Nile University
جامعة النيل

Automate Parking Appeal Process using historical data

Project Final Report

Author
Hager Mohamed Magdy

1 Table of Contents

1. Introduction

Time taken by the process of appeals is major problem for many drivers, many drivers are interesting in quick response for their appeals in order to take the right action for their cars if its towed or clamped.

Observing penalty charge notices issued by civil enforcement officers on street and those issued by civil enforcement officers via CCTV can facilitate to solve this problem. Consequently,

the driver will be able to receive respond to his appeal early accordingly he could plan for further action. In addition, the city could predict the reason why appeals taken in certain locations are always valid, so could take corrective action and reduce the agent workload.

This Report summarize the work done and the result of using machine learning and data science techniques to acquire, clean and analyze PCN historical data to automate the process of routine appeals.

2. Approach

By Exploring the PCN values over time , machine learning algorithm can learn the relationship between the change values of PCN data e.g. (street, Civil Enforcement Officer Error, Ticket Issued Via CCTV Camera , Spatial Accuracy, Cancellation Reason and location) and the change in the status of the case (paid /cancelled), it is assumed that the status of the case reflects appeal status .

As a result, the model can learn if the appeal is valid or invalid depending on the relation between the change in the appeal status values and the change in the status of the case to the historical Acceptance or rejection of the appeal in order to predict the status of the appeal in the future.

Classification modeling algorithms will be used to predict if the appeal is accepted or not.

Binary Classification algorithms is used to predict if the appeal is accepted or not:

- Logistic Regression
- KNN
- Random forest
- DecisionTreeClassifier

3. Development Environment

- Jupyter Notebook running on python 3.
- Pandas and numbly for data wrangling and analysis.
- Matplotlib and seaborn for visualization.
- Scikit learn for machine learning algorithm.

Project Repo on GitHub :

https://github.com/HagerMohamedMagdy/Data-Science-Project

4. Data

4.1.Data Acquisition

The dataset contains transactional penalty charge notice data held in the London Borough of Camden's parking management system, inclusive of penalty charge notices issued by Civil Enforcement Officers on street and those issued by Civil Enforcement Officers via CCTV. Attribution includes contravention code, ticket type, street, parking restriction, vehicle category and status of case. Where possible, the approximate location of the PCN has been captured.

Dataset Columns	
Contravention Date	Foreign Vehicle
Contravention In Last 7 Days	Country Vehicle Registered To
Ticket Type	Has Appeal
Ticket Description	Formal Representation
Contravention Code	Ward Code
Contravention Code Suffix	Ward Name
Contravention Code Description	Easting
Ticket Issued Via CCTV Camera	Northing
Controlled Parking Zone Area	Longitude
Street	Latitude
Vehicle Category	Location
Vehicle Removed	Spatial Accuracy
Status Of Case	Last Uploaded
Charging Band Description	Socrata ID
Civil Enforcement Officer Error	PenaltyChargeNotice Cancelled
	Cancellation Reason

Note:

Civil Enforcement Officer GPS Location = the GPS location reported by the CEO's handheld device Fixed CCTV Camera = the location of the CCTV camera, this could be many meters from the vehicle in contravention Unknown = No location has been captured

Data URL:
https://data.gov.uk/dataset/248a9d69-f9cb-420f-98b7-e03121fee3bd/parking-services-penalty-charge-notices-2016-17?fbclid=IwAR12FXNbHIbbiRWEq4hm-iRUfrrtdeET-GswG6PSLRauqA-H0oi9aZqfWeg

4.2. Data Wrangling

- Most Features are of type object except numeric features are float.
- Some features contain Null Values.
- Scorta ID has all values unique so it could be used as ticket identifier.
- define the output class ('Status of the case')
- data is split between the different output classes (Paid/Closed, cancelled, written off/outstanding).

4.2.1. Prepare Data for Molding

- Null value in cancellation reason is replaced with 0. cancel-->1 not cancelled-->0
- Feature Extraction is also applied to the data set by introducing three new columns:
- (month number of the year/ day of the week /hour of the day).
- Labels for data were created as follow:
- If "Status of the case is Paid/Close then label =1 else label =0.
- Scale the latitude, longitude features.
- Implement one hot encoding on the categorical columns.

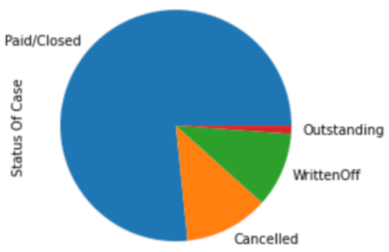
5.Explanatory Data Analysis

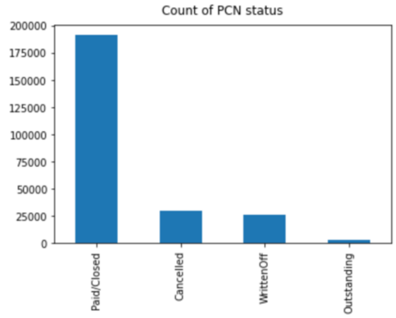
Feature variability and distribution were examined to uncover underlying structure and extract important features:

- Get some statics on numerical data.
- Frequency of occurrence of each ticket type by status of the case .
- Check if the cancelled tickets must have appeal or not.
- Check the no of the ticket where the status of the case is Paid/closed
- Find number of PCN taken in each street.
- Get max 10 street with max ticket occurrence.
- Plot streets with maximum no of tickets addressed.

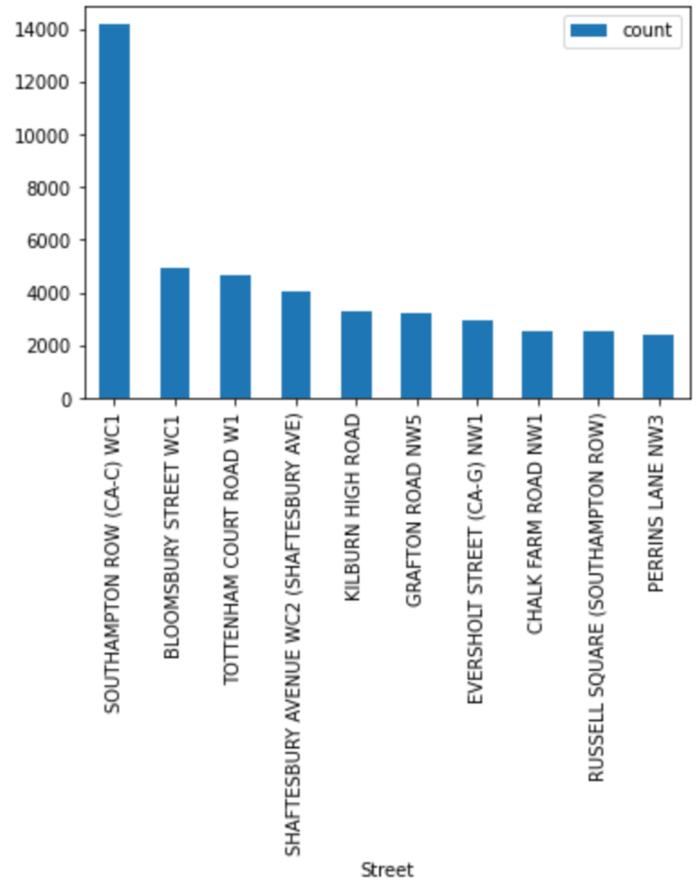
A number of EDA charts were also used to have more insight of each feature:

- understand how data is split between the different output classes

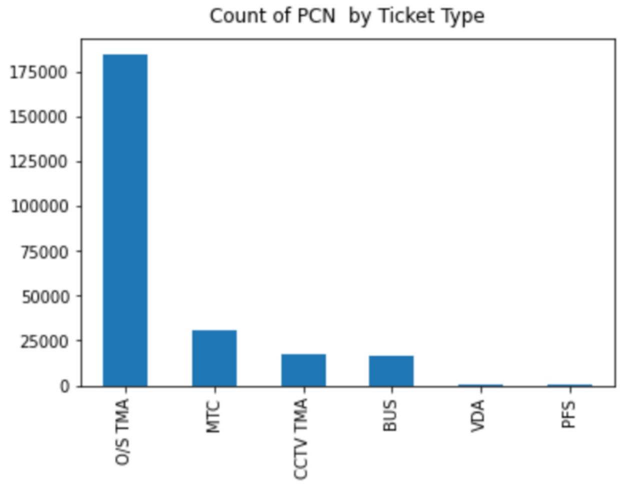




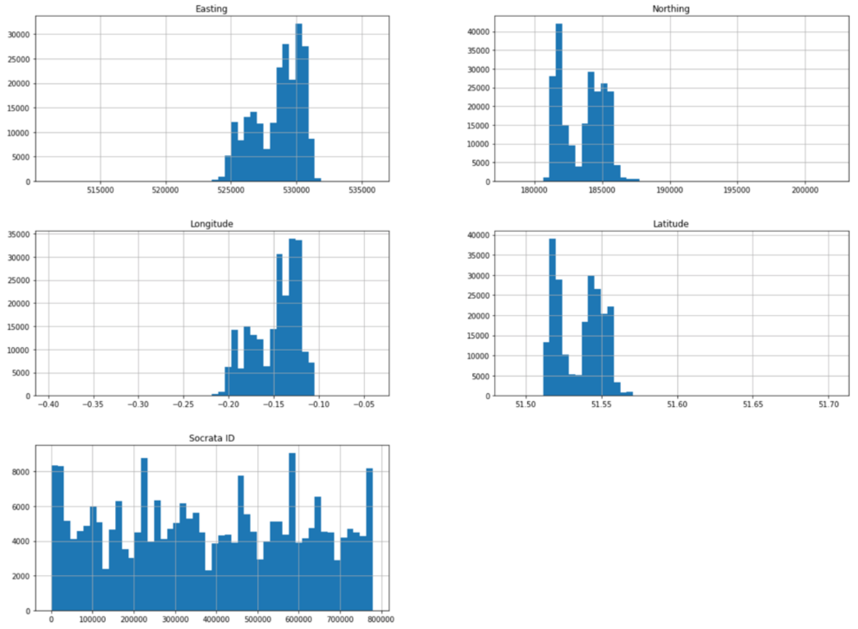
- Get max 10 street with max ticket occurrence.



- Frequency of occurrence of each ticket type



- Numerical data distribution



6.Predicting the status of the appeal

- The following machine learning algorithms were tried and their performance metrics were calculated and evaluated: Logistic Regression, KNN, Random Forest, Decision Tree Classifier, A short definition of these algorithms:

1. **Logistic Regression** is the classification counterpart to linear regression. Predictions are mapped to be between 0 and 1 through the logistic function, which means that predictions can be interpreted as class probabilities.

2. **KNN** are instance-based algorithms, which mean that they save each training observation. They then make predictions for new observations by searching for the most similar training observations and pooling their values.

3. **Decision Trees** learn in a hierarchical fashion by repeatedly splitting the dataset into separate branches that maximize the information gain of each split. In regression tree, the value obtained by terminal nodes in the training data is the mean response of observation falling in that region, whereas in classification tree, the value (class) obtained by terminal node in the training data is the mode of observations falling in that region.

4. **Random Forests** are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

Main binary classification metrics were calculated Accuracy, Precision, F1-Score, and Recall

The results on test dataset listed below:

	Logistic Regression	KNN	Random forest	DecisionTreeClassifier
accuracy	0.4992080653380528	0.7379316043907612	0.9750161830126571	0.9301720219813516
precision	0.4992080653380528	0.6558415698200645	0.9523859575362623	0.9589024964672633
recall	1.0	0.99955857198035	0.9999448214975446	0.8986370909893505
f1	0.6659623528979449	0.7920164393144457	0.9755861215041318	0.9277921781980801

7.Classification Summary

- Random Forests scored best .
- Random Forest scored better than other classifiers in Recall (Sensitivity) while DecisionTreeClassifier scored better in Precision

8.Expected Profit

Cost-benefit matrix should be provided by Product manager .In our case we have the below business metrics :

- If the system approves an invalid appeal, the city loses revenue from the parking ticket. The matrix is the number of invalid appeals approved .
- If the system passes a valid appeal to the agent, the city sustain the cost of the agent's salary . So the matrix is the number of valid appeals examined by an agent..

we can translate them to the following:

- **False positive rate:** The percentage of invalid appeals approved by the system.
- **False negative rate**: the percentage of valid appeals handled by an agent.

9.Business goals

1. Maximize profit

- If the city wants to maximize profit from PCN. Then the model should balance the false positive and false negative rates (select between the the lost revenue for a false approval against the cost of handling an appeal).
- If the hourly rate of an agent is low, the model should approve fewer cases. But if the hourly rate is high, the model should approve more cases.
- The modeling metric to optimize will end up being a type of weighted average between false positive and false negative rates.

2. Maximize profit [without approve many false appeals]

- if the system approves falsely too many appeals, drivers will park illegally more often.
- The matrix to optimize is still an average of the false positive and false negative rates. But there will be a limit on the false positive rate.

3. Reduce response time when appeals' number is too high

- the government may use the system to reduce the turnaround time when the number of appeals is very high.

- In this case, we should know how many cases we want the agents to handle. The false negative rate is now not an important matrix. but the model should reduce false approvals relative to the volume of cases handled by agents as a given input.

10.Summary

The Project tries to solve an essential problem regarding the parking appeal routine process.
By applying machine Logistic Regression, KNN, Random Forest, Decision Tree Classifier, to historical data of PCN, the project was able to automate the parking appeal routine process by implementing a system to detect if the appeal is valid or invalid depending on the given historical data. Consequently, both the driver and the government take further corrective actions.

11.Next Steps

- Deploy selected models to be accessible online.
- The model can be translated to mobile application for the drivers. [Future work]