accuracies were assessed as seen in Table 4. This method highlighted that Link is a critical feature in determining the accuracy of the different regression models as seen in Table 4 (E) which drastically reduced the testing R squared of all models, independent variable TCS1 and TCS2 seem not to contribute to the models accuracy and in fact increased the performance of the RFRegressor increasing the R2 for training and testing while reducing the RMSE and MAPE as seen in Table 4 (C).

****

**Table 4: Reverse feature selection R squared, RMSE and MAPE.** (A) All features - # Route, Link, Direction, STT, AccSTT, TCS1 and TCS2. (B) Without Direction. (C) Without TCS1 and TCS2. (D) Without Route. (E) Without Link. (F) Only including STT, AccSTT and Link. (G) Only including AccSTT and Link.

The benefit of removing certain features is to reduce model complexity and increase accuracy, and thus this may explain why the removal of TCS1 and TCS2 increased the accuracy of RFRegressor during reverse feature selection.

**3.5.2 Alternative machine learning model comparisons**

To ascertain whether other widely used models in machine learning would be suitable for the Trips dataset and compare them with the linear models previously used, Random forest, Naïve Bayes Classifier, K-nearest neighbors and Support vector machine were applied to the dataset. A summary of the model accuracies is provided in Table 5.

****

**Table 5. Alternative model accuracies for Random Forest, Naïve Bayes Classifier, KNN and support vector machine applied to the Trips dataset.**

As shown in Table 5 the accuracies for these respective models are low in comparison to the regression models previously utilised for the Trips dataset and thus are less suitable for the analysis of the Trips dataset. This is unsurprising given most of these models are used for classification. The current study highlights that regression models are more suited to the analysis and prediction of the Trips dataset, and generation of an accurate machine learning model for the chosen dependent variable, AccSTT. The models given have been used historically for travel time predictions as cited by Qiu and Fan in 2021 where they performed a critical analysis of machine learning methods for travel time predictions however the implementation, data preparation and data granularity is beyond the scope of this study (Qiu and Fan, 2021).

**4.Discussion**

***Significance of study:*** The current study aimed to explore and analyse the Trips dataset provided by the DCC which provides travel time data for 50 different routes in Dublin City. After exploring and analysing the dataset with the appropriate EDA and statistics, the data was processed for machine learning models with the aim of predicting the dependent variable, AccSTT. The benefit of this model to the DCC would be to predict travel times at particular Routes so to avoid increasing travel times in areas where travel times may be excessively high and to prevent congestion, leading to disruption of the flow of people in and out of the city. The implication of the effective management of travel times in cities such as Dublin is of great importance as excessive travel times can cause city wide issues, and in particular render public services such as buses with unsustainable travel times. A study in 2020 by Serin *et al.* harnessed a bus arrival time dataset in order to predict travel times for city buses (Serin, Alisan and Erturkler, 2022). In this study they used a 3-layer approach to achieve the highest model accuracy and compared the RMSE and MAPE for particular segments of bus routes. Another study by Zhang *et al.* 2022 compared a number of ML based approaches to travel time prediction and notes that the uncertainty is high in short term travel time predictions, finally concluding that the Kalman filter method was best suited to travel time prediction (Zhang *et al.*, 2022). Qiu and Fan 2020 performed a critical evaluation of machine learning approaches to Travel time prediction. In this extensive study they concluded that of all the machine learning models used for travel time prediction Random forest appeared to the strongest in terms of accuracy and generate the models with highest precision (Qiu and Fan, 2021). Interestingly, in the current study, Random forest regressor was deemed to have the highest accuracy and lowest RMSE and MAPE. They also highlight that increase the number of trees for Random forest beyond 10 up to 500 leads to a lower MAPE result increasing the accuracy of the model (Qiu and Fan, 2021). The number of Trees used in this study was 100 which falls within this range. Overall the significance of this study to the DCC would be to predict travel times and manage certain routes with above average travel times accordingly to benefit the functioning of Dublin City.

**5.Conclusion**

In conclusion the outcome of this study has shown that regression models are best suited to the development of accurate machine learning models for the AccSTT variable in the Trips dataset provided by the DCC. A number of regression based approaches were used to achieve the highest accuracy in prediction of AccSTT, with RFRegressor and DTRregressor displaying the highest R2 values for training and testing data, along with the lowest RMSE and MAPE. Overall the RFRegressor displayed the highest accuracy, with this model being a good candidate for use on the Trips dataset. The accuracy of this model would likely be increased if more data was provided on speed, weather, presence of bus lanes or pedestrian zones thus making the granularity of the data imperative to increase the strength of the model in the prediction of the AccSTT variable. More work is clearly needed to develop stronger prediction models for travel times based on this dataset, and perhaps more complex machine learning approaches such as neural networks along with a higher data granularity, and larger datasets, would aid in generating models with increased accuracy.

**6.Bibliography**

Balamurugan, N. M. *et al.* (2022) ‘A Novel Method for Improved Network Traffic Prediction Using Enhanced Deep Reinforcement Learning Algorithm’, *Sensors*, 22(13), pp. 1–17. doi: 10.3390/s22135006.

Downey, A. (2014) *Think Stats: Exploratory Data Analysis*. O’Reilly Media. Available at: https://books.google.ie/books?id=b7HXBAAAQBAJ.

Jiber, M. *et al.* (2020) ‘Road traffic prediction model using extreme learning machine: The case study of tangier, morocco’, *Information (Switzerland)*, 11(12), pp. 1–15. doi: 10.3390/info11120542.

Liu, Y. and Wu, H. (2017) ‘Prediction of Road Traffic Congestion Based on Random Forest’. doi: 10.1109/ISCID.2017.216.

Müller, A. C. and Guido, S. (2016) *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O’Reilly Media, Incorporated. Available at: https://books.google.ie/books?id=qjUVogEACAAJ.

Qiu, B. and Fan, W. (2021) ‘Machine learning based short-term travel time prediction: Numerical results and comparative analyses’, *Sustainability (Switzerland)*, 13(13). doi: 10.3390/su13137454.

Schröer, C., Kruse, F. and Gómez, J. M. (2021) ‘A systematic literature review on applying CRISP-DM process model’, *Procedia Computer Science*, 181, pp. 526–534. doi: 10.1016/J.PROCS.2021.01.199.

Serin, F., Alisan, Y. and Erturkler, M. (2022) ‘Predicting bus travel time using machine learning methods with three-layer architecture’, *Measurement: Journal of the International Measurement Confederation*, 198(May), p. 111403. doi: 10.1016/j.measurement.2022.111403.

Zhang, L. *et al.* (2013) ‘An Improved K-nearest Neighbor Model for Short-term Traffic Flow Prediction’, *Procedia - Social and Behavioral Sciences*, 96, pp. 653–662. doi: 10.1016/J.SBSPRO.2013.08.076.

Zhang, X. *et al.* (2022) ‘Travel time prediction of urban public transportation based on detection of single routes’, *PLoS ONE*, 17(1 January 2022), pp. 1–23. doi: 10.1371/journal.pone.0262535.