Faculty of Natural and Mathematical Sciences
Department of Informatics

Bush House, King's College London
Strand Campus,30 Aldwych
London WC2B 4BG
Telephone 20 7848 2145
Fax 020 7848 2851

7CCSMDPJ

Individual Project Submission 2018/19

Name:	Alex Wilkes
Student Number:	1833720
Degree Programme:	MSc Data Science
Project Title:	Predicting Tips in the New York City Taxi Market
Supervisor:	Dr Alfie Abdul-Rahman
Word Count:	15,000

RELEASE OF PROJECT

Following the submission of your project, the Department would like to make it publicly available via the library electronic resources. You will retain copyright of the project.

☑ I **agree** to the release of my project

☐ I **do not** agree to the release of my project

Signature: **Date:** August 28, 2019

Department of Informatics
King's College London
United Kingdom

7CCSMPRJ Individual Project

Predicting Tips in the New York City Taxi Market

Name: **Alex Wilkes**
Student Number: 1833720
Course: MSc Data Science

Supervisor: Dr Alfie Abdul-Rahman

This dissertation is submitted for the degree of MSc in MSc Data Science.

Acknowledgements

I must acknowledge and offer a sincere thank you to my supervisor, Dr Alfie Abdul-Rahman. Dr Abdul-Rahman has been extremely generous with her time, providing thoughtful feedback and encouragement from the beginning of the project.

Abstract

Can we predict whether a taxi driver in New York will be tipped for their journey and how generous this tip will be? How is tipping influenced by geography, time, weather and other features of the trip? We examine these questions using the data from the 2018 New York City Taxi and Limousine Commission Dataset and corresponding weather data. We apply a portfolio of regression models and find that tip percentages can be predicted with an r^2 score of 12.5% using a random forests. We also apply a portfolio of classification model to predict when a customer will decline to leave a tip at all. The best classifier, XGBoost, performs with a recall of 56% and a precision of 24%. Inspection of the models reveal the most important features in determining tipping behaviours are spatial features, whilst temporal features and other features have less impact. The weather features, i.e temperature and precipitation, are found not to be important or significant in tipping behaviours.

Nomenclature

CRS	Coordinate Reference System
CSV	Comma Separated Values
ESRI	Environmental Systems Research Institute
GIS	Geographic Information System
GMM	Gaussian Mixture Model
GPS	Global Positioning System
GPU	Graphics Processing Unit
JFK	John Fitzgerald Kennedy Airport
NAD83	The North American Datum of 1983
r^2	Percentage of Variation in Target Variable Explained by Variance in Features
RMSE	Root Mean Square Error
SQL	Structured Query Language
TLC	Taxi and Limousine Commission
w	Weight Vectors
X	Feature Vector
y	Target Variable

Contents

1 Introduction **1**
- 1.1 Research Questions . . . 2
- 1.2 Report Structure . . . 2

2 Background **3**
- 2.1 Supervised Learning . . . 3
  - 2.1.1 Naïve Bayes . . . 3
  - 2.1.2 Linear Regression and Linear Classification . . . 4
  - 2.1.3 Artificial Neural Networks . . . 4
  - 2.1.4 Decision Trees and Related Ensemble Methods . . . 5
  - 2.1.5 Unsupervised Learning . . . 7

3 Related Work **8**
- 3.1 Visualisation Literature . . . 8
  - 3.1.1 General Visualisation Approaches . . . 8
  - 3.1.2 Product Demonstrations . . . 9
- 3.2 Big Data Literature . . . 10
- 3.3 Machine Learning Literature . . . 11
  - 3.3.1 Regression Models . . . 11
  - 3.3.2 Clustering Models . . . 12
  - 3.3.3 Relationship With Weather . . . 13

4 Data Acquisition **14**
- 4.1 Tabular Trip Data . . . 14
- 4.2 Spatial Data . . . 14
- 4.3 Weather Data . . . 14

5 Data Cleaning **15**
- 5.1 Tabular Trip Data . . . 15
  - 5.1.1 Number of Passengers . . . 15
  - 5.1.2 Fares . . . 19
  - 5.1.3 Negative Fares . . . 20
  - 5.1.4 Very High Fares . . . 21
  - 5.1.5 Duration . . . 21
  - 5.1.6 Distance . . . 22
  - 5.1.7 Tabular Trip Data Cleaning Summary . . . 25
- 5.2 Spatial Data . . . 26
- 5.3 Weather Data . . . 27

6 Overall Approach **29**
6.1 Feature Engineering . . . 29
6.2 Environment . . . 29
6.3 Training and Testing Split . . . 29

7 Initial Exploration **31**
7.1 Approach . . . 31
7.2 Results and Discussion . . . 31
7.2.1 Tipping Percentage . . . 31
7.2.2 Cab Colour . . . 32
7.2.3 Number of Passengers . . . 32
7.2.4 Temporal Features . . . 34
7.2.5 Spatial Factors . . . 35
7.2.6 Weather . . . 37

8 Predicting the Tip Percentage - Regression Approach **38**
8.1 Approach . . . 38
8.1.1 Linear Regression . . . 38
8.1.2 Random Forests Regression . . . 38
8.1.3 Neural Network Regression . . . 38
8.2 Results and Discussion . . . 39
8.2.1 Linear Regression . . . 39
8.2.2 Neural Network Regression . . . 39
8.2.3 Random Forests Regression . . . 41

9 Predicting Zero Tips - Classification Approach **44**
9.1 Approach . . . 44
9.1.1 Naïve Bayes . . . 44
9.1.2 Tree-Based Classifiers . . . 44
9.1.3 Neural Network Classifier . . . 44
9.2 Results and Discussion . . . 45
9.2.1 Insights from the Decision Tree . . . 47
9.2.2 Insights from the Logistic Regression . . . 48
9.2.3 Insights from Random Forests . . . 50

10 Conclusion **54**
10.1 Future Work . . . 55
10.2 Lessons Learned . . . 56

References **57**

List of Figures

1 K-means algorithm as documented by Murphy [1] . . . 7
2 Nelson's visualisation approach . . . 10
3 Plots showing distribution of all features (part 1) . . . 16
4 Plots showing distribution of all features (part 2) . . . 17
5 Data cleaning checklist . . . 18
6 Distribution of passenger count across all trips . . . 19
7 Kernel density estimate plot of fare distribution . . . 19
8 Journeys with negative fares . . . 20
9 Kernel density estimate plot of duration feature . . . 21
10 Kernel density estimate plot of distance feature . . . 22
11 Distribution of journeys beginning or ending outside of NYC . . . 23
12 Distribution of fare code . . . 23
13 Pair plots of distance, duration and total fare . . . 24
14 Clustering model of cost vs. duration . . . 24
15 Linear regression model of cost vs. distance . . . 25
16 Summary of data cleaning filtering operations . . . 26
17 Weather features . . . 28
18 Tip percentage distribution plotted with mixture model means . . . 32
19 Average tips and tip percentage distributions for green vs. yellow Cabs. . 33
20 Average fares segmented by passenger count. . . . 34
21 Number of journeys and average fare by day-of-week and hour-of-day . . . 35
22 Count of trips and mean tip percentage by pickup and dropoff Zones . . . 36
23 Relationship between average tip percentage and temperature, precipitation 37
24 Distribution of predictions vs. actual tip percentages for linear regression 41
25 Distribution of predictions vs. actual tip percentages for neural network . 41
26 Distribution of predictions vs. actual tip percentages for Random Forests 42
27 Feature importances in trained Random Forest . . . 43
28 Receiver operating characteristic curves . . . 47
29 Decision Tree . . . 51
30 Decision Tree feature importances . . . 52
31 Random Forests classifier feature importances . . . 53

List of Tables

1 New features . . . 30
2 Existing features . . . 31
3 Weights for statistically significant features in linear regression . . . 40
4 Classifier performance metrics . . . 45
5 Weights for logistic regression . . . 48

1 Introduction

This project aims to model and predict tipping behaviours for taxis in the New York City area using publicly available data from the city's Taxi and Limousine commission. The main dataset records tip amounts and a number of features for all trips taken in taxis in New York since 2009.

We apply a number of supervised learning techniques, both regression and classification, to model tipping behaviours. A classification approach predicts whether a tip is left at all, whilst a regression approach attempts to predict the value of it.

The domain of this project is that of smart cities, urban transport and public policy. The results of an accurate prediction model would be of interest to taxi drivers as it would allow them to optimise their decisions for maximal income. The salary of a taxi driver is estimated at $30,000 [2] and our data shows that tips on this income vary significantly. The lowest common tip value is 0% and the highest common tip is 23%. Being able to increase the frequency of the latter over the former would increase annual income for a cab driver by a figure in the order of thousands of dollars.

A model of tipping would be of equal interest to public policy makers who are required to understand the market. For example, tax authorities who receive tax returns must judge whether income is being honestly reported. Those enforcing anti-money laundering regulations must judge whether it is realistic that wealth has been obtained through legal earnings. Any such public policy questions require us to have a model that can predict the tips a driver will receive and a confidence interval around that model.

This project surveys a number of supervised learning techniques for predicting tipping behaviours, both the existence of a tip and the value of a tip. It provides the basis of a model for an application which could direct taxi drivers to improve the tips they receive. The kind of recommendations we could expect to give a driver are either spatial ("Go to JFK airport"), temporal ("Avoid saturdays"), weather related ("It is raining so tips will be higher") or even something else entirely ("Upgrade from a green license to a yellow medallion"). The model could also be used by an investigator from government. For example in the tax context, if a driver reports that they only received a 10% tip on average in their income declarations, the model can predict the average tip for the journeys the driver has claimed to have operated. This could form evidence for or against such a claim.

The project encounters a number of challenges typically associated with "big data." These can be analysed in terms of the five "Vs" which typically quoted to characterise big data problems [3]. The volume of the dataset is extremely large, with 19 features for over 1 billion journeys in the city over a 10.5 year period. The veracity of the data is challenged, something demonstrated in the data cleaning section of this report. The value of the dataset, both to taxi drivers and policymakers, is significant as discussed above. The variety of datasets is broad, including not just the trip data, but spatial reference data also supplied by the Taxi and Limousine Commission and weather data that is sourced separately.

1.1 Research Questions

The following research questions have been set out as the focus of this project. The first three were selected on the basis that they are of interest to both the taxi driver looking to improve their income through higher tips and to the policymaker looking to understand the market.

- **Can we accurately predict what tip will be received on a given journey?** Given the a journey, if we know the time, origin, destination and other features such as the number of passengers and weather, can we predict the tip percentage of that journey?
- **Can we accurately predict when no tip will be left?** Given the journey, if we know these same features, can we accurately predict when a customer won't tip at all?
- **What are the biggest influencers of tipping behaviours in New York City?** Of the different features we have in our dataset; spatial, temporal and other, which are the most importance in determining tipping behaviours?
- **Does the weather impact tipping behaviours in New York City?** A number of drivers have looked at the relationship between the taxi market and weather [4, 5]. Devaraj and Patel [6] found there is a relationship between weather and tipping. Is there evidence of this in our dataset and if so, what is the nature of the relationship?

1.2 Report Structure

Background literature and related work is discussed in Sections 2 and 3 respectively. The sourcing of the data is then described in Section 4, followed by a description of the extensive data cleaning undertaken in Section 5. The overall approach to modelling, including the technology choices made are describe in Section 6. Prior to development of the key models, the data is explored to understand correlations between the key groups of features and the tipping percentage. The results of this exploration are set out in Section 7. Section 8 sets out the regression approach to tip prediction including the results and discussion, whilst Section 9 sets out the classification approach including the results and discussion. Final discussion and conclusions on the research questions set out above are then drawn in Section 10.

2 Background

2.1 Supervised Learning

Russell and Norvig [7] define supervised learning as the task of taking a training set of example input-output pairs (x_1, y_1), (x_2, y_2), $\ldots$, (x_N, y_N) where each y_j was generated by an unknown function $y = f(x)$ to discover a function h that approximations the true function f. Classification is defined as the subset of supervised learning in which the outputs y are always "one of a finite set of values" [7]. This is in contrast to regression in which the output "y is a number." [7].

Classification can be broken down into a number of distinct individual techniques. A number of them are applied in the techniques discussed below. Classification is used in this project to predict where a driver will not receive a tip for a journey.

Breiman [8] discusses the purpose of classification, noting "an important criterion for a good classification procedure is that it not only produce accurate classifiers (within the limits of the data) but that it also provide insight and understanding in to the predictive structure of the data." Applied in the context of this project, we understand the purpose is not just to understand whether tips will be low or high, but to understand how variance in the features might drive certain outcomes. More concretely, if our model predicts high tips in Manhattan then it would be useful to understand that it is the spatial feature of being in Manhattan driving this prediction.

As well as classification models, regression models are also built, predicting not a 'low' or 'high' category for the tip, but the value for the tip percentage.

2.1.1 Naïve Bayes

Naïve Bayes [9] is a simple classification technique that assumes that each of the features are independent, or that the probability of a class "will factorise in to the product of its univariate marginals." The algorithmic approach calculates the probability of a new example belonging to each possible class using the Bayesian probability identity, the frequencies with which each class appears and the frequencies with which each value of each feature is seen, given the class. Classification is achieved by selecting the class that maximises this probability.

Naïve Bayes makes this strong assumption, even though it is often known that the assumption does not hold. Hand and Yu [9] note that "despite this, the model often seems to perform surprisingly well." They suggest that this is because its simplicity means "low variance in its probability estimates" and that even though the probability estimates may be biased, "the rank order is preserved." This is to say it mostly still selects the same class as its prediction, despite not knowing the full joint distribution of the probabilities.

Naïve Bayes is applied as a simple classifier which requires little tuning and also as a baseline against which to measure more sophisticated algorithms.

2.1.2 Linear Regression and Linear Classification

Linear regression is a supervised learning technique, more informally know as the process of "fitting a straight line" [7]. It attempts to find a set of weights $\mathbf{w}$ that best fits a function

$$h(\mathbf{w}) = \mathbf{w} \cdot \mathbf{x} \tag{2.1}$$

as an approximation for f.

In its simplest form, Linear Regression predicts using a single feature (univariate linear regression) but can be extended to multiple features (multivariate linear regression). In this project, multivariate linear regression is used both to predict tips, and also to help clean the data before applying the models.

Related to linear regression is the perceptron. This uses the linear form associated with linear regression but passes the output through a threshold function that generates a class prediction rather than a numeric prediction. In the case of a perceptron this is a simple step function which returns 1 if the output is above a certain value, and 0 otherwise. More sophisticated threshold functions are used such as the logistic function

$$g(z) = \frac{1}{1 + e^{-z}} \tag{2.2}$$

which also outputs values between 0 and 1. When this is used the technique is referred to as logistic regression.

Ridge regression [10] refers to a version of this in which the cost function used to fit the weights is augmented with a penalty equal to some proportion of the sum of the squares of the weights.

$$\min \sum (\mathbf{w} \cdot \mathbf{x}) - y)^2 + \lambda ||\mathbf{w}||^2 \tag{2.3}$$

This is useful where there are a large number of highly collinear features, as is the case with this dataset because when used the weights tend towards zero. Logistic regression models in this project apply this penalty, reflecting the known collinearity of the features.

The disadvantage of both the linear regression and logistic regression approach for this dataset is it assumes the underlying relationships are linear in nature. It cannot learn decision boundaries that are not linearly-separable. This will not fit well with the spatial data where non-linear boundaries are likely to exist. The benefit of this approach is that the trained model is relatively explainable. The sign and magnitude of the weights corresponding to each feature can be inspected in order to interpret the model.

2.1.3 Artificial Neural Networks

Artificial neural networks extend the linear regression and the perceptron by connecting multiple units together in layers where each unit takes the weighted linear product, typically passed through an activation function, of other units in the previous layer. For a classification neural network the activation function, like the threshold function, maps the weighted linear product to class outcomes, such as 0 and 1, or -1 and 1. For a

regression neural network, a range of values is predicted.

The advantage of using artificial neural networks is they can approximate very complex functions and are able to produce non-linearly separable decision boundaries which is something we expect to be relevant to our spatial and temporal features. The disadvantage is the trained network is relatively opaque. That is to say they are difficult to interpret and explain, even though they are able to achieve high levels of accuracy. They also risk overfitting to the data. A sufficiently large neural network will classify training data perfectly by encoding a lookup table, but a lookup table wouldn't necessarily perform well on new data. Techniques are available to limit this, including the use of unseen test data, which is used to detect when this has happened, and the use of validation data during training, which is used to avoid it.

2.1.4 Decision Trees and Related Ensemble Methods

Decision trees are discussed by Russell and Norvig [7] as a supervised learning approach. The authors describe decision trees as representing "a function that takes as input a vector of attribute values and returns a 'decision.'" and that it reaches the decision by "performing a sequence of tests." Decision trees at their most fundamental are indeed a tree: a graph which is connected, directed, has a root node, does not contain cycles and is such that each node has at most one parent. Each non-terminal node represents a decision point on one of the features in the model which directs to either another non-terminal node or to a terminal node. Terminal nodes are associated with a class prediction or regression value. Decision trees can be learned from examples and a number of algorithms exist for doing this. Most most well known are ID3 [11], CART [8] and C4.5 [12]. The general approach is to recursively split the training data using a "test based on a single outcome" and grow the tree until a terminating condition is met. ID3, CART and C4.5 differ on the test they use, as noted by Wu [13]. CART uses the gini-impurity index whilst ID3 and C4.5 uses information based criteria.

CART implements a pruning process which sees subtrees of the graph replaced with terminal nodes, in an attempt to reduce the complexity of the resulting tree where it doesn't provide sufficient additional predictive power.

Quinlan [11] set out the widely ID3 algorithm for growing decision trees in a 1986 paper. Breiman [8] states that work on decision trees began in the 1960s when the AID (automatic interaction detection) program at the University of Michigan applied regression trees in social science research. In 1973 Breiman and Friedman independently applied decision trees to solve classification problems and published the seminal 'Classification and Regression Trees.' [8]

Decision trees are applied in this project as both a regression and classification tool. This is because they balance high predictive power with explainability. The logic for a decision can be inspected and understood by humans, as noted by Russell and Norvig [7]: "one important property of decision trees is that it is possible for a human to understand the reason for the output of the learning algorithm." They are also one of the classifiers that can provide non-linear decision boundaries. This is relevant to the dataset, particularly spatial features. For example, we may well expect that tipping behaviour differs

at pickups and dropoffs around airports from the immediately surrounding regions they are in.

Decision trees have often been combined in to ensemble methods. This refers to the combining of multiple models to improve predictive power. Breiman writes specifically about the most popular tree-based ensemble method: Random Forests.

> "A random forest is a classifier consisting of a collection of tree-structured classifiers $\{ h(\mathbf{x}, \Theta_k), k = 1, ...\}$ where the Θ_k are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input $\mathbf{x}$." [14]

Random Forests implements bootstrap aggregating, also known as 'bagging'. This process involves taking random samples of the data, with replacement, and building separate decision trees on those samples (boosting). As well as taking random samples of the data, each tree is also built on a small subset of the features: a technique referred to by Ho [15] as the "random subspace method." The resulting decisions of the individual trees are then aggregated using a voting mechanism.

Random Forests is selected for use in this project because, as noted by Scornet, Biau and Vert [16], they are "simple to use," and the method is "generally recognized for its accuracy and its ability to deal with small sample sizes, high-dimensional feature spaces and complex data structures." They are also less prone to overfitting [17]. Breiman [14] provides a mathematical proof of why they are not prone to overfitting where we might expect they might become so.

The last supervised learning approach considered is XGBoost [18]. Chen and Guestrin extend the Random Forest approach using gradient boosting. Boosting is the technique of using errors on previously trained trees to improve the training of new trees. Gradient boosting refers specifically to the technique of adapting the loss function which is used to train the model using errors from previous iterations. Chen and Guestrin apply this, along with a technique to reduce overfitting (shrinkage and column subsampling), a technique to improve the identification of split points within features (weighted quantile sketch) and a technique for dealing with sparse data. The authors note that sparse data is is common where "one-hot encoding" is used. This refers to the expression of categorical features such as "ratecode" as a set of variables, one for each value the feature can take that are equal to 1 if the feature takes that value, 0 otherwise. This is highly relevant to the New York Taxi data which includes a number of features that have to be encoded this way. The authors of XGBoost also address the computational challenges that come with training on very large datasets, including techniques to enable the training process to be distributed across multiple machines in parallel, techniques to make use of caching and techniques for storing data on disk (compression and sharding) to make it more immediately available when required. In using their algorithm, this project is able to benefit from these techniques.

2.1.5 Unsupervised Learning

Murphy [1] refers to unsupervised learning as the attempt "to find 'interesting structure' within the data," sometimes called "knowledge discovery." It is defined in contrast to supervised learning, where "we are not told what the desired output is for each input."

A typical unsupervised learning technique is clustering, in which examples are assigned to clusters or subgroups. The k-means algorithm is one of the simplest, most popular clustering algorithms. The algorithm is set out by Murphy [1] as follows:

Algorithm 11.1: K-means algorithm

initialize $\mathbf{m}_k$;
repeat
  Assign each data point to its closest cluster center: $z_i = \arg\min_k ||\mathbf{x}_i - \boldsymbol{\mu}_k||_2^2$;
  Update each cluster center by computing the mean of all points assigned to it: $\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{i:z_i=k} \mathbf{x}_i$;
until *converged*;

Figure 1: K-means algorithm as documented by Murphy [1]

Gaussian Mixture Models (GMMs) are a generalisation of this clustering approach, defined by Russell and Norvig [7] as having a probability density function

$$P(\mathbf{x}) = \sum_{i=1}^{k} P(C = i)P(\mathbf{x} \mid C = i) \tag{2.4}$$

where $\mathbf{x}$ is the random variable being modelled and C are the underlying component distributions. Russell and Norvig [7] describe the expectation-maximisation algorithm, similar to the k-means algorithm, which is used for recovering the parameters of the underlying component distribution parameters. K-means clustering is used in this project as a data cleaning approach to separate erroneous and legitimate data. GMMs are used to model the distribution of tip percentages.

3 Related Work

3.1 Visualisation Literature

The New York City taxi dataset is very popular with the visualisation community because of its richness in both features and the number of trips in the data. Most approaches are concerned with creating broad and flexible visualisations or tools that can answer a broad range of research questions. This project is primarily a machine learning approach but it is useful to understand how the visualisation community has been able to extract insights from the data.

3.1.1 General Visualisation Approaches

Savage and Vo [19] use the New York dataset to create a number of visualisations to identify trends within the taxi market. A number of these are similar to those identified by analysis within this project. Animation is used as a visual channel to convey movement trends identified within the dataset. They also use a first order markov assumption to calculate a 'transitional matrix' capturing the likelihood that a trip beginning in one neighbourhood will end in another. This first order markov assumption translates to "the likelihood a cab goes from Midtown to the Upper East Side does not depend on the fact that it previously picked up and dropped off in the Upper West Side."

The authors mention no data cleaning in their work which seems unusual as the dataset is widely acknowledged by other authors to be subject to a number of data quality issues: something validated later in this project.

Ferreria et al. [20] use the dataset to create a visual query tool called 'TaxiVis' that can be used by traffic engineers and economists for planning purposes. The query tool allows users to manipulate the visualisation directly to build queries, which are structured as traditional structured query lanaguge (SQL). Their explicit grouping of features in spatial, temporal and other is a framework adopted by this project. Their performance requirement is extremely challenging because they aim to enable real-time visualisation of the data. They use a k-d tree index to allow queries to complete in a performant amount of time.

Their approach is notably flexible in the range of queries and research questions it can support. The authors share three case studies showing how queries can investigate various hypotheses. For example, they show that the cost of journeys per mile is higher in Manhattan than other areas which they suggest explains driver reluctance to leave Manhattan. In their conclusions they note their product 'attains a good balance between simplicity and expressiveness.' It is helpful to note that with complex datasets, these two aims are often at odds and such a balance must be sought. With many more features than could possibly be visualised simultaneously, and a dataset so large that in fact almost all edge cases are represented within the dataset, it is difficult to separate trends from noise.

A downside of their approach is that despite being a tool designed to be adopted by other researchers, it requires specific hardware to run as a result of design decisions they

make. The authors note that they were limited by existing technology and would like to refactor the product to be more 'portable.'

Lu, Wang and Yuan [21] create 'trajectory ranking' diagrams based on a similar dataset from Beijing. The diagrams rank the different routes that pass through a road segment selected by a user. The authors highlight the value of taxi datasets, noting that they can be considered 'sensors of the city traffic situation' and 'a reasonable sample of the full traffic flow.' This understanding significantly broadens the set of those potentially interested in the insights produced to all traffic modellers and urban planners.

The disadvantage of their approach is that their visualisations are limited to a very specific analysis: the ranking of different journeys passing through a single route by some attribute. They have put significant work into identifying complex methods to visualise these rankings, but the overall approach is not relevant for a wide set of research questions, including our question of whether we can predict tips.

3.1.2 Product Demonstrations

A number of companies have used the dataset to demonstrate their visualisation product's capabilities. Omnisci [22], a products company, create an interactive dashboard for visualising the full dataset. This dashboard uses the dataset to present various summary graphics and map visualisations, e.g. average tip per building. Their query process sees the full dataset processed using eight high-powered graphical processing units (GPUs) and sent to the user's browser as an image tile. Their visual analytical approach is able to identify notable relationships between the taxi datasets and other data, for example the relationship between a hotel group's share price and taxi activity around their hotels. Nelson [23] loads the data into the ESRI GIS platform and then produces a very visual and highly consumable story focused on the spatial features of the data. This includes a specific investigation on tip percentages, where he finds that the highest occur in the busiest parts of Manhattan within 'distinct boundaries of generosity.' Nelson uses a specific visualisation to show how tipping percentages vary for a given location, both in terms of trips that pickup there and trips that dropoff there. This solves a specific issue which is that generally, a map-based visualisation could only show one of these two features, and in some locations there were quite distinct stories being told by the data by these two different features. The approach is shown in Figure 2. Miles [24] uses "KeyLines," a graph visualisation approach to visualise trips on a relatively small subset of the data.

They use graph theory concepts, e.g. eigencentrality of notes on the network. They encode these concepts within their visual channels in order to help the user understand the dataset. Using this, the user can quickly identify the most popular pick up and drop off locations. They can overlay the graph on top of the map, and quickly see frequent journeys e.g. journeys between lower Manhattan and JFK. They can also use aggregating methods, "combos", to simplify the visualisation, breaking it into clusters to reduce visual clutter.

Their use of "donuts" to summarise a feature within the group (e.g. pickups vs. dropoffs) is highly effective, allowing additional information to be encoded in to map

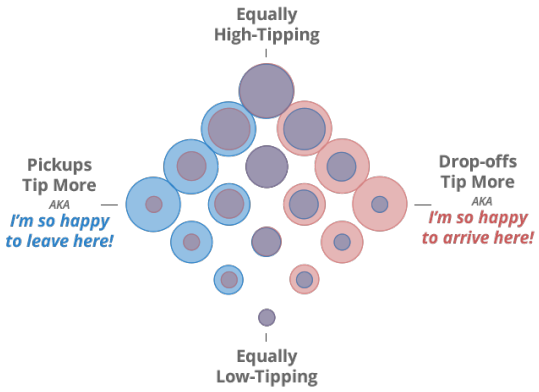

Figure 2: Nelson's visualisation approach

based visualisations. The downside of their approach is because of the rich visualisations they create, they only load a very small subset of data. Just three hours from one day of trips. As a purely visual approach, it's difficult to interrogate the visual outputs or inspect the underlying and peripheral data to understand why certain phenomena are occurring.

3.2 Big Data Literature

A number of authors in the literature have used the dataset as an opportunity to explore the computational challenges associated with processing very large datasets as discussed in the introduction.

Deri, Franchetti and Moura [25] apply Dijkstra's algorithm to the dataset in order to identify routes between the pickup and dropoff locations, which are not provided within the raw data. Their paper 'focuses on the design considerations for implementing Dijkstra's algorithm in a high-throughput environment that requires a low memory footprint'. Using their approach, they are able to process four years of historical data in one day, where it would previously have been estimated to take 3,000 days to process.

Most authors have dealt with the volume of the data by sampling it which is the approach taken by this project. For some this means taking a random sample of the whole dataset, as this project goes on to do. For others, they look only at very narrow timeslices. For some applications, being able to process the complete dataset is important and so there is real value is overcoming these computational challenges. For example, when analysing the dataset, duplicate trip records appear where a refund has been applied. To reliably find these duplicate pairings requires the full dataset and not just a sample to be loaded.

The focus of their approach is very much on solving the computational challenges associated with big data. Although the authors have succeeded in loading and analysing the full dataset, the research questions they address do not identify a research question that requires this. In this project, the computational heavy task of applying Dijkstra's

algorithm is not required, and computational complexity is managed by limiting sample size.

3.3 Machine Learning Literature

3.3.1 Regression Models

Finer [26] uses the taxi data to investigate the hypothesis that there are increasing numbers of meetings between the US Federal Reserve and major US commercial banks during information blackouts: periods in which information is not supposed to be shared with banks. Using regression techniques, the author finds "evidence suggestive of an increase in meetings between them both at the New York Fed's offices and in areas associated with dining and shopping. The occurrence of the changes very late at night and during typical lunch hours suggests informal or discreet communication."

The author is using the extremely large dataset on the city's taxis across a number of years to perform detective work on an extremely specific hypothesis relating only to a handful of locations in a small period of time. The author uses the dataset to make a remarkably specific accusation of foul play, but a number of assumptions are not addressed. Such missing assumptions include that the employees of any of the accused organisations would take taxis to or from their offices, that there aren't other legitimate reasons for meetings during these periods, and perhaps most fundamentally, if taxis from the federal reserve and a bank converge on the same location then this represents a meeting of the two. Finer's claim is not that any of the individual trips represent an improper meeting, but that at an aggregate level there are too many trips beginning or ending in pertinent locations so we should reject a hypothesis of no information leaking from the federal reserve.

Similar regression techniques to those implemented by the author are used in this project. Finer specifically bases their model on a Poisson distribution of events, whereas we apply linear and logistic regression. Finer selects the Poisson regression aproach because he is modelling an event-based phenomena (journeys being taken), whilst this project is attempting to predict tipping behaviours.

Koehrsen [27] develops two models, the first a linear regression, the second a random forest model to predict fares with the dataset. Koehrsen's model is highly relevant to our approach. Koehrsen addresses data quality issues with the dataset and proposes appropriate cleaning. They are able to accurately predict fares with a root mean square error (RMSE) of 4.86 on their linear regression and 3.38 on their random forests model against a validation data set. The key difference between Koehrsen's work and this project is that they are predicting fares, not tip percentages. Koehrsen demonstrates the success of the regression by visually comparing the distribution of predictions from the regression to the distribution of actual fares. This is an approach adopted by this project. Li [28] also develops regression models for predicting fares. They include linear regression, random forests, a neural network and LightGBM (a boosted tree model). Compared to Koehrsen, Li achieves a lower root-mean-square-error (RMSE), but undertakes significantly less data cleaning than Koehrsen which makes their approaches not

fully comparable.

Jain, See and Shandilya [29] use an earlier version of the NYC taxi data to build a linear regression that predicts tips by using time of day, location as well as taxi speed. They note that the highest tips are consistently associated with journeys in the densest parts of Manhattan. They bin the locations into 20 spatial bins to successfully improve the accuracy of their model. They find location to be the most significant predictor in their model, something this project refers to as the spatial features.

3.3.2 Clustering Models

Ge et al. [30] use data from taxis in San Francisco to develop a 'recommend system' for taxi drivers to use that will maximise their utilisation. A clustering algorithm is used to identify popular pick-up locations where drivers looking for passengers can expect to find them. The authors focus on a pair of algorithms they develop (LCP and SkyRoute) that will take a graph of these centroids and return a path within the graph. The path is a recommendation on the locations the driver should visit in order to maximise the probability of finding a passenger.

Ge et al's approach is not attempting to optimise tip percentages but to optimise utilisation. We note that actually maximising utilisation and maximising tipping may be competing objectives. Tips may be increased if a driver goes to a different part of the city, but if they spend an hour getting there and don't have any passengers en route then their utilisation will fall. A practical application of a recommendation system needs to balance these competing preferences of the user and these preferences may not just be to maximise tips or utilisation, but more subjective such as avoiding stressful drives or avoiding specific areas.

One learning we can take from their approach is their application of clustering to the spatial data. Doing this reflects the sense that cities are organised into neighbourhoods, and that those neighbourhoods have natural centroids in which a taxi pickup/dropoff would take place, which would be apparent to both drivers and passengers in the area. This maps naturally to the approach of this project, where such neighbourhood boundaries are supplied by the NYC Taxi and Limousine commission and from which centroids are calculated.

Peng et al. [31] use similar data to the New York City data but from Shanghai. Using the technique of 'non-negative matrix factorisation' they find that journeys can be characterised as combinations of three basis journeys: 'commuting between workplace and home,' 'business travel between workplaces' and 'journeys to and from other places.' A big difference between their data and the New York City data is that the Shanghai data includes a GPS trace of the route taken. This approach is an unsupervised learning approach in which the patterns emerge from the data. The authors are primarily trying to understand the historic travelling patterns within the city that are revealed by the data at an aggregate level, whereas this project is trying to predict some feature of future individual trips.

3.3.3 Relationship With Weather

The economics and psychology literature has has analysed data at the driver level, combining the data with weather data to explore competing theories of labour supply.

Camerer et al. [4] undertake a log-linear regression of driver income against hours worked, temperature, rainfall and a collection of other temporal features. The authors find evidence for their theory of 'income targeting which suggests that cab drivers stop working once they've hit a psychological limit for the day, and that on days it rains, this is more likely to be hit earlier which reduces the overall supply of cabs in the city.

Farber [5] furthers Camerer et al.'s [4] approach by developing his own regression model to estimate the economic elasticity of labour supply amongst cab drivers. Farber asserts that contrary to the previous conclusion, 'there is no relationship between earnings and rainfall' but that the supply of cabs is decreased during rainfall simply because driving conditions worsen during rainfall.

Like this project, both Camerer and Farber include weather features in their models to determine whether the weather influences behaviours. However, the behaviours they are interested in are analysing hours worked rather than tipping behaviours.

A different example comes from the cognitive neuroscience literature. Devaraj and Patel [6] use the NYC taxi dataset to consider the relationship between sunlight and tipping. Their model is a linear regression which includes fixed effects for individual drivers, the day of the year and the month of the year. They find a small but statistically significant positive relationship between the magnitude of sunlight and tipping percentages. This approach is highly related to the approach of this project, studying the same behaviours using similar features. However, this project uses different weather features (rainfall and temperature) and extends the approach using techniques which are better suited to the non-linear nature of the dataset.

4 Data Acquisition

4.1 Tabular Trip Data

Tabular trip data is supplied by NYC Taxi and Limousine commission (TLC) [32] in CSV files, three for each month. Each file contains a row of data for each journey, with each column representing a different feature of that journey.

The three files for each month correspond to the different types of cab operating in New York City: yellow cabs, green cabs and "for-hire" vehicles. The yellow cab data corresponds to trips taken in the traditional yellow New York Taxicab. The green cab data corresponds to trips taken in "Boro Taxis." According to the Trip Record User Guide [33] these were introduced in 2013 and are restricted to pickups "above W 110 St/E 96th St in Manhattan and in the boroughs." Finally, "for hire vehicle" data refers to companies such as Uber or Lyft. This last dataset has significantly fewer features than the first two datasets and so is not used in this project.

The tabular trip data is extremely voluminous, with 111 million rows of data in 2018. To manage this, it is randomly sampled with one twelfth of the data from each month. This approach ensures the dataset still contains any seasonal effects that may be present throughout the year whilst still being able to yield useful insights within the computing resources available.

A data acquisition script was created that automatically downloads the tabular trip data and spatial data from the NYC TLC website.

4.2 Spatial Data

Spatial data is also downloaded from the Taxi and Limousine commission (TLC) in a shape file [34]. The shape file contains geometries which set out the boundaries of the zones referenced by the tabular data. The geometries are sets of polygons, each of which is specified by points on its border. The points are referenced in a bespoke coordinate reference system (CRS); NAD83. This system is specific to New York and the surrounding areas.

4.3 Weather Data

Weather data is acquired from Raspisaniye Pogodi Ltd [35] who publish publicly available data sourced from the National Oceanic and Atmospheric Administration, part of the United States Federal Government. Data is specifically for the weather station at LaGuardia Airport. Implicit in the use of this data is an assumption that weather in this single location is a suitable proxy for weather across the New York area. This data is made available as a CSV file which is loaded manually.

5 Data Cleaning

5.1 Tabular Trip Data

After loading the tabular trip data, each of the features was analysed to identify spurious data. Figures 3 and 4 show a plot for each feature from which its overall distribution can be understood.

Data cleaning was undertaken using a checklist approach in which each feature was analysed for:

- The feature type: including whether it is categorical or continuous. We checked whether this made sense for the phenomena the feature was representing.
- The distribution of values for the feature. We checked whether features exhibited normal distributions and whether distributions contained multiple peaks. We also checked whether the shape of the distribution and number of peaks matched the likely underlying phenomena generating the feature.
- What the most extreme values within the feature's distribution were and were those extremes credible?
- Are there known policies which constrain the data we would expect to see? And does the data conform to them?
- Is there any missing data?

When looking for extreme values, we found it particularly difficult to draw a line between those which are unusual and those which are incorrect. Hawkins [36] defined extreme data as that which "deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism." Osborne [37] suggests in the case of normally distributed data, extreme values should be considered those falling in the range of 3x the standard deviation from the mean as a starting point for classifying data as an outlier. The risk of admitting outliers according to Osborne [37] is that of increasing the Type II error rate; the null hypothesis is false but isn't rejected by our test. Typically this means we do not find there is a significant relationship when we should do.

The results of this checklist are shown in Figure 5. Where this checklist identified areas for further investigation, those investigations have been detailed below.

5.1.1 Number of Passengers

Figure 6 shows the distribution of values in the number of passengers feature. We can see these are all integer values, as we would expect, but not all the values are credible. Official NYC documentation suggests the maximum number of passengers in a cab is 6[38]. It is also assumed that 0 is not a valid passenger count. The dataset's data dictionary does tell us this value is manually entered by drivers. It is therefore assumed these extreme values are simply erroneous entries from drivers. Trips are dropped from the dataset where the passenger count is either zero, or greater than six.

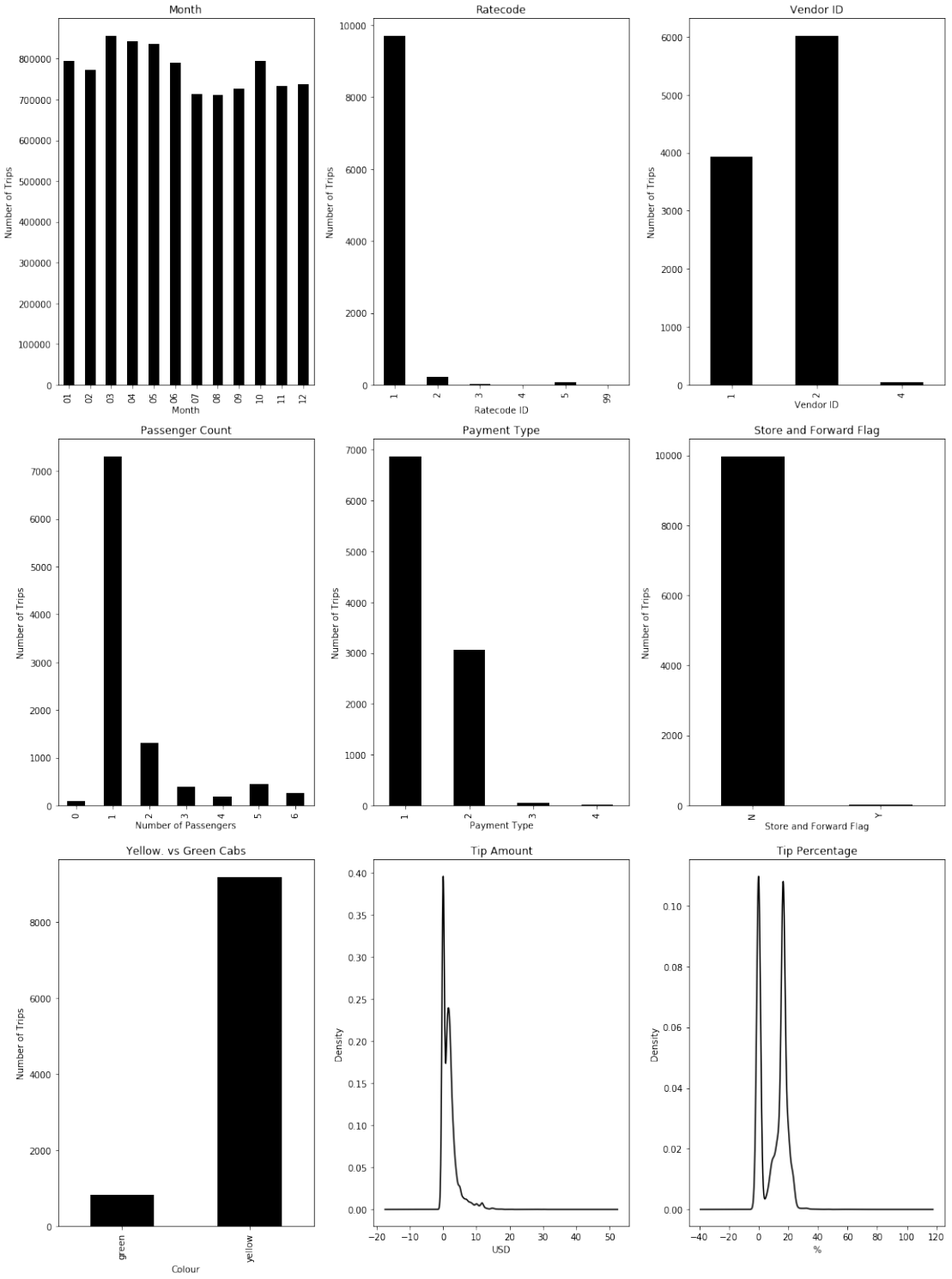

Figure 3: Plots showing distribution of all features (part 1)

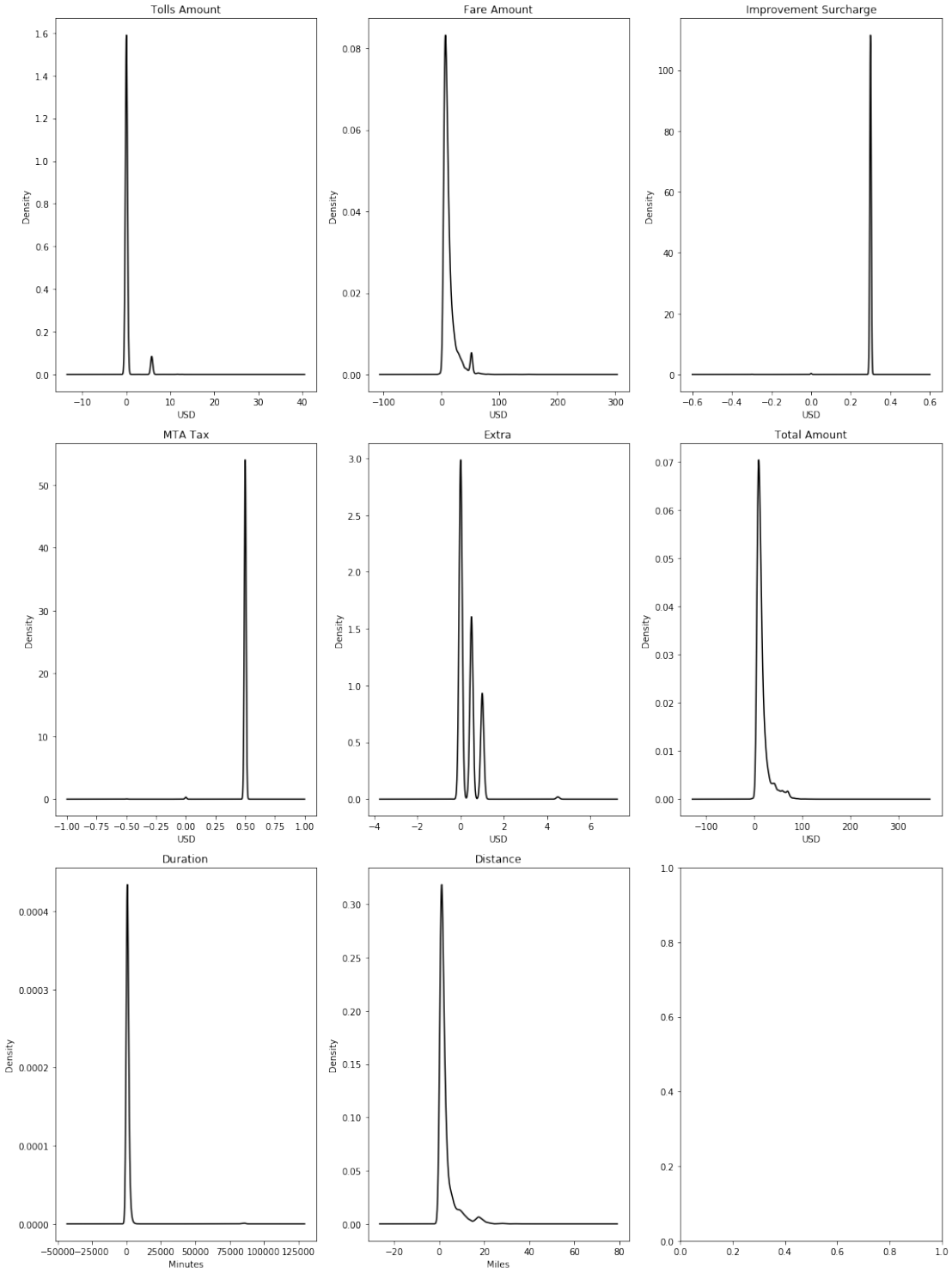

Figure 4: Plots showing distribution of all features (part 2)

Cleansing Table.pdf

Feature	Feature Type	Overall Distribution	Extreme values	Known policy	Missing Data
Ratecode	Categorical - Nominal	Dominated by ratecode 1	Includes examples with ratecode 99 which is not interpreted by the documentation	Rate code should be an integer between 1 and 5	
Vendor ID	Categorical - Nominal	A mixture of vendor ID 1, 2 and a very small number of vendor ID 4	All values are integer within expected range	Yes	
Passenger Count	Categorical - Ordinal	Dominated by 1 passenger trips with a second peak at 5 passengers	Includes infeasibly high passenger counts, up to 192.	Passenger count should be between 1 and 6 inclusive	
Payment Type	Categorical - Nominal	Dominated by payment type 1 and 2	All values are integer within expected range	Yes	
Store and Forward Flag	Categorical - Nominal	Dominated by 'N'	All values are boolean within expected range	Yes	
Colour	Categorical - Nominal	Dominated by yellow trips	All values are either 'yellow' or 'green' as expected	Yes	
Tip Amount	Continuous	Appears to be a mixture of multiple normal distributions with a floor effect at 0	Contains some negative values and some very high values	Tips should be strictly positive	Tips are only available for credit card transactions
Tip Percentage	Continuous	Two large peaks at 0 and 20%	Contains some negative values and some very high values	Tip percentages should be strictly positive	Tips are only available for credit card transactions
Tolls Amount	Continuous	Dominated by peaks at 0 and $5.76	All values are real valued within expected range		
Fare Amount	Continuous	Bimodal distribution featuring a single peak around $15 and a second peak around $52	Contains some negative values and some very high values		
Improvement Surcharge	Continuous	Single peaked at $0.30	Real valued within expected range		
MTA Tax	Continuous	Dominated by a single peak at $0.50	Real valued within expected range		
Extra	Continuous	Three peaks close together all at less than $2	Real valued within expected range		
Total Amount	Continuous	Appears to be a mixture of multiple normal distributions with a floor effect at 0	Contains some negative values and some very high values	Fares should be strictly positive	
Duration	Continuous	Appears to be a normal distribution with a floor at 0	Contains some negative values and some very high values	Duration should be strictly positive	
Distance	Continuous	Appears to be a mixture of multiple normal distributions with a floor effect at 0	Contains some negative values and some very high values	Distance should be strictly positive	
Pickup Zone	Categorical	The highest concentration of pickups is in Manhattan, neighbouring areas and airports	Contains some values which are placeholders for non-NYC locations		
Dropoff Zone	Categorical	The highest concentration of dropoffs is in Manhattan, neighbouring areas and airports	Contains some values which are placeholders for non-NYC locations		
Pickup Datetime	Continuous - Datetime	Broadly uniform with small peaks for certain weekdays and times of the day	Contains observations outside of the period (2018)	All trips should be in 2018	
Dropoff Datetime	Continuous - Datetime	Broadly uniform with small peaks for certain weekdays and times of the day	Contains observations outside of the period (2018)	All trips should be in 2018	

Figure 5: Data cleaning checklist

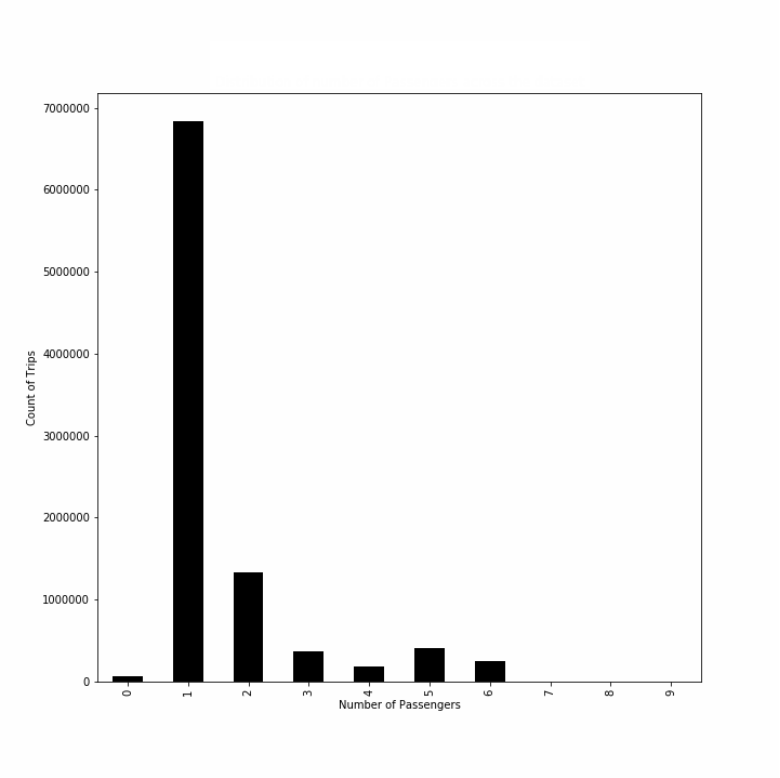

Figure 6: Distribution of passenger count across all trips

5.1.2 Fares

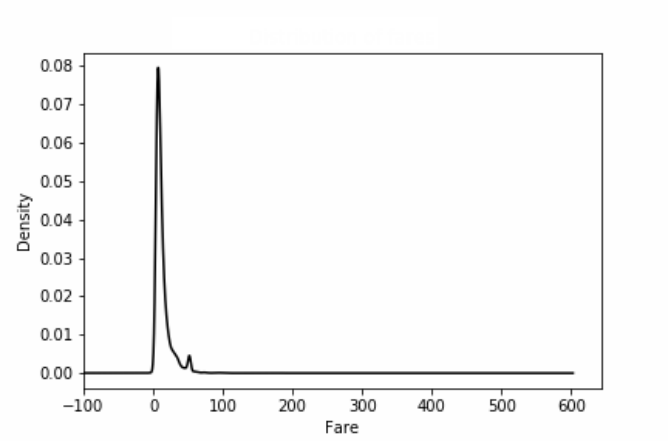

Figure 7: Kernel density estimate plot of fare distribution

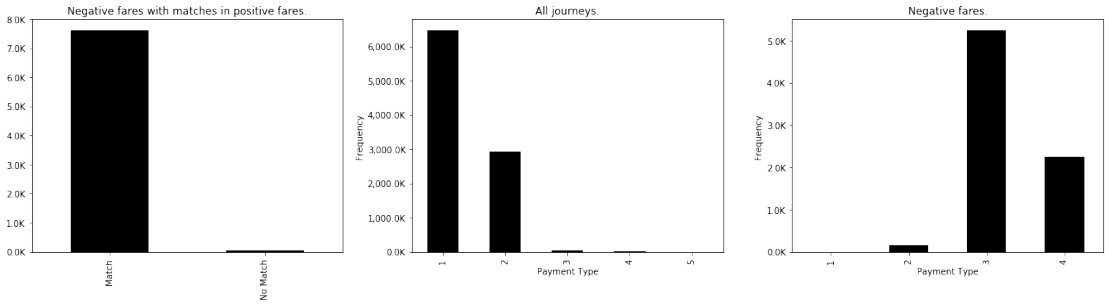

Figure 8: Journeys with negative fares

The distribution of fares present within the dataset is considered and shown in the kernel density estimate plot in Figure 7. This plot actually shows a truncated distribution to allow us to see the shape of the peaks in the centre. The fares actually range from -$465 to $403408, with extremely long tails on both sides of the peaks. The first peak represents what we imagine a typical trip within New York City might cost, the second represents the very common fare of $52- the fixed fare to/from JFK airport.

5.1.3 Negative Fares

The tail of negatively priced trips is considered as potentially erroneous data. For example, the lowest fare is -$465.00. It is unlikely that cab drivers are paying for the privilege of driving their customers around New York City so we try to understand what this negative value represents. Figure 8 shows the majority of negative fares are associated with payment mechanism 3 and 4 ("no charge" or "dispute" respectively) which suggests they may perhaps be reversals of other journeys in the dataset. In order to investigate this idea we consider that if a negative fare represents a reversal of a previous transaction, then it is likely we can pair up the negative fares with corresponding positive fares. A simple matching algorithm is designed that iterates through each of the trips with a negative fare and identifies the corresponding positive fares in the dataset, matching them only if the following features match

- Pickup Location ID
- Dropoff Location ID
- Pickup Timestamp
- Dropoff Timestamp
- Fare, although we match on magnitude, expecting the sign to differ.

When loading the dataset, only a sample is loaded which means that if there are matching pairs of trips, they won't necessarily both appear in the data. To work around

this problem, the entire June 2018 dataset is loaded without sampling to test this hypothesis.

Applying this matching algorithm to the negative fares, it quickly becomes apparent that for each negative fare, there is generally a positive fare that it is reversing. Fig 8 shows this in the first plot. Ideally we would remove both the negative and corresponding positive journeys. Unfortunately because of the size of the dataset, it cannot be loaded in full to run this matching algorithm on. Instead the negative fares are simply removed as the next best option and it is proposed a future project considers whether their inclusion impacts the results.

5.1.4 Very High Fares

The long tail of expensive journeys may also be erroneous. For example, the highest total payment at $403,408 probably does not actually represent a fare paid by a customer. Looking at the features of this individual journey, it is apparently that it was paid for by cash, was zero distance and zero duration and began/ended within the same zone. Yet other journeys with very high fare have corresponding features which may be more credible.

It begs the question of where the line is drawn? When do we imagine that a very wealthy individual chose to spend a lot of money and when do we choose to believe an error occurred during data collection, processing or storage? This question is tackled jointly between the fare features, duration feature and distance feature.

5.1.5 Duration

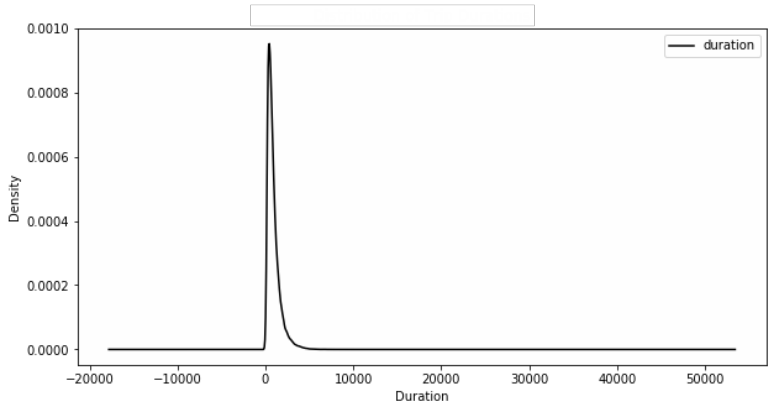

Figure 9: Kernel density estimate plot of duration feature

Figure 9 shows the distribution of the duration feature. There are 94 journeys in the sample with negative duration.

There are some very high values for duration, the highest being 168 hours which is an entire week. Such long journeys seem spurious but it is difficult to draw the line. We note the USA is a large place in which a journey spanning multiple days is possible. We also note that passengers are specifically allowed to negotiate long trips with drivers which may well take them far away from New York City. Rather than simply drop all long journeys, these long duration trips are admitted only when the duration feature is congruent with the distance and fare features.

5.1.6 Distance

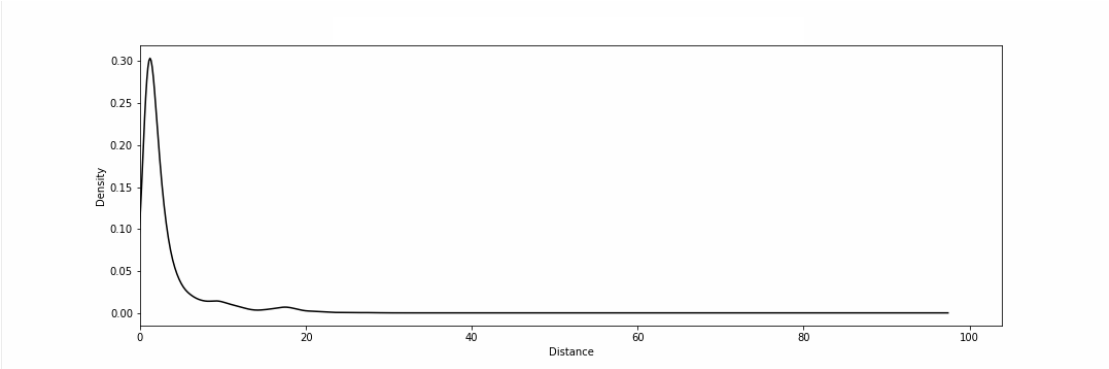

Figure 10: Kernel density estimate plot of distance feature

Figure 10 shows the distribution of the distance feature for a random sample of 10,000 journeys. We can see there are no journeys with negative distance which is as we would expect. We can see there are some journeys with zero distance. This may occur if a passenger changes their mind after the meter is started but before the vehicle moved. There are then three peaks, a very high one at around two miles, and two much flatter peaks at approximately 10 miles and another at approximately 18 miles. These are the distances from LaGuardia Airport and Newark Airport to Manhattan respectively. There does not appear to be a peak representing the distance from JFK Airport (14 miles).

There is a long tail of very high distances which could be erroneous data. One thing we can assert is that if a taxi journey is travelling a very long distance, then it is likely it either started or ended outside of New York City. Figure 11 shows how the distribution of journeys changes between the whole set of trips and those greater than 50 miles. The graph confirms that this is the case, but it is worth noting that the majority of these high fare journeys still take place within the city, its airports and Westchester/Nassau counties. One explanation for such a journey is that it is in fact a round-trip in which the meter was left running throughout. This would be a journey where a passenger has been dropped off close to their pickup, but has driven a long way in the meantime, perhaps to collect something. We can also predict that longer journeys are more likely

to be non-metered fares, either airport trips or negotiated fares. This is supported by the graph in Figure 12 which shows much higher occurrence of these fare codes (2, 3, 4 and 5) than in the general population of trips

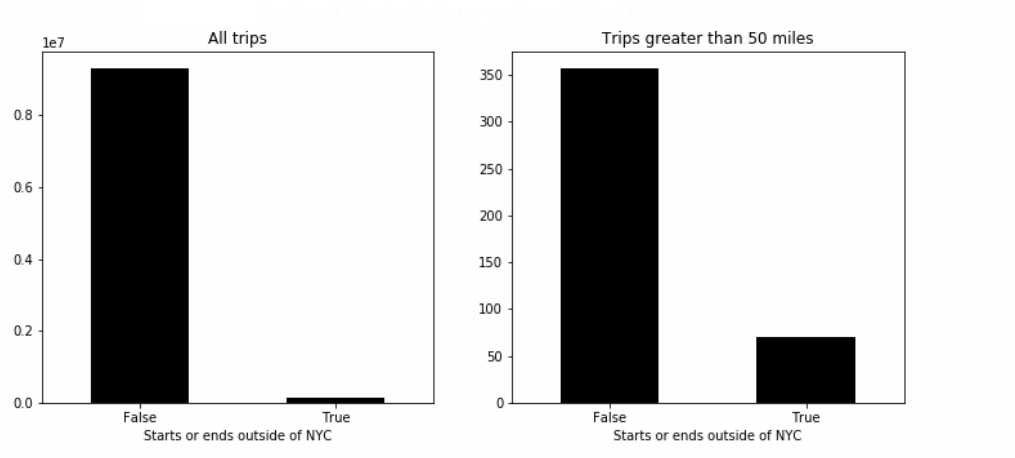

Figure 11: Distribution of journeys beginning or ending outside of NYC

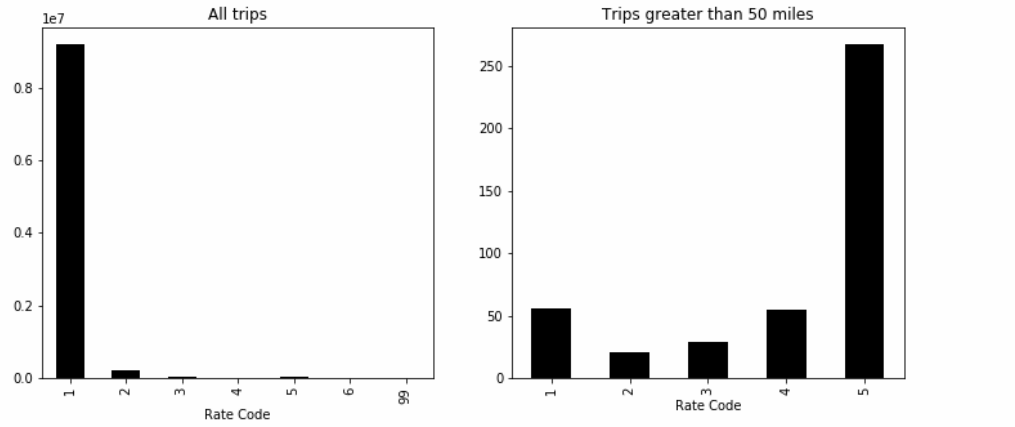

Figure 12: Distribution of fare code

Rather than consider individual cut-offs for each of the fare, duration and distance features, each of the three features are considered together in order to decide which are credible. If a trip has a high distance, a high duration and a high fare then it is more believable than a trip with a very high fare but very low duration. Figure 13 shows three pair plots for these features, and we can see a broadly positive relationship for all three pairings. We can also see a separate cluster of high duration trips which do not obey

this relationship.

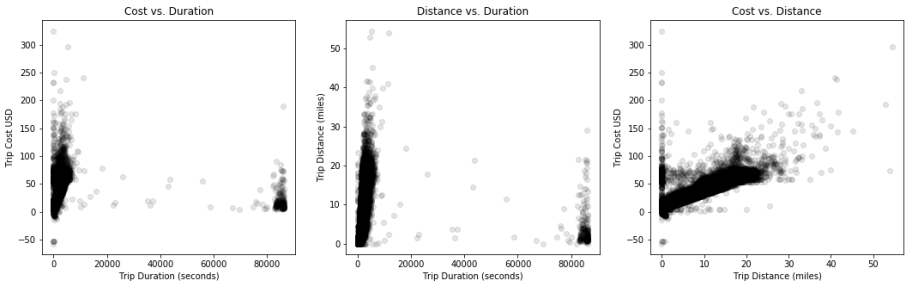

Figure 13: Pair plots of distance, duration and total fare

The first two plots in Figure 13 show a cluster of very high duration trips in excess of 22 hours. This may relate to a clock issue. If the date of the dropoff is out by 1 day compared to the pickup, then journeys of approximately 1 hour would show up as a 23 hour journeys like this. However, the dates of these journeys do not relate to seasonal clock changes or or to a leap year phenomena. Unable to explain this cluster, we proceed by dropping this very small volume of the data.

K-means is used to separate the two clusters of trips and drop those belonging to the second cluster. This is illustrated in Figure 14. The green points represent data in the credible cluster which we retain, the red data is the spurious cluster which we drop. The second plot shows the data after we remove the spurious cluster.

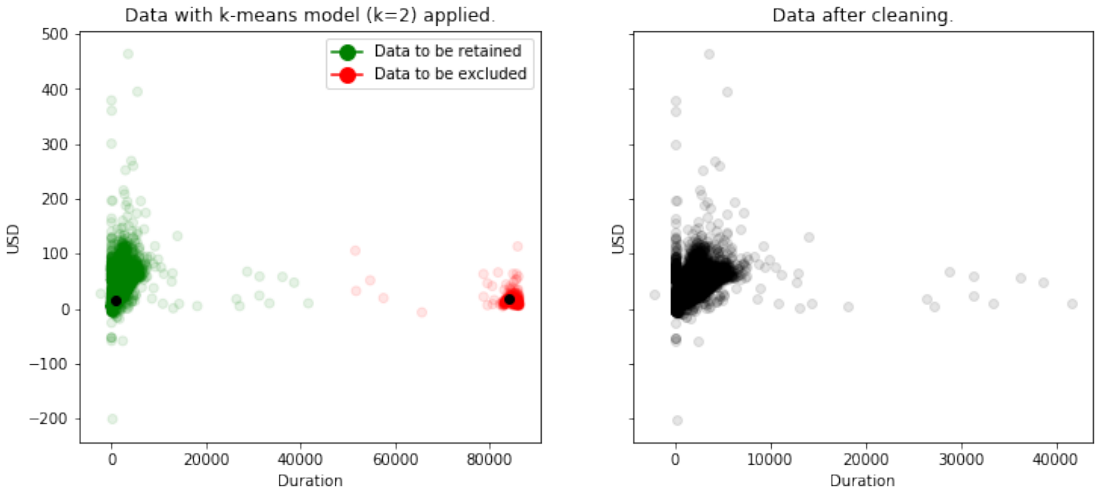

Figure 14: Clustering model of cost vs. duration

The third plot in Figure 13 shows a positive, linear relationship between cost and

distance, as we would expect. However, there are a number of high cost trips of very short or zero distance. It does not seem credible that a user has paid in excess of $100 to drive just a couple of miles.

A linear regression is fitted, predicting cost using just distance as a linear regression. We then apply a tolerance either side of this prediction, and discard any data that sits outside of this tolerance. The effect of this is to to remove data that is not consistent with the model we expect to exist between fares and distance. The benefit of this approach over a simple cut-off for each feature is that we retain very small and very large values of both features if they seem credible but discard them where they do not. There is a risk that with this approach, a specific model has been forced on to the data. However, the data removed only constitutes 0.07% of the overall dataset. This approach is illustrated in Figure 15.

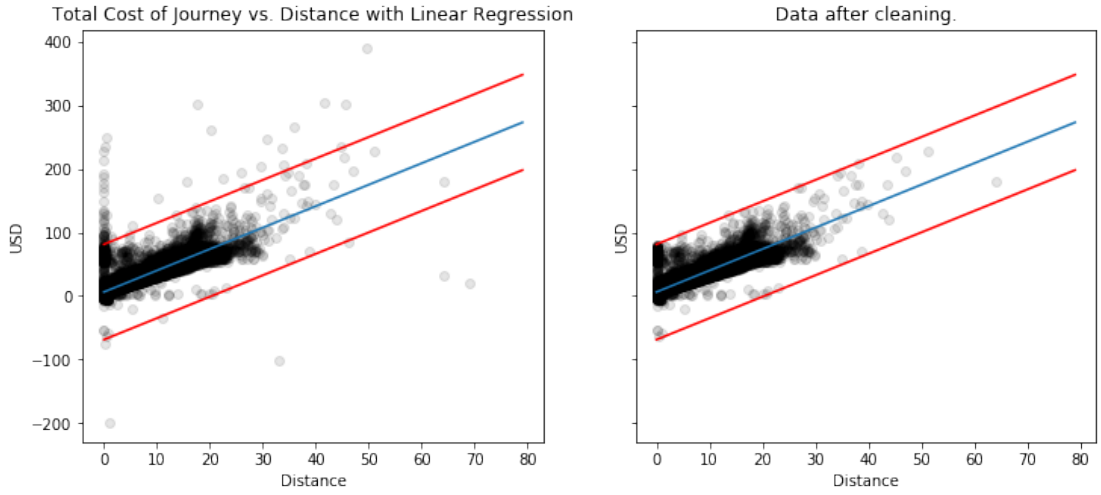

Figure 15: Linear regression model of cost vs. distance

5.1.7 Tabular Trip Data Cleaning Summary

Data cleaning for the tabular data was a particularly challenging exercise. We have had to balance the discovery of data which seems spurious with the understanding that in such a large set of examples, sometimes very unusual phenomena will be accurately recorded in the data. Further still, it may be the case that it is these extreme examples we want our learning algorithms to find. For example, we can well imagine that very long trips of multiple days are erroneous, but if not, that they are associated with exceptionally generous tipping. To remove them as examples from the dataset is a simple approach but there is a risk that any learning algorithm does not explore the extremities of the feature space where potentially the most informative examples sit. Reflecting this, a conservative approach to removing data has been taken.

In order to clean the data, a script was written that performed the following filtering operations to the dataset.

- Removed non-credit card trips.
- Removed spurious passenger count trips.
- Removed trips with negative duration.
- Removed trips with negative fare
- Removed trips with dates outside of range.
- Removed trips with a fare beyond the tolerance predicted given the distance.
- Removed trips with high duration but low fare using clustering approach.

Figure 16 shows the impact of each individual filtering transformation on the size of the dataset. We note that if the filters were applied in a different order, they may have a different size associated with them, but the same set of trips would remain at the end.

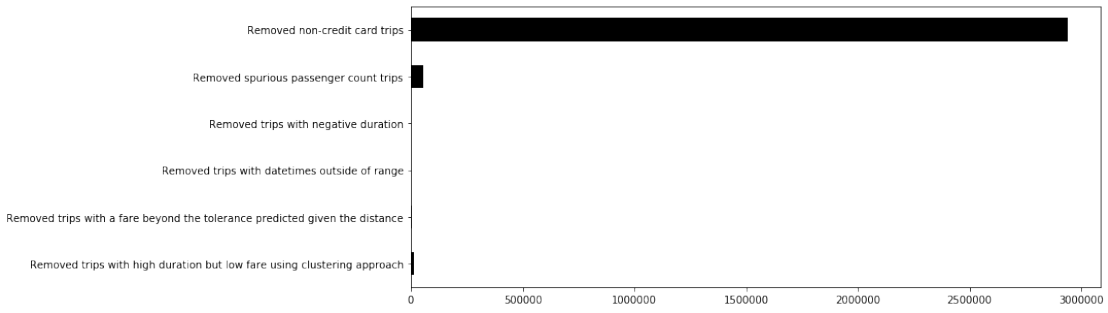

Figure 16: Summary of data cleaning filtering operations

5.2 Spatial Data

The spatial data is loaded in to a geopandas DataFrame from the shape files provided by NYC TLC. Each entry in the dataframe contains

- Zone ID
- Shape Length
- Shape Area
- Borough
- Geometry

The geometry in each case is a polygon or set of polygons which define the boundaries of the zones. The polygons are defined in the NAD83: New York Long Island local coordinate system [39].

There is no missing data in the dataset. The 'Borough' feature is checked for extreme values but does not reveal any. The geometries are also checked to ensure they all lie within the approximate bounds of the New York City area, which they do.

5.3 Weather Data

The weather data includes 30 different features for each observation in the time series. Only two features are used (temperature and precipitation) as these are typically used to describe weather on a day to day basis by those experiencing it. These features are plotted in Figure 17.

When checking missing data, the temperature data was found to contain a single missing value. This is replaced with the actual value from timeanddate.com's [40] historic weather archive for the relevant date and time.

The precipitation data requires more cleaning. The data contains empty values for zero precipitation so requires these entries to be replaced with 0. It also contains "Trace of Precipitation" as a frequent value. Such observations are discussed by Akyüz, Shein and Asmus [41]. They state that a Trace of Precipitation is "not a measurable amount but just enough to wet the rain gauge that it is observed in." This project adopts their proposal that such values are replaced with a very small numeric value: 0.00001.

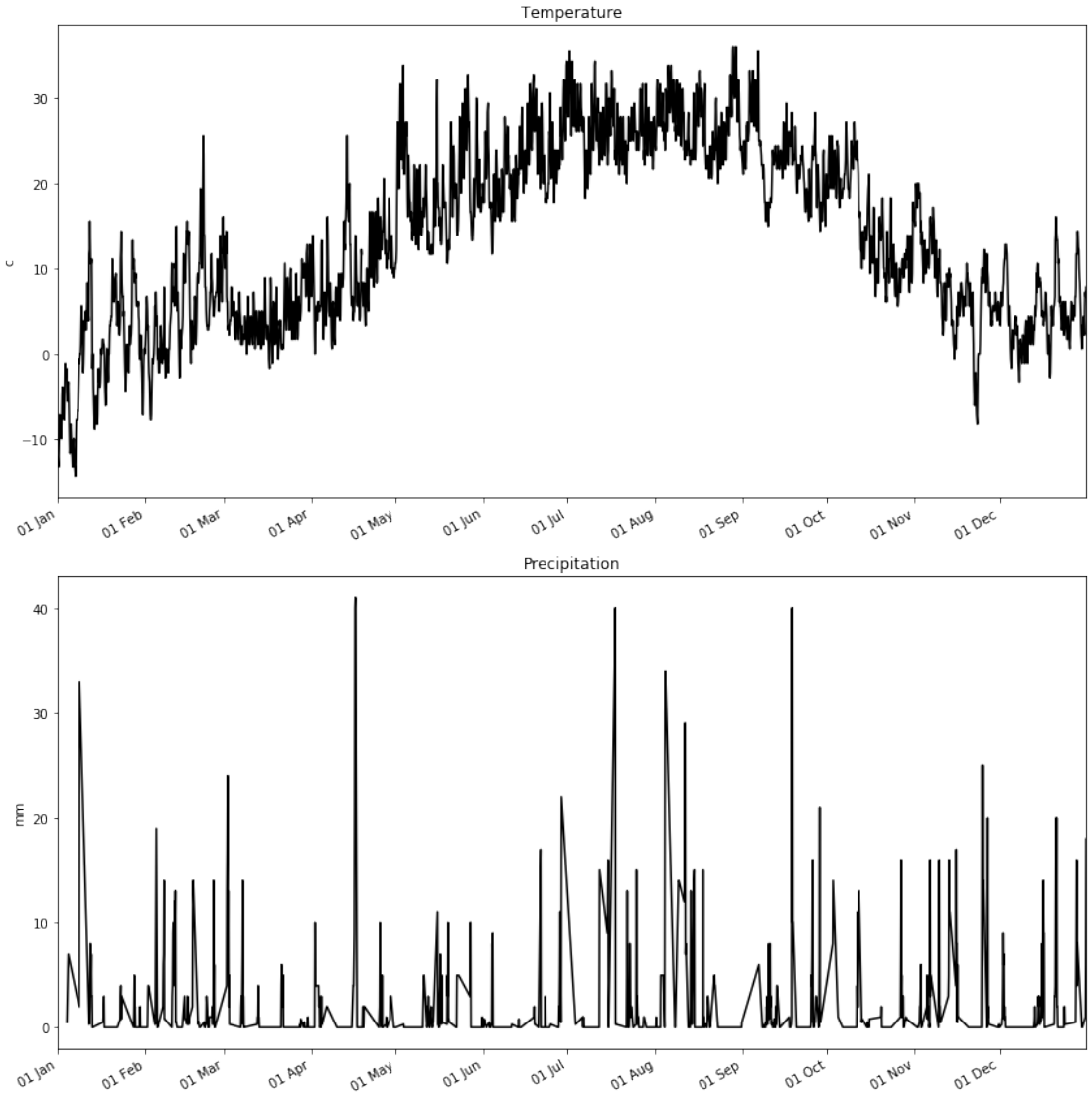

Figure 17: Weather features

6 Overall Approach

6.1 Feature Engineering

A number of new features were created in order to be used within the models. These features are all transformations of existing features from the trip data and spatial data that allow the data to be used by the learning algorithms. The use of dummy variables ensures that categorical data encoded as numbers is not used as a continuous numerical feature. For example, rate code can be any value between 1 and 5, but these are categorical encodings with no ordering. A rate code of 5 is not 'more' than a rate code of 1. An unintended consequence of dummy creation is it creates collinear features. For example, colour_0 and colour_1 are perfectly correlated.

The last category is dropped from the dummy variables for models that are unable to train on data with collinear features.

Alongside this existing list of new features, the following features are included in all models.

6.2 Environment

All explorations and modelling was undertaken in Python running on a computer with 52gb RAM and 8 CPUs on the Google Cloud Platform. This environment offered a significantly higher processing power and memory capacity than a typical desktop computing environment. This allowed loading of a sufficiently large sample of data into memory. Packages used were numpy [42], pandas [43], geopandas [44], geopy [45], sklearn [46], statsmodels [47], keras [48] and matplotlib [49].

6.3 Training and Testing Split

For both the regression and classification models, the data is split into training and test data. The purpose of this split is to hold back data, unseen during training to test the performance of the trained classifier. This ensures we can detect and prevent overfitting. The split of data is two thirds training data to one third test data. This ensures there is sufficient test data for the result to be statistically robust. In some cases this could come at a cost of not having enough training data, but data is plentiful in this context and further data can always be obtained either by increasing the sample size or using a wider date range.

In the case of the neural networks, some of the training data is held back as validation data. This serves a similar purpose to test data but is used during the training process. At the end of each epoch, i.e. when all the training data has been used, the loss and accuracy on the validation data is calculated to detect overfitting during training.

category	Feature Name(s)	Description
Other	colour_0 ... colour_1	For yellow cabs, colour_0 is equal to 1 and colour_1 is equal to 0. The opposite is true for green cabs.
	ratecode_1 ... ratecode_5	This is a set of dummy variables, one for each variable which equals 1 if that was the ratecode for the trip and 0 otherwise.
Spatial	dropoff_borough_Brooklyn dropoff_borough_Manhattan dropoff_borough_Queens dropoff_borough_Bronx dropoff_borough_Staten_Island dropoff_borough_EWR	This is a set of dummy features, one for each borough (as well as Newark Airport) which is equal to 1 if the dropoff occurred in that borough and 0 otherwise.
	pickup_borough_Brooklyn pickup_borough_Manhattan pickup_borough_Queens pickup_borough_Bronx pickup_borough_Staten_Island pickup_borough_EWR	This is a set of dummy features, one for each borough (as well as Newark Airport) which is equal to 1 if the dropoff occurred in that borough and 0 otherwise.
Temporal	temporal_timeInMinutes	This is a numeric representation of the dropoff time recorded for the trip. It is the number of minutes elapsed since 00:00 that day.
	temporal_dayOfWeek_0 ... temporal_dayOfWeek_6	A set of dummy variables, one for each day of the week that equal 1 if the trip began on that day of the week and 0 otherwise.
	temporal_month_1 ... temporal_month_12	A set of dummy variables, one for each month that equals 1 if the trip began in that month

Table 1: New features

Category	Feature Name(s)	Description
	duration	This is the number of seconds between the trip start time and end time
	trip_distance	This is the distance as measured by the vehicle's meter
Spatial	pickup_x pickup_y dropoff_x dropoff_y	These correspond to the coordinates of the centroid of the pickup zone and dropoff zone respectively. These are in the local coordinate system.
Weather	T	This is the temperature at the last weather observation prior to the trip beginning
	RRR	This is the amount of precipitation at the last weather observation prior to the trip beginning

Table 2: Existing features

7 Initial Exploration

7.1 Approach

Prior to model development, the data was explored to identify basic correlations between the features and tipping behaviours. After cleaning, the data was explored by reviewing the distribution of the tip percentage target variable, as well as its correlation with key features including cab colour, number of passengers, temporal features, spatial features and weather.

7.2 Results and Discussion

7.2.1 Tipping Percentage

The tipping percentage appears to be a mixture of underlying Gaussian distributions, each centred around common tipping percentages. A Gaussian Mixture Model is fitted to the data and the means of each component are plotted in Figure 18 against the distribution of the tip percentage.

The GMM estimates that the means of the underlying component distributions are 0.00%, 9.09%, 13.44%, 16.64%, 20.00% and 22.10%, although we can see in the plot the last value of 22.10% is not exactly at the peak of the underlying component distribution.

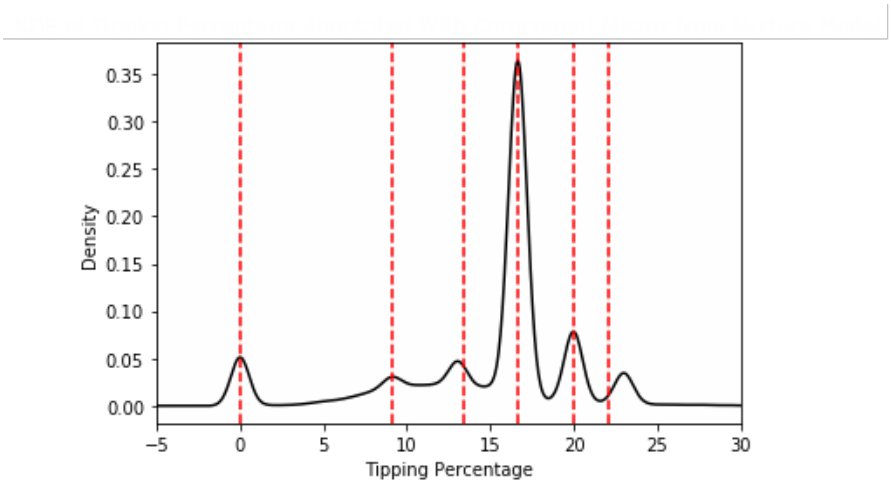

Figure 18: Tip percentage distribution plotted with mixture model means

7.2.2 Cab Colour

We may well expect yellow cabs to be associated with different tipping behaviours than green ones because of the different areas in which they operate. For example, we may well expect more tourists in Manhattan than other areas of New York City. We may expect traffic to be more difficult, which could be either rewarded, or even punished by customers with different tipping rates. This feature is related to the spatial features discussed later.

Figure 19 shows three plots. The first shows the average tipping rate for yellow and green cabs. Yellow cabs have higher average tipping rates than green cabs. The other two plots in the figure show the distribution of tip percentages for yellow and green cabs separately. These two distributions suggest that one of the main reasons the overall mean is lower for green cabs is because of the higher frequency of zero tips. The rest of the distributions look similar between the two, but there is a distinctly higher peak for a tip of zero in the green distribution.

7.2.3 Number of Passengers

As the number of passengers increases, it may be the case that overall tip percentages change. Perhaps passengers who are splitting a fare between more people feel more able to afford a higher tip percentage. Perhaps in a larger group, people feel less able to make generous gestures on behalf of their fellow passengers and take a conservative approach.

Figure 20 shows the average fare for each distinct passenger count in the dataset. It appears there is a relatively small decrease in the average tip percentage as the passenger count increases, bottoming out at four passengers, followed by an increase for the largest passenger numbers of 5 and 6.

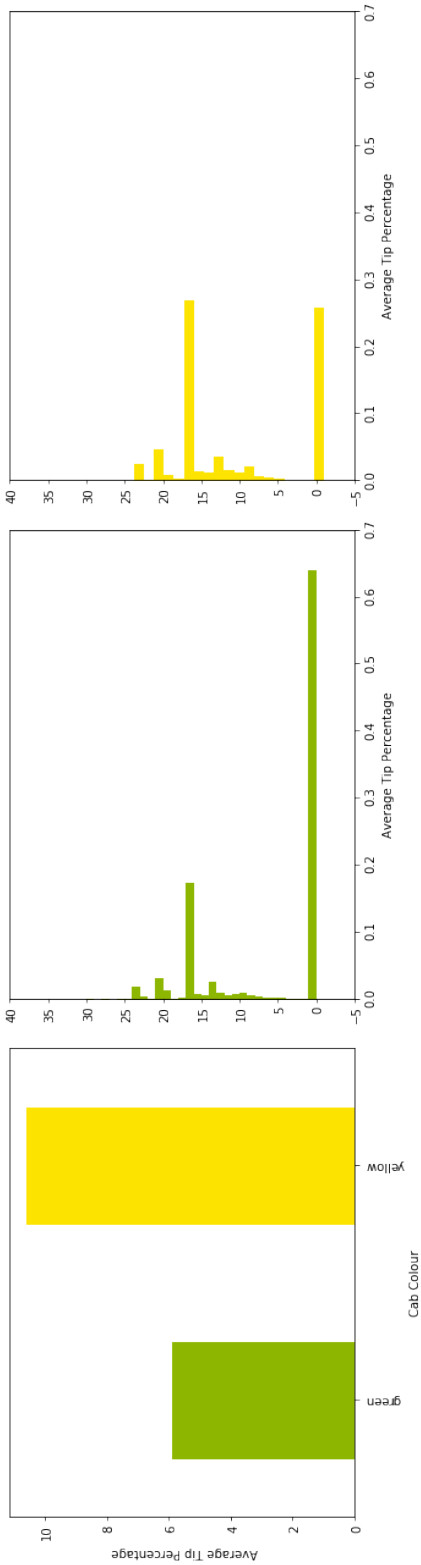

Figure 19: Average tips and tip percentage distributions for green vs. yellow Cabs.

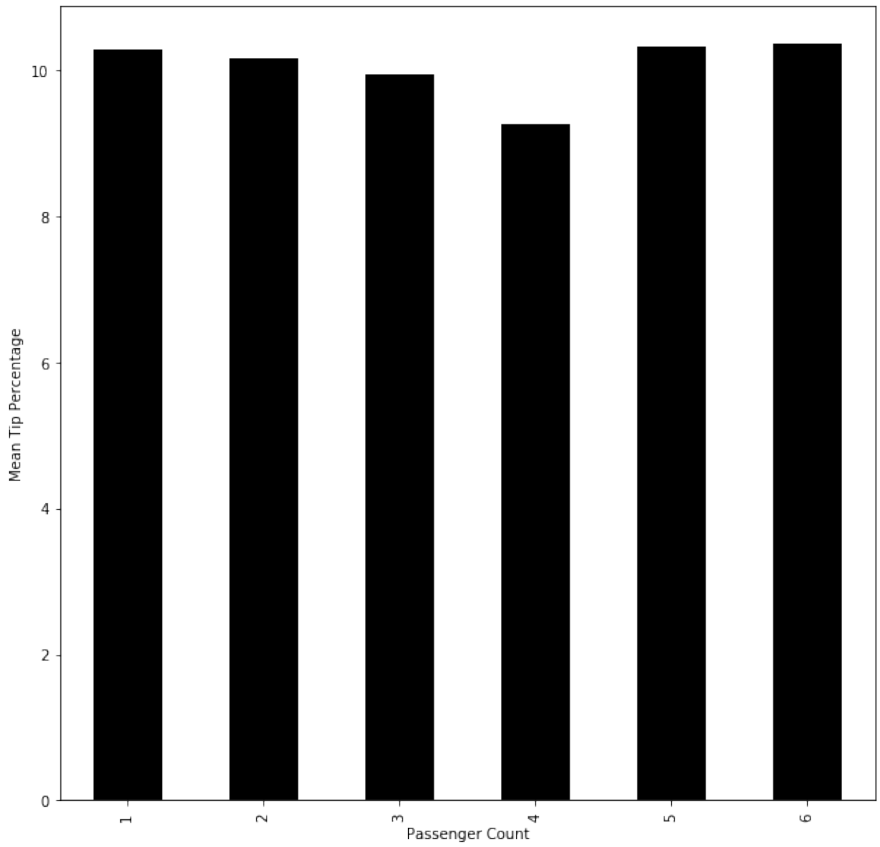

Figure 20: Average fares segmented by passenger count.

7.2.4 Temporal Features

Depending on the time of day and also the day of week, we may expect the average tip percentage to vary. Perhaps those travelling in the middle of the night acknowledge a higher burden of working in the night and compensate their drivers with higher tips. Perhaps those travelling in the night are more likely to be doing so for leisure purposes, and therefore in better spirits driving higher generosity. Or perhaps such leisure journeys are less likely to be charged to an employer and perhaps leisure passengers are more conservative with their own money than that of their employer.

Figure 21 shows the average tip by day of the week and also by the hour of the day. For reference, there are also plots showing the number of trips for the same days of week/hours of day. A few trends emerge. Despite being the busiest day in terms of number of trips, Saturday has the lowest average tip percentage. Generally, for a given day, the higher the number of trips, the lower the average trip percentage.

The pattern for hour of day suggests a bi-modal distribution for tip percentage. The first peak occurs in the morning commute hours, then it flattens out throughout

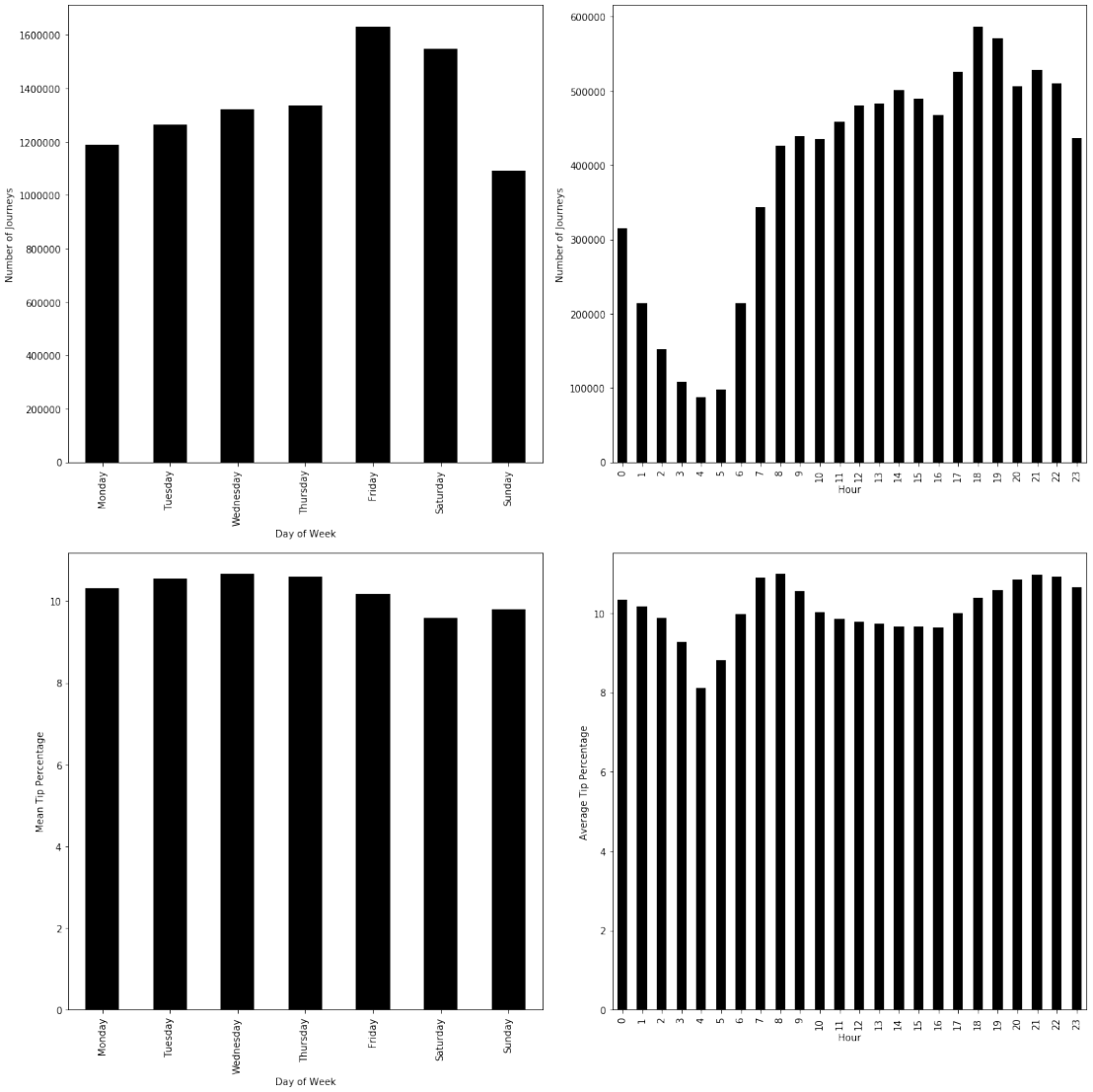

Figure 21: Number of journeys and average fare by day-of-week and hour-of-day

the working day, and then increases to a second peak in the late evening hours before dipping more dramatically in the early hours. Unlike day-of-week, the tip percentage is positively correlated with number of trips for a given hour of the day.

7.2.5 Spatial Factors

Journeys in different parts of New York will be associated with different tipping behaviours, reflecting the socioeconomic diversity of the city. The spatial features are also

likely a proxy for other hidden features, e.g. journeys in Manhattan may be more challenging and stressful for drivers and this may be compensated with better tips. A trip always consists of a pick up location and a dropoff location and we can always seek out spatial trends using either of these features. It is not obvious that one is more valid than the other and so the same analysis is undertaken for both.

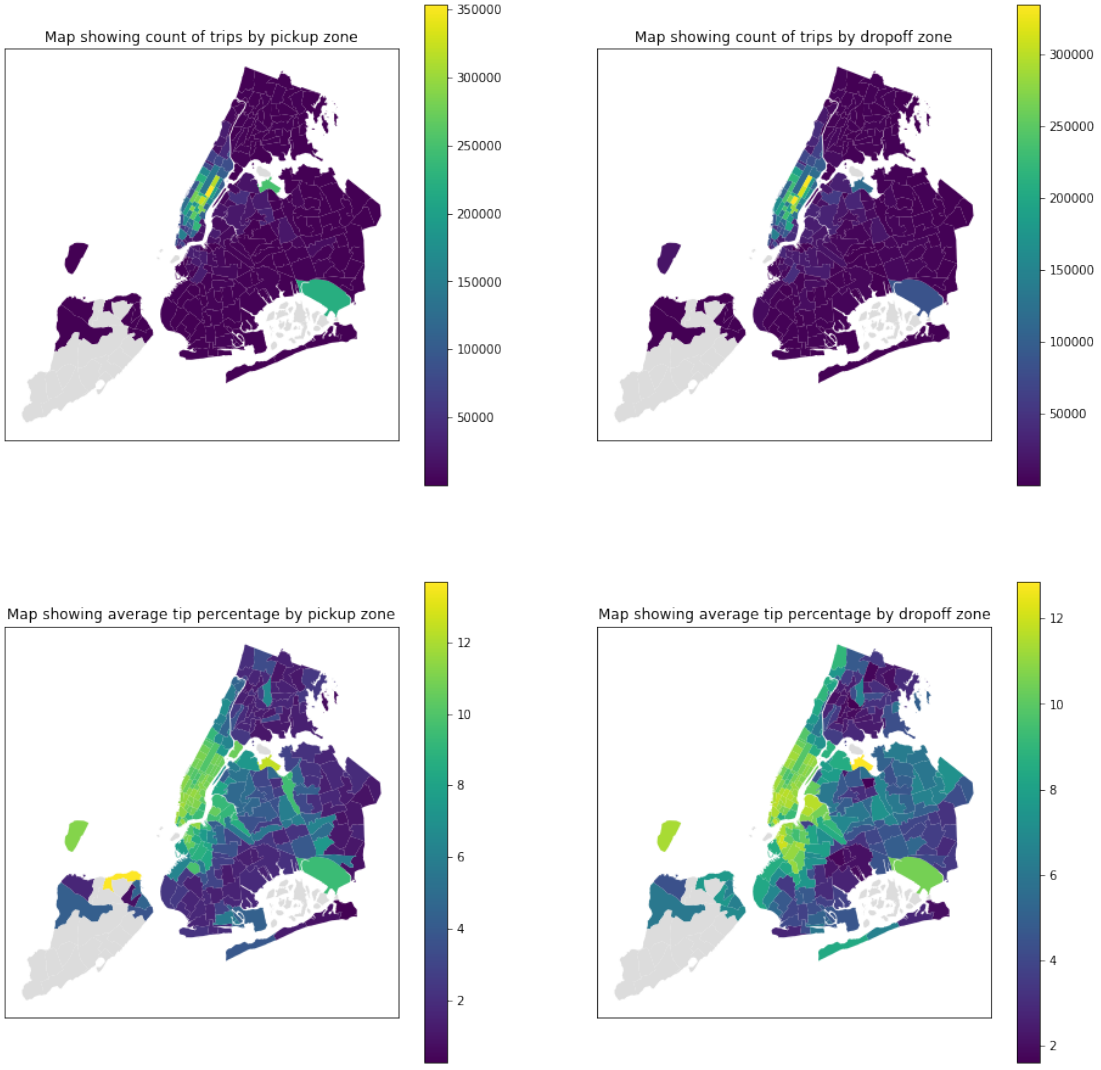

Figure 22: Count of trips and mean tip percentage by pickup and dropoff Zones

Figure 22 shows four heatmaps. Two of them show the count of trips beginning and ending within each zone. The other two show the average tip percentage for journeys beginning and ending within each zone. A number of zones are shown in gray. These are zones where there is no data or very little data. They are generally areas of low

population.

The pattern that emerges is that the busiest areas, for both pickup and dropoff are Manhattan, LaGuardia Airport and JFK Airport. The trend in tips follows this pattern, the highest tips are in Manhattan, and all three airports (Newark, LaGuardia and JFK). Average tips in the outer boroughs are generally low and there are fewer journeys here.

7.2.6 Weather

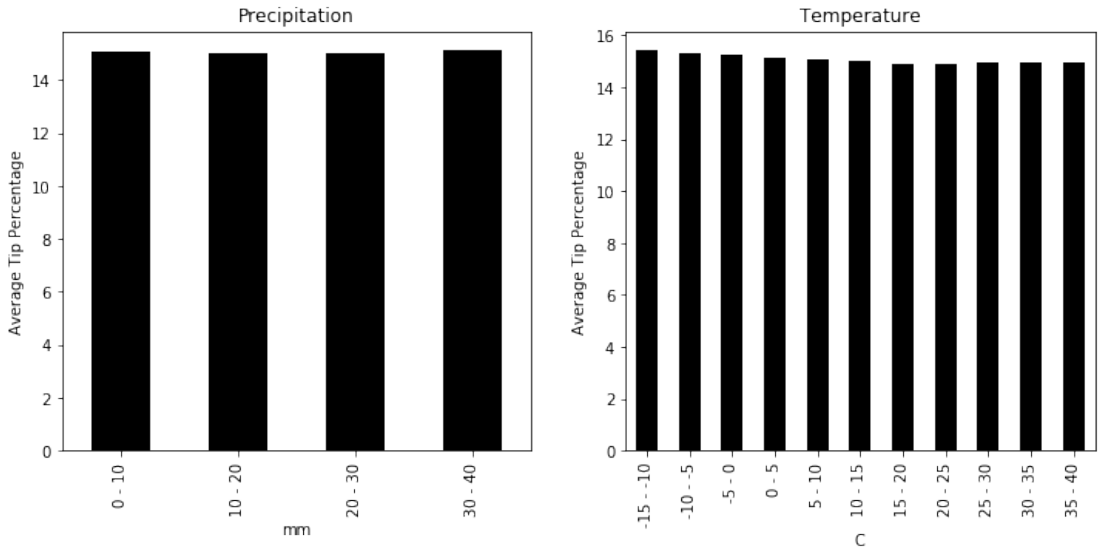

Figure 23: Relationship between average tip percentage and temperature, precipitation

Figure 23 shows two bar charts. The first shows the average tip percentage for different equally sized bins of precipitation, the second for bins of temperature. It shows very little difference between the mean tip percentage in each bin. However, there is a very slight decreasing trend in mean tip percentage as the temperature increases. There is no apparent relationship between precipitation and mean tip percentage. For the temperature feature, a formal hypothesis test is conducted on whether the mean tip percentage differs between trips when the temperature is above 20 degrees and trips when the temperature is below 20 degrees. A hypothesis that the two means are the same is rejected, which we interpret as the tipping rate in these two cases is statistically significantly different ($p < .001$). However, it is clear that the magnitude of the difference is extremely small.

8 Predicting the Tip Percentage - Regression Approach

8.1 Approach

In order to predict the value of the tip percentage, three different regression approaches are designed. These are all supervised learning techniques that predict a real number: the tip percentage.

8.1.1 Linear Regression

Linear regression is the simplest technique applied. The linear regression is of the form

$$y = w_0 + \mathbf{w}.\mathbf{x} \tag{8.1}$$

Where y is the tip percentage for a trip and $\mathbf{x}$ are the features associated with that tip.

The spatial coordinate features are scaled down by a factor of 1000 and centred on zero to make the corresponding weights easier to interpret.

8.1.2 Random Forests Regression

A random forests regression is trained on the full feature set using 100 trees with a maximum depth of 10. The maximum depth limits overfitting, as does using a large number of individually trained trees. These hyperparameters are chosen using trial and error.

8.1.3 Neural Network Regression

A neural network is designed and trained to predict the tip percentage. The structure of the network is a 5 layer network with 64 nodes in the first layer, 32 nodes in the second, 16 in the third and 8 nodes in the fourth layer. There is a single output node in the final layer. Each node in all but the final layer has a rectified linear activation function applied. This function is of the form

$$f(x) = max(0, x) \tag{8.2}$$

The final layer is a single node with a linear activation function which simply sums its weighted inputs.

All nodes are fully connected between layers. Each feature was normalised prior to training. The choice of network shape was chosen in order to ensure the largest layer was sufficiently large to express the 43 input features, and sufficiently deep in order to find complex, non-linear boundaries that are expected to appear in the dataset, whilst also being trainable within the computing resources available and within a reasonable time.

8.2 Results and Discussion

8.2.1 Linear Regression

The results of the linear regression are shown in Table 3. The r^2 score of the regressioon is 0.08. This is interpreted as 8% of variation in the tip percentage value is explained by variance in the features. This was calculated on unseen test data to ensure the regression has not become overfitted.

For each feature, a hypothesis test is run to test whether the weight significantly differs from 0. A number of features are not found to significantly differ from 0, and the model is rerun without the associated features. They are: (1) Passenger Count (2) Pickup Borough is Brooklyn (3) Pickup Borough is Manhattan (3) Monday (4) Friday (5) Saturday (6) March (7) April (8) Temperature (9) Rainfall. We interpret this as variance in these features does not significantly impact the tipping rate. For the dummy variables which are part of wider sets, this is to say that tipping rates for that particular value of the underlying feature (e.g. Day of Week) are not significantly different from the average.

Although the linear regression has the lowest r^2 score of any of the regression approaches, it is a highly explainable model. For each feature we can interpret the weight as the impact on the tipping rate for a unit change of the feature. For example, being a yellow cab (colour_0 = 1) adds a premium of 3.35 percentage points to the average tip. Moving 1000 units in the y direction adds a premium of 0.02 percentage points. The dummy variables are insightful, showing the tip improves most for a trip either picked up or dropped off in Queens. The Bronx has the lowest premium. The weather features are significant but very small, with a 0.01 percentage point drop in tips for every increase of 1 degree celcius and a 0.01 percentage point increase in tips for every 1mm of precipitation.

The temporal factors are all very small suggesting they have not been influential in the model. The linear model is challenged with these features, and it is not surprising to see a very small weight against the timeInMinutes feature. It's likely that tips associated with certain times of the day would be discrete within the day, e.g. during the morning rush. The linear model enforces that a change within the feature always results in the same magnitude and sign of effect on the target and that will clearly not be true here.

The distributions plotted in Figure 24 show the distribution of predicted values of tips using the linear regression with the test data against the actual values of tip percentages in that data. It is clear that the regression is able to predict the peak of tips around the 17% peak, but not some of the smaller peaks around 0%, 20% and 23%.

8.2.2 Neural Network Regression

The neural network is trained and the loss function stops decreasing after just a few epochs. It performs slightly better in predicting the tips, with $r^2 = 10.7\%$, a marginally higher r^2 than the linear regression. Although able to explain more of the variance in tips than the linear regression model, the neural network is more difficult to interrogate and explain.

	Weight
const	-5.58
pickup_x	-0.03
pickup_y	0.02
dropoff_x	-0.05
dropoff_y	0.02
passenger_count	-0.01
duration	0
dropoff_borough_Bronx	-1.43
dropoff_borough_Brooklyn	2.75
dropoff_borough_Manhattan	2.32
dropoff_borough_Queens	3.91
dropoff_borough_Staten Island	-1.85
pickup_borough_Bronx	9.71
pickup_borough_Brooklyn	14.97
pickup_borough_Manhattan	15.52
pickup_borough_Queens	17.33
pickup_borough_Staten Island	9.06
colour_0	3.35
temporal_dayofweek_1	0.09
temporal_dayofweek_2	0.09
temporal_dayofweek_3	0.12
temporal_month_1	0.11
temporal_month_8	-0.09
temporal_month_9	-0.14
temporal_month_10	-0.1
temporal_month_11	-0.23
temporal_timeInMinutes	0
T (temperature)	-0.01
RRR (precipitation)	0.01

Table 3: Weights for statistically significant features in linear regression

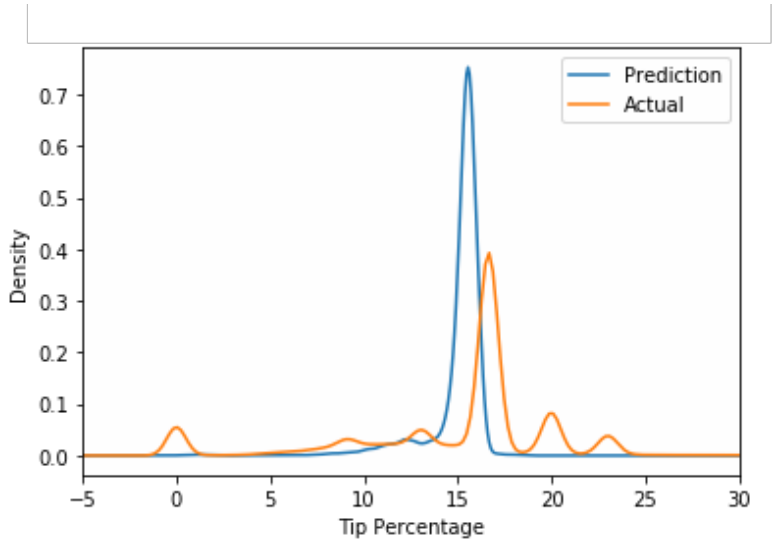

Figure 24: Distribution of predictions vs. actual tip percentages for linear regression

The distribution of predicted tips vs. acutal tips is shown in Figure 25.

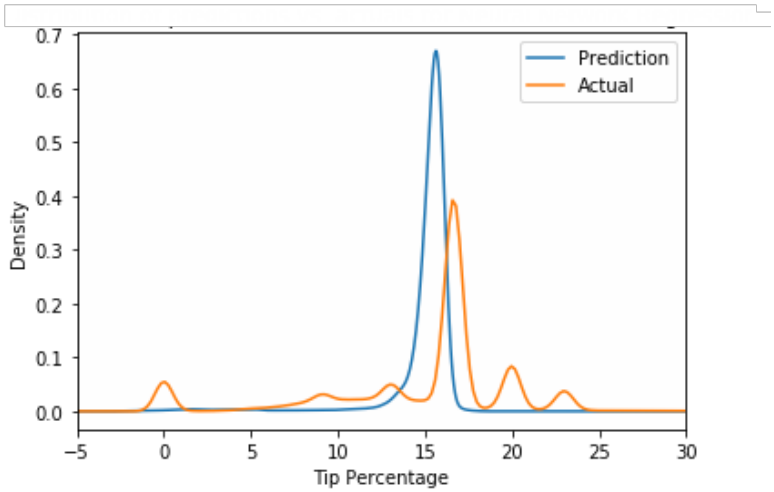

Figure 25: Distribution of predictions vs. actual tip percentages for neural network

8.2.3 Random Forests Regression

The random forests regression achieves a r^2 of 12.5% on the unseen data. This is the highest of any of the regression approaches.

The distribution of predictions vs. actuals for the random forests model is shown

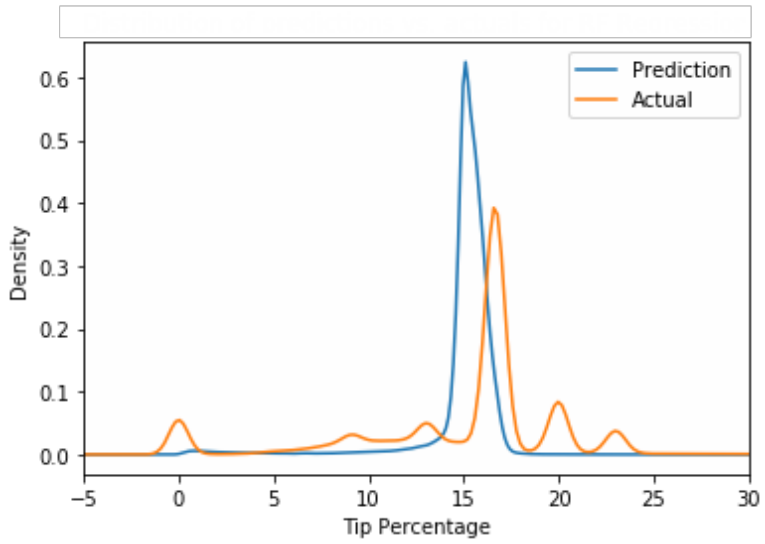

Figure 26: Distribution of predictions vs. actual tip percentages for Random Forests

Like the linear regression, the distribution is generally single peaked. The regression has found the largest component of the mixture but is unable to predict either very high tips around 0%, 20% or 23%.

The approach is also more explainable than the neural network. Feature importance can be calculated for each feature. This is done on a tree basis and then averaged across the trees in the random forest. For each feature, the tree is traversed and at each point where a feature is used to split the training data, the average decrease in variance achieved by that split is accounted to a feature.

Figure 27 shows these feature importances. The five most important features by this measure is the colour of the cab (yellow vs. green), the duration of the trip, one of the spatial coordinates, the time of day and the trip distance.

The individual decision trees are able to use features such as time and duration in a more sophisticated way than the linear regression. They are, for example, able to identify specific periods of the day such as the morning rush, by splitting twice on timeInMinutes. Their ability to use these features more effectively may be one of the reasons for its superior performance.

What is clear across the three regression approaches is that although some perform better than others, all are performing modestly. The vast majority of variation in tips is not explained and this suggests there are elements of the true model we simply haven't captured. It's likely that a large amount of variance would be explained by the individual generosity of taxi customers. Are they travelling for business or leisure? Are they affluent with disposable income? This data on the customers is not something we have. Similarly, features inate to the individual drivers would also explain significant variance:

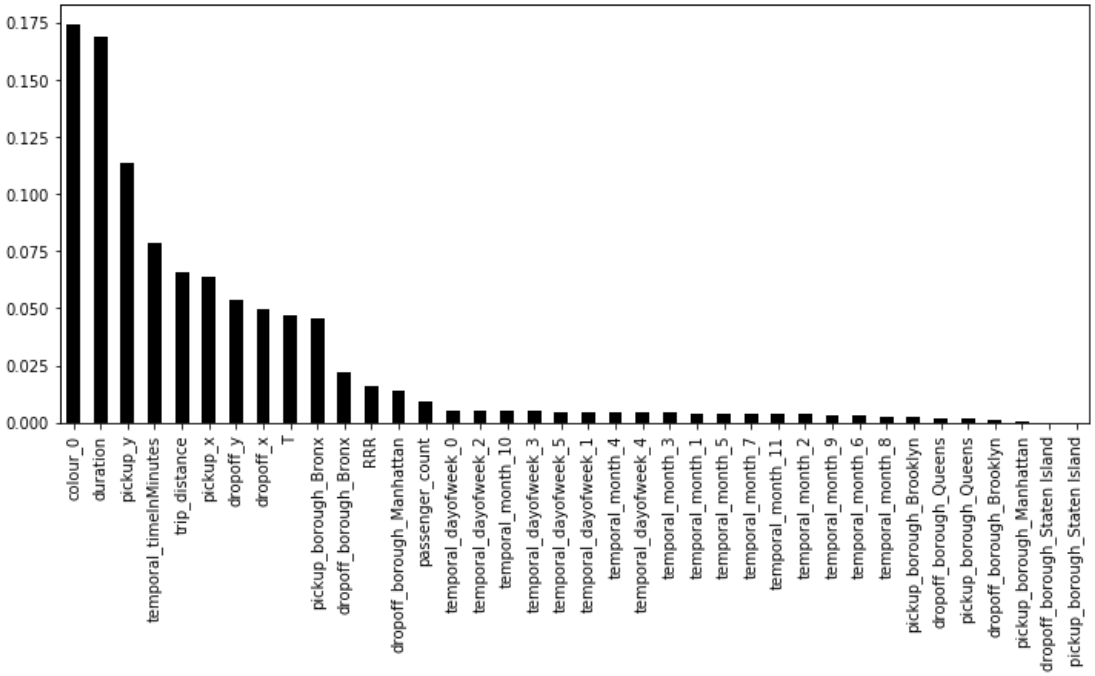

Figure 27: Feature importances in trained Random Forest

charm, willingness to help, perceived competence are also missing. These features have been used by other authors, e.g. Koehrsen [27], who uses the driver identity in their model. This feature is not available now as it was removed from the dataset following a discovery that despite being hashed to protect the real identities of drivers, this process was reversible through the use of a rainbow table [50].

We conclude that with the data available, this is the best accuracy with which tips can be predicted. The predictive power we do have is not insignificant in either a statistical or real sense. The most useful insights are those on the importance and impact of the spatial features. These provide actionable insights for taxi drivers, for example going to specific boroughs or travelling in the positive direction of the y-coordinate is likely to increase tipping rates. Other insights are less actionable, for example we have seen the importance of cab colour as a feature, but this is not something a driver is realistically able to choose. What is very apparent is that all the regressions typically predict a single peaked distribution instead of the many-peaked distribution apparent in the actual tip values. Perhaps further progress could be made on this problem if we could predict values that are not part of this single peak. This is the basis of the the second research question: can we predict when a tip will be zero?

9 Predicting Zero Tips - Classification Approach

9.1 Approach

In order to predict when a tip will be zero, a classification approach is used. In order to apply classification, the target variable must be a class. Therefore a new target is generated: 'tip bin' which is either '0' or '1.' These values correspond to 'zero' or 'above zero' tips respectively. Zero tips are those where the tip percentage is less than 0.1. This ensures that trivially small tips, e.g. '1 cent' are classed as zero. The dataset is balanced by removing journeys with non-zero tip so that the number of trips in each class is equal.

As in the case of regression, the data is split into training data and test data. The test data is not used during training but at the end to test the performance of the trained classifier. A validation set is also held back in the specific case of the neural network for validation during the training process.

9.1.1 Naïve Bayes

A number of classification approaches are trialled, the first being Naïve Bayes. This was selected because it's simple. It serves as a baseline for classifier performance against which we compare more sophisticated classifiers. Logistic regression is also used which is again a relatively simple classifier not expected to perform well due to its underlying parametric assumptions.

9.1.2 Tree-Based Classifiers

Three tree-based classifiers are trained and evaluated. The first is a single decision tree, using the gini criteria for splitting and a maximum depth of 5. The maximum depth constraint is included in order to limit overfitting of the tree and also to promote explainability of the trained decision tree. A constraint of a minimum 500 trips in each terminal node is also applied to limit overfitting. It's observed that without this constraint, the tree does generate terminal nodes with very few trips.

The decision tree approach is then extended using ensemble methods Random Forests and XGBoost. Random Forests is trained with 1000 individual trees, each with a maximum depth of 5 splits. Each tree is trained on a random subset of the data and random subspace of the features. After training, the classifier works by taking the majority vote across the individual trees. This classifier trains relatively quickly, whilst avoiding overfitting by limiting the amount of data and features each individual tree can use. XGBoost furthers the random forests approach, using stochastic and regularized gradient boosting as described in the background section to improve performance.

9.1.3 Neural Network Classifier

The last approach was a neural network classifier. The structure of the network is a 5 layer network with 64 nodes in the first layer, 32 nodes in the second, 16 in the third and 8 nodes in the fourth layer. Unlike in the regression neural network there are 2

	Accuracy	Precision	Recall	F1 Score	Area Under Curve
XGBoost	0.85	0.24	0.56	0.33	0.77
Naive Bayes	0.81	0.18	0.48	0.26	0.68
Random Forests	0.84	0.22	0.54	0.31	0.75
Logistic Regression	0.67	0.10	0.51	0.17	0.61
Decision Tree	0.85	0.23	0.52	0.32	0.73
Neural Net	0.87	0.24	0.43	0.31	0.71

Table 4: Classifier performance metrics

output nodes in the final layer. They are trained to simultaneously output (1,0) in the case of a zero tip and (0,1) in the case of a non-zero tip. Each node throughout the network has a sigmoid activation function, which reflects this is a binary classification problem. All nodes are fully connected between layers. The choice of network shape was chosen in order to ensure the largest layer was sufficiently large to represent the 43 input features, and sufficiently deep in order to find complex, non-linear boundaries that are expected to appear in the dataset, whilst also being trainable within the computing resources available and within a reasonable time. The choice of two output nodes and the sigmoid activation function reflects that this is a binary classification problem. Finally, each feature was rescaled prior to training. For each value of each feature, the minimum value of that feature is subtracted and then divided by the range of the feature. The result of this is that every feature is scaled between 0 and 1. This promotes faster convergence of the weights as otherwise at the beginning of the training process the network would be biased towards the largest scaled features.

9.2 Results and Discussion

Evaluation metrics of the classifiers are reported on test data in Table 4. Prior to training, 33% of the data is held back to serve as unseen data to test the classifier performance against. These metrics are reported for the classifier's performance on unseen data. This is done to ensure that the classifiers have not been overfitted to their training data.

Reported for each classifier is accuracy, precision, recall and F1 Score. Precision and accuracy both refer to a specific class. We are interested specifically in predicting zero-tips, so this is the class they are reported for.

The metrics reported in Table 4 are those on unseen data. Unlike the training data, the test data is not balanced, as it would not be if being used to predict future trips on unbalanced data.

Accuracy indicates the percentage of trips that were correctly classified as either zero or non-zero by the classifier. This metric has limited use on the unbalanced data, but is reported for completeness. We note that for accuracy, a classifier that simply predicted a non-zero tip for every trip would perform with higher accuracy than any of the real

classifiers. This demonstrates why the accuracy measure is limited in this context as clearly such as classifier would not be useful. Instead we accept lower accuracy focusing on precision and recall as our metrics of interest.

Precision is the number of trips classified as 'zero tip' that are correctly classified. Recall is the proportion of actual zero tip trips that the classifier identifies as zero tip. These two measures are often in conflict: to increase precision comes at a cost of reducing recall and vice-versa. The F1 score is a balanced measure that takes both in to account.

The highest perfosectionrming classifiers in terms of precision are XGBoost and the Neural Network, but the Neural Network is the lowest performing classifier in terms of recall whereas XGBoost is the highest. This is reflected in the F1 score which is highest for XGBoost. This suggests that XGBoost is more successfully discriminating between the two classes whilst the Neural Network is achieving high precision by being conservative in what it labels as 'zero tip' at a cost of less frequently recalling them.

The logistic regression is much worse performing than the other classifiers in terms of precision. Just 10% of those it labels as zero-tip are correctly classified. This is perhaps not surprising as the logistic regression is unable to use some of the spatial features and temporal features to identify non-linear decision boundaries as some of the more sophisticated classifiers can. The data is not meeting the parametric assumptions of the logistic regression model. We also note that recall for the logistic regression is modest, placing it in the middle of performers.

Naive Bayes is also a relatively poor classifier with the second lowest precision and the second lowest recall metric. This perhaps reflects that it is the most simple of the classifiers. It is not subject to the same parametric restrictions as the logistic model.

The family of decision tree approaches which includes the single decision tree, the random forest and XGBoost seem the most effective. This reflects that they are able to split where fundamentally different processes are generating the data and that is captured within the features. For example, they are able to split yellow and green cabs in to different branches which is what we see them go on to do. Through this they achieve the three highest F1 scores of the classifiers.

Finally, we review the receiver operating characteristic (RoC) curves for each classifier. These are computed and plotted in Figure 28. The classifiers return an estimate of the probability a trip will have a zero-tip and if that probability is above a specified threshold then that trip is classified as zero-tip. This threshold is varied between 0 and 1 to trace the RoC curves. The further the curves extend towards the top left of the plot, the more successful the classifier is in discriminating between the two classes. The area under the curve serves as a metric for this ability.

It confirms what is previously noted: that the highest performing classifiers are the tree-based classifiers with XGBoost the most able to discriminate between the two classifiers, and the logistic regression the poorest performer. The neural net and Naive Bayes perform somewhere in the middle.

As well as assessing classification performance, some of the classifiers offer insights into why certain trips may be associated with zero trip.

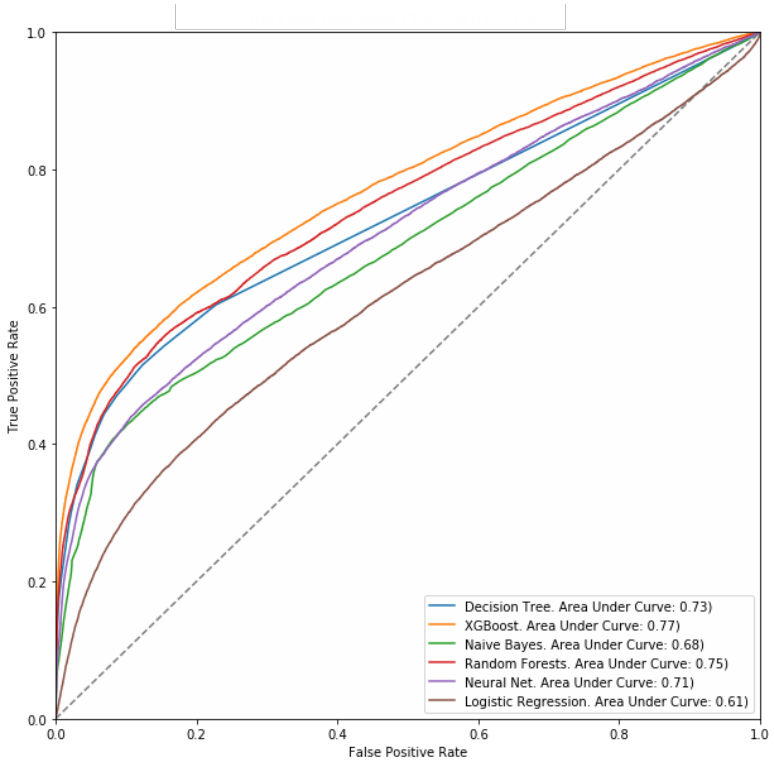

Figure 28: Receiver operating characteristic curves

9.2.1 Insights from the Decision Tree

We can inspect the trained decision tree to further understand which features it has selected and how successful they are in classifying the trips in the training data.

The tree first splits on whether the dropoff borough was Manhattan or not. This is unsurprising as initial exploration showed tip percentages were typically higher in Manhattan than other boroughs. Traversing the tree, we can see that actually the tree only predicts non-zero tips in two cases. The first is when the dropoff borough is in Manhattan, the pickup y coordinate is less than a certain threshold, the pickup borough is not Brooklyn, the dropoff y coordinate is less than a certain threshold, and the trip distance is above 0.295 miles. The second is when the dropoff borough is not Manhattan but the pickup borough is, the dropoff y coordinate is less than a certain threshold and the duration is less than 37 minutes. The thresholds it finds for y coordinates roughly correspond to locations above or below Central Park. In all other cases it predicts a zero tip. Generally we are seeing the spatial features are the most important in determining whether a tip is zero or not within the tree.

Feature importances are also reviewed as they were in the regression approach which confirm this conclusion, the most important features in predicting zero tips are the

	weight	exp(weight)
const	3.40	29.88
pickup_x	0.09	1.10
pickup_y	-0.04	0.96
dropoff_x	0.25	1.29
dropoff_y	-0.05	0.95
passenger_count	0.05	1.05
trip_distance	-0.03	0.97
duration	0.00	1.00
dropoff_borough_Brooklyn	-1.54	0.21
dropoff_borough_Manhattan	-1.94	0.14
dropoff_borough_Queens	-2.17	0.11
colour_0	-1.69	0.19
temporal_dayofweek_2	-0.07	0.93
temporal_dayofweek_3	-0.07	0.93
temporal_dayofweek_5	0.11	1.12
temporal_month_1	-0.16	0.85
temporal_month_2	-0.21	0.81
temporal_month_8	0.15	1.16
temporal_timeInMinutes	-0.00	1.00

Table 5: Weights for logistic regression

spatial features. These are shown in Figure 30. The five most important features by this measure were dropoffs in Manhattan, the y coordinate of the pickup, pickups in Manhattan, the duration of the trip and the x coordinate of the pickup. We note these are all spatial features.

9.2.2 Insights from the Logistic Regression

Although the logistic regression was a relatively poorly performing classifier, it is one of the most explainable approaches. The weights of the fitted regression, transformed into odds ratios, are listed in Table 5. These are to be interpreted as the percentage change in the odds of a zero tip being predicted. For example, an additional passenger increases the odds of a zero tip by 4%, an additional mile decreases the odds of a zero-tip by 2%.

The sign and the magnitude of each of the weight can be interpreted. Again we see the biggest effects seem to be the spatial features. In particular the dummy variables for specific boroughs. For example, pickups in Manhattan, Brooklyn or Queens (which includes JFK airport) decrease the odds of a zero-tip. A cab being yellow decreases the odds of a zero tip by 81%.

Some of the temporal features are also significant, it being August increases the odds of a zero tip by 8% whilst January and February, both have lower than average odds of a zero tip. Wednesdays and Thursdays seem to have slightly lower odds of a zero tip,

whilst Saturdays appear to increase the odds of a zero tip by 12%, something we also saw in the exploration section. The timeInMinutes feature is significant. This is perhaps surprising given we do not expect the logistic function to be able to effectively use the information encoded within this feature. However, the weight associated with it means the magnitude of impact on the odds this feature has is extremely small.

The weights associated with a number of features were not found to be significantly different than zero in a Wald test:

- pickup_borough_Bronx
- pickup_borough_Brooklyn
- pickup_borough_EWR
- pickup_borough_Manhattan
- pickup_borough_Queens
- dropoff_borough_Bronx
- dropoff_borough_EWR
- temporal_dayofweek_0
- temporal_dayofweek_1
- temporal_dayofweek_4
- temporal_month_3
- temporal_month_4
- temporal_month_5
- temporal_month_6
- temporal_month_7
- temporal_month_9
- temporal_month_10
- temporal_month_11
- T (temperature)
- RRR (precipitation).

This suggests that regardless of the value of those features, there was no impact on the odds of a zero tip.

9.2.3 Insights from Random Forests

Finally, the feature importance approach is also applied to the random forests model, averaging across the trees.

The top five most important features appear as the colour of the cab, dropoffs and pickups in Manhattan and the pickup/dropoff x cooordinates. The features are listed in rank order of highest to lowest feature importance. It is apparent that almost all the spatial features rank higher than almost all the temporal features. The main exception seems to be the time of day feature (temporal_timeInMinutes).

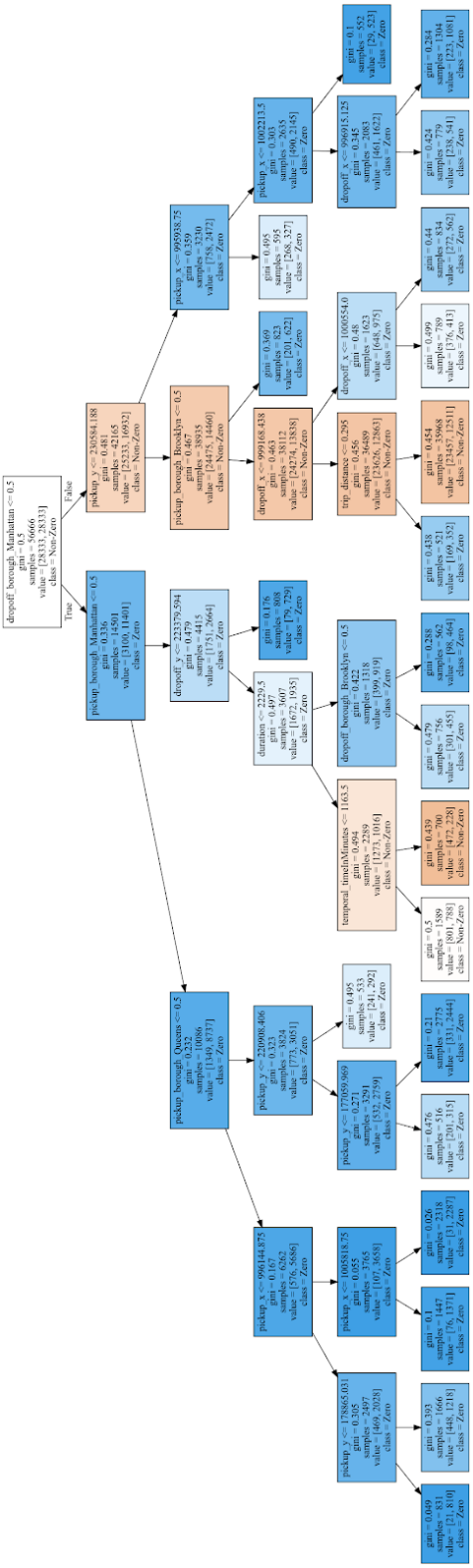

Figure 29: Decision Tree

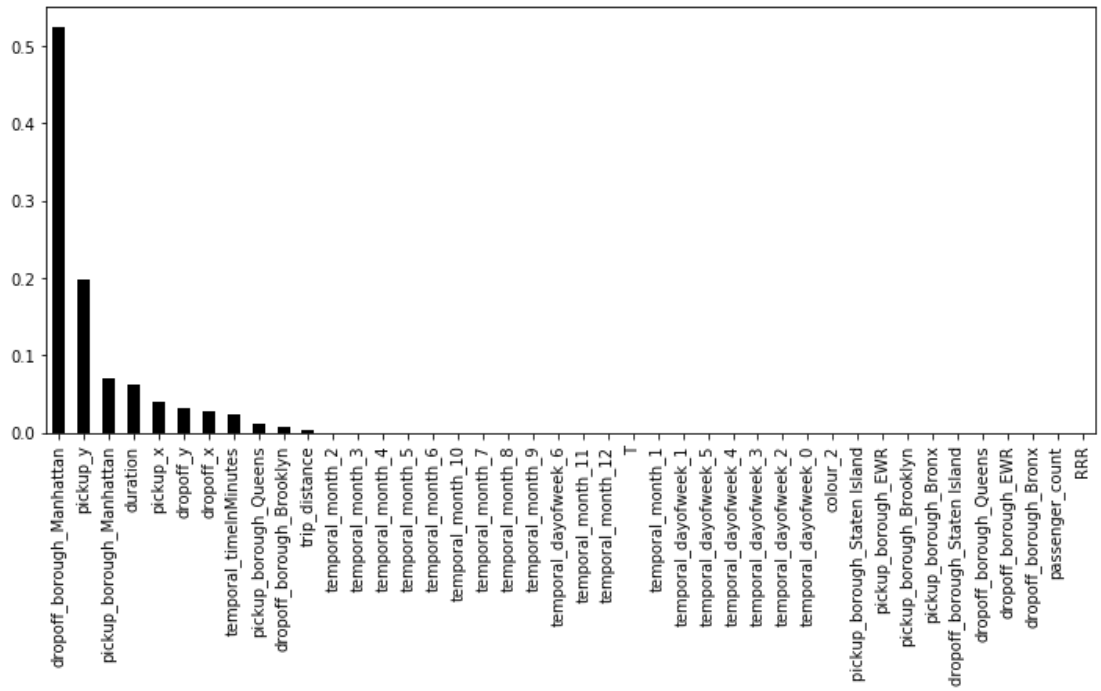

Figure 30: Decision Tree feature importances

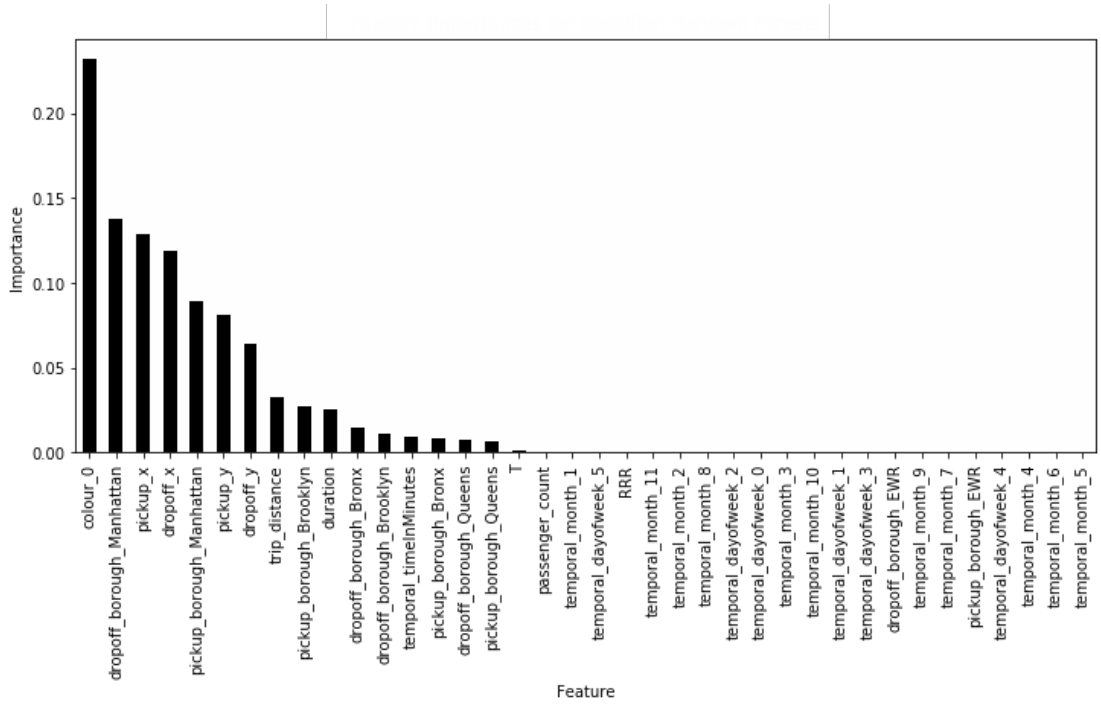

Figure 31: Random Forests classifier feature importances

10 Conclusion

All research questions have been answered. The breakdown is as follows:

- **Can we accurately predict what tip will be received on a given journey?** Three different regression models are trained and evaluated to address this. The highest performing model is random forests reported in Section 8.2.3 which has an r^2 score of 12.5%. This suggests that some small, but non-zero proportion of the variance in tips is being successfully explained by our model and its features. We can effectively make predictions which would be useful to taxi drivers, but these predictions would still be subject to significant noise. In the short term, for a driver using these predictions to optimise their driving, it is likely they would not perceive these predictions as sufficiently accurate. The accuracy of any individual prediction is simply too low. However, over longer periods of time, with greater numbers of trips, the variance of these predictions would decrease at an aggregate level allowing the taxi driver to consistently increase their tip income. Similarly, if being used by, for example, a tax authority to estimate tip income, the predictions would not be sufficiently accurate at the level of an individual trip. However, at an aggregate level, across a year's worth of trips, it could provide useful evidence in assessing the veracity of an income declaration.

- **Can we accurately predict when no tip will be left?** Six different classifiers are trained to address this research question. Table 4 shows that the highest performing by all measures is XGBoost, a tree-based ensemble classifier. Recalling 56% of zero-tip trips, the classifier could be used by a driver who was extremely averse to avoiding trips where no tip is left by using this predictions of this classifier albeit at low precision in absolute terms. Translated in to practice, most trips with no tip could be avoided, but most trips that were avoided would actually have received a tip. Just as in the regression, the trained model is successfully discriminating between different tipping behaviours, but there is significant noise in the data which cannot be eliminated.

  For both of the first two research questions, it is likely that most of the features we would need for a better performing model are simply unavailable to us. Tipping is social interaction between two individuals, the driver and the passengers. Beyond the number of passengers we have no data on either. Even the features we do have may well be acting as proxies for those we do not. Pickups in Manhattan may be biased to passengers who are wealthy or travelling for business. Pickups on the weekend may be biased to tourists, whilst those in the late hours may be in high spirits. This is why our models have uncovered consistent patterns, but we find that most of the variance is still unexplained.

- **What are the biggest determinants of tipping behaviours in New York City?** The decision tree in Figure 29 shows that generally the spatial features are selected by the individual decision tree for classification. This tells us they had

the biggest impact on gini impurity when used to split the training data. Feature importances calculated for the decision tree and also random forests, shown in figures 30 and 31 confirm this view. Random forests also tells us that the colour of the cab is important, something we saw in Figure 19: that yellow cabs earn higher average tips and receive a zero-tip less frequently. The logistic regression weights in Table 5 and linear regression weights in Table 3 show a similar pattern with the dummy spatial features and cab colour having the largest corresponding weights. The random forest regression model, which was the most successful of the regression models builds a slightly different picture. Although the spatial features, particularly the coordinates rank highly, it also includes duration and distance within its top 5 more important features. Most of the analysis tells us that spatial features are the most important when it comes to tips. Even cab colour is, in fact, a proxy for a spatial feature. We note that the key difference between green and yellow cabs is a spatial restriction on where they are allowed to operate. Yet as discussed, these spatial features may themselves be proxies for the populations who live and work in these areas. Helpfully, this insight is one of the most actionable, at least for yellow cab drivers. They can go to the boroughs and specific areas that are predicted to have the highest tips. Had we found that the most important feature(s) were the month in which the journey took place, it would be much more difficult for a driver to use this information usefully. The conclusion is in line with that of Nelson [23]: trips that begin or end in Manhattan are associated with some of the most generous tips. However, we also find that Queens is also associated with highly generous tips. This may reflect that Queens is home to both JFK and LaGuardia Airport.

- **Does the weather impact tipping behaviours in New York City?** Weather features were included in the model but rarely emerged as important features. Figure 23 suggested there may be a small but significant relationship. Both the linear regression weights in Table 3 and logistic regression weights in Table 5 for the weather features were not found to differ significantly from zero. In the feature importances for the decision tree in Figure 30 they both ranked in the bottom half of features, with precipitation ranking last. In the random forest feature importances in Figure 31 they feature higher but still much lower than any of the spatial features. Taken together, we interpret that precipitation has no impact on tipping behaviours. Temperature may have an impact, but the magnitude of the impact is extremely small and not something which might change the action of a driver who had that information. This is in line with the finding of Devaraj and Patel [6] who find the effect of sunlight is statistically significant but very small.

10.1 Future Work

This project has undertaken a broad survey of supervised learning techniques. This has left little opportunity to investigate the specific parameterisation of each model. For each model there is a set of parameters that could be changed to potentially improve

performance. The neural networks alone could be varied in structure, layers, number of nodes, activation function and loss function. The decision trees can be varied in the criteria used, the maximum depth of the tree, the minimum number of examples in each node and the size of the ensemble. Although beyond the capacity of this project, it would be useful to choose some of the more successful models and focus on a search to optimise these hyperparameters. The abundance of data available means this could be done across a large space of possible parameters, using a large validation set to test each combination.

Our view is that this project reached limits in its predictive power not because of limitations in the sophistication of the methods or the computing resources available, but because of a lack of relevant features in the data. Future work in this area might focus on sourcing these so they could be included. Most obviously, this could be information about the individuals involved in the tipping transaction: the driver and the passenger.

Another improvement would be to take a wider, more-balanced view of what the driver is trying to achieve combining other metrics such as utilisation, driving conditions or overall earnings including fares. This could be achieved through an agent based simulation which was tuned to reflect the insights this model has produced. Such a model could then be run with different proposed driver policies, the impact of which could be tested.

A passive reinforcement learning model could also be developed in which drivers were modelled as agents that could make decisions based on their current state. Such actions might be to stop working, start working, or move to a different area. The benefit of such an approach is it is more driver-centric. The models in this project are all computed at the level of the trip, not the driver. A passive reinforcement learning model would be able to maximise agent performance rather than just tips. Instead of simply greedily attempting to maximise tips, drivers may choose to take a short term hit in order to optimise long term utility.

10.2 Lessons Learned

- Data cleaning is one of the biggest challenges for a project using datasets like this one and can be easily underestimated. When planning, significant time and resource should be set aside to achieve this.

- Evaluating the performance of classifiers on highly unbalanced data requires a careful choice of metrics. It is possible to present what appears as a highly successful model which is in fact just reflecting an underlying class imbalance. Accuracy is not a useful metric in this situation.

- Understanding the general nature of the dataset is required prior to model selection. Decision tree based models have performed the best on this dataset because it is hierarchical in nature. Knowing the nature of the data set and the techniques likely to perform well would have focused our approach in advance.

References

[1] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[2] A. Liu, "Uber's remarkable growth could end the era of poorly paid cab drivers," 2014 (accessed August 20, 2019). [Online] Available `https://www.washingtonpost.com/news/innovations/wp/2014/05/27/ubers-remarkable-growth-could-end-the-era-of-poorly-paid-cab-drivers/?noredirect=on`.

[3] M. Hilbert, "Big data for development: A review of promises and challenges," *Development Policy Review*, vol. 34, no. 1, pp. 135–174, 2016.

[4] C. Camerer, L. Babcock, G. Loewenstein, and R. Thaler, "Labor Supply of New York City Cabdrivers: One Day at a Time," *The Quarterly Journal of Economics*, vol. 112, pp. 407–441, May 1997.

[5] H. S. Farber, "Why you can't find a taxi in the rain and other labor supply lessons from cab drivers," Working Paper 20604, National Bureau of Economic Research, October 2014.

[6] S. Devaraj and P. C. Patel, "Taxicab tipping and sunlight," *PLOS ONE*, vol. 12, pp. 1–16, 06 2017.

[7] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ, USA: Prentice Hall Press, 3rd ed., 2009.

[8] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks, 1984.

[9] D. J. Hand and K. Yu, "Idiot's bayes: Not so stupid after all?," *International Statistical Review / Revue Internationale de Statistique*, vol. 69, no. 3, pp. 385–398, 2001.

[10] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[11] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81–106, Mar 1986.

[12] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[13] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, "Top 10 algorithms in data mining," *Knowl. Inf. Syst.*, vol. 14, pp. 1–37, Dec. 2007.

[14] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, Oct 2001.

[15] Tin Kam Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 832–844, Aug 1998.

[16] E. Scornet, G. Biau, and J.-P. Vert, "Consistency of random forests," *Ann. Statist.*, vol. 43, pp. 1716–1741, 08 2015.

[17] G. Biau, "Analysis of a random forests model," *J. Mach. Learn. Res.*, vol. 13, pp. 1063–1095, Apr. 2012.

[18] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, (New York, NY, USA), pp. 785–794, ACM, 2016.

[19] T. Savage and H. T. Vo, "Yellow cabs as red corpuscles," *SSRN Electronic Journal*, 01 2013.

[20] N. Ferreira, J. Poco, H. T Vo, J. Freire, and C. Silva, "Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips," *IEEE transactions on visualization and computer graphics*, vol. 19, pp. 2149–58, 12 2013.

[21] M. Lu, Z. Wang, and X. Yuan, "Trajrank: Exploring travel behaviour on a route by trajectory ranking," in *2015 IEEE Pacific Visualization Symposium*, pp. 311–318, April 2015.

[22] Omnisci, "Data visualization example / demo: Nyc taxi ride data," 2019 (accessed August 15, 2019). [Online] Available `https://www.omnisci.com/demos/taxis/`.

[23] Nelson, "Taxi cab terrain. millions of cab rides over one year paint a portrait of new york city," 2017 (accessed August 15, 2019). [Online] Available `https://nation.maps.arcgis.com/apps/Cascade/index.html?appid=6984ffb035ed40b8b11e23f41236aac2`.

[24] C. Miles, "Visualizing nyc taxi cab data," 2019 (accessed August 15, 2019). [Online] Available `https://cambridge-intelligence.com/visualizing-nyc-taxi-cab-data/`.

[25] J. A. Deri, F. Franchetti, and J. M. F. Moura, "Big data computation of taxi movement in new york city," *2016 IEEE International Conference on Big Data (Big Data)*, pp. 2616–2625, 2016.

[26] D. A. Finer, "What Insights Do Taxi Rides Offer into Federal Reserve Leakage?," *SSRN Electronic Journal*, March 2018. [Online] Available `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3134953`.

[27] W. Koehrsen, "Another machine learning walk-through and a challenge," 2018 (accessed August 15, 2019). [Online] Available https://towardsdatascience.com/another-machine-learning-walk-through-and-a-challenge-8fae1e187a64.

[28] S. Li, "How taxis arrive at fares? predicting new york city yellow cab fares," 2018 (accessed July 26, 2019). [Online] Available https://towardsdatascience.com/how-taxis-arrive-at-fares-predicting-new-york-city-yellow-cab-fares-71a8c43b7c50.

[29] S. Jain, A. See, and A. Shandilya, "Predicting taxi tip-rates in nyc," 2015 (accessed August 15, 2019). [Online] Available https://pdfs.semanticscholar.org/a5bb/97457cd9f86c74528156236c1d43e7d13242.pdf.

[30] Y. Ge, H. Xiong, A. Tuzhilin, K. Xiao, M. Gruteser, and M. Pazzani, "An energy-efficient mobile recommender system," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, (New York, NY, USA), pp. 899–908, ACM, 2010.

[31] C. Peng, X. Jin, K.-C. Wong, M. Shi, and P. Liò, "Collective Human Mobility Pattern from Taxi Trips in Urban Area," *PLoS ONE*, vol. 7, p. e34487, Apr. 2012.

[32] N. Y. C. Taxi and L. Commission, "Tlc trip record data," 2019 (accessed August 15, 2019). [Online] Available https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page.

[33] N. Y. C. Taxi and L. Commission, "Tlc trip records user guide," 2019 (accessed July 26, 2019). [Online] Available https://www1.nyc.gov/assets/tlc/downloads/pdf/trip_record_user_guide.pdf.

[34] N. Y. C. Taxi and L. Commission, "Taxi zones shapefile," 2019 (accessed August 15, 2019). [Online] Available https://s3.amazonaws.com/nyc-tlc/misc/taxi_zones.zip.

[35] R. P. L. on behalf of National Oceanic and A. Administration, "Weather archive in new york / la guardia (airport)," 2019 (accessed August 15, 2019). [Online] Available https://rp5.ru/Weather_archive_in_New_York,_La_Guardia_(airport).

[36] D. M. Hawkins, *Identification of outliers*. London ; New York : Chapman and Hall, 1980.

[37] J. W. Osborne, *Best practices in data cleaning : a complete guide to everything you need to do before and after collecting your data*. Thousand Oaks : Sage, 2013.

[38] C. of New York, "How many passengers are allowed in a taxi? — city of new york," 2019 (accessed August 15, 2019). [Online] Available https://www1.nyc.gov/nyc-resources/faq/484/how-many-passengers-are-allowed-in-a-taxi.

[39] epsg.io, "Nad83 / new york long island (ftus)," 2019 (accessed August 22, 2019). [Online] Available https://epsg.io/2263.

[40] Time and D. A. Norway, "Timeanddate.com weather archive," 2018 (accessed August 15, 2019). [Online] Available https://www.timeanddate.com/weather/usa/new-york/historic?month=1&year=2018.

[41] A. Akyüz, K. Shein, and M. Asmus, "Procedure for assigning a value for trace precipitation data without changing the climatic history," *Journal of Service Climatology*, vol. 6, no. 1, 2013.

[42] "Numpy," 2019 (accessed August 22, 2019). [Online] Available https://www.numpy.org/.

[43] "pandas," 2019 (accessed August 22, 2019). [Online] Available https://pandas.pydata.org/.

[44] "Geopandas," 2019 (accessed August 22, 2019). [Online] Available http://geopandas.org/.

[45] "geopy," 2019 (accessed August 22, 2019). [Online] Available https://pypi.org/project/geopy/.

[46] "scikit-learn," 2019 (accessed August 22, 2019). [Online] Available https://scikit-learn.org.

[47] "Statsmodels," 2019 (accessed August 22, 2019). [Online] Available https://www.statsmodels.org/stable/index.html.

[48] "Keras documentation," 2019 (accessed August 22, 2019). [Online] Available https://keras.io/.

[49] "Matplotlib," 2019 (accessed August 22, 2019). [Online] Available https://matplotlib.org/.

[50] A. Hern, "New york taxi details can be extracted from anonymised data, researchers say," 2014 (accessed August 15, 2019). [Online] Available https://www.theguardian.com/technology/2014/jun/27/new-york-taxi-details-anonymised-data-researchers-warn.