PLANNING A SUCCESSFUL ELECTRIC VEHICLE RELEASE

Issei Kuzuki

17/10/2022

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusions
- Appendix

Executive Summary

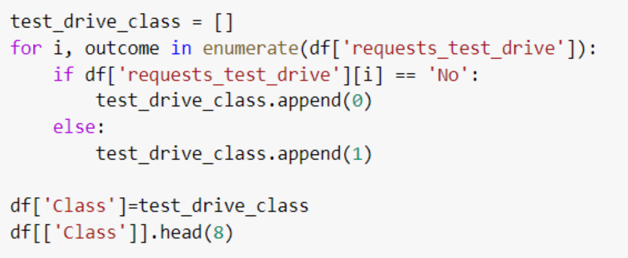
- Performed exploratory data analysis on the survey data to find what features from the survey data may be related to a user requesting a test drive for the new Electric Vehicle
- Built a machine learning model using the Python library Scikit learn to see if it may be possible to predict whether someone would want to request a test drive based on other information
- Made an interactive map using the Python library Folium to try to visualise which regions in the country had the most potential for new customers
- Compared the information gained from this Python data analysis to the Mosaic UK 7 Grand Index to find out a profile of a typical customer
- Built an algorithm using the google analytics data to sort users visiting the BMW website into groups to see whether they are worth re-targeting

Introduction

- BMW are preparing the launch of a new luxury Electric Vehicle (EV) and would like to attract potential buyers to test drive in the new EV
- The aim of this project is to use data to determine the type of person who is most likely to be interested in test driving an EV
- 3 datasets were provided: Survey data, Mosaic UK 7 Grand Index and Google analytics data for the BMW website related to the new EV

Methodology: Exploratory Data Analysis

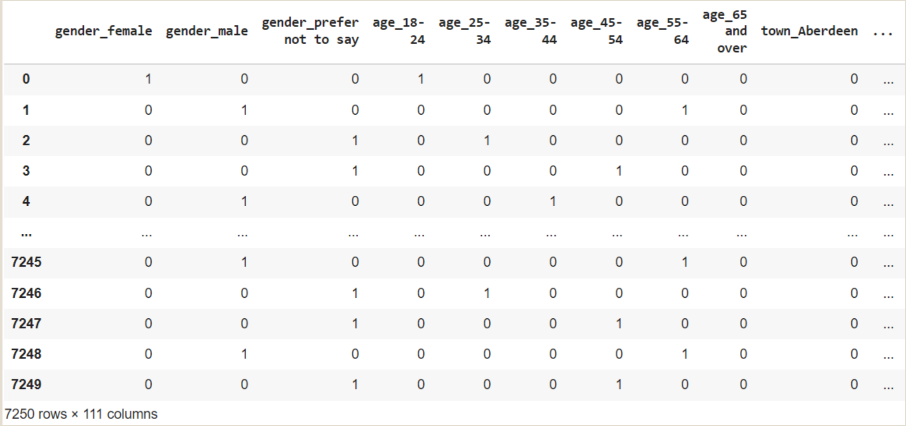
- Python was used to perform exploratory data analysis on the survey data
- The key outcome from the survey data is the ‘requests a test drive’ column, this is the variable we want to maximize
- The ‘requests a test drive’ column was converted to a binary variable called ‘class’ which outputs a 0 if a test drive was not requested and a 1 if one was requested
- The other survey results were plotted against this ‘class’ variable to see what may be indicators that a person will request a test drive



```python
test_drive_class = []
for i, outcome in enumerate(df['requests_test_drive']):
    if df['requests_test_drive'][i] == 'No':
        test_drive_class.append(0)
    else:
        test_drive_class.append(1)

df['Class']=test_drive_class
df[['Class']].head(8)
```

Methodology: Predictive Analysis

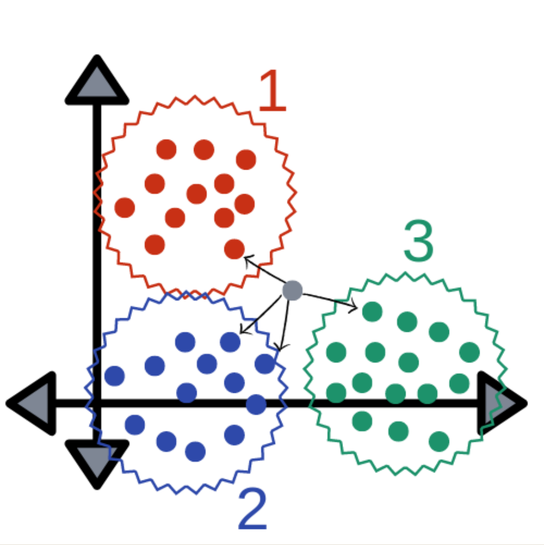
- The Python library Scikit learn has prebuilt Machine learning models which can be used for classification tasks
- The survey data was used to make a model to predict whether someone would request a test drive
- A technique called one hot encoding was used to convert all survey outputs into binary variables so that the results can be processed by the machine learning model



	gender_female	gender_male	gender_prefer not to say	age_18-24	age_25-34	age_35-44	age_45-54	age_55-64	age_65 and over	town_Aberdeen	...
0	1	0	0	1	0	0	0	0	0	0	...
1	0	1	0	0	0	0	0	1	0	0	...
2	0	0	1	0	1	0	0	0	0	0	...
3	0	0	1	0	0	0	1	0	0	0	...
4	0	1	0	0	0	1	0	0	0	0	...
...	...	...	...	...	...	...	...	...	...	...	...
7245	0	1	0	0	0	0	0	1	0	0	...
7246	0	0	1	0	1	0	0	0	0	0	...
7247	0	0	1	0	0	0	1	0	0	0	...
7248	0	1	0	0	0	0	0	1	0	0	...
7249	0	0	1	0	0	0	1	0	0	0	...

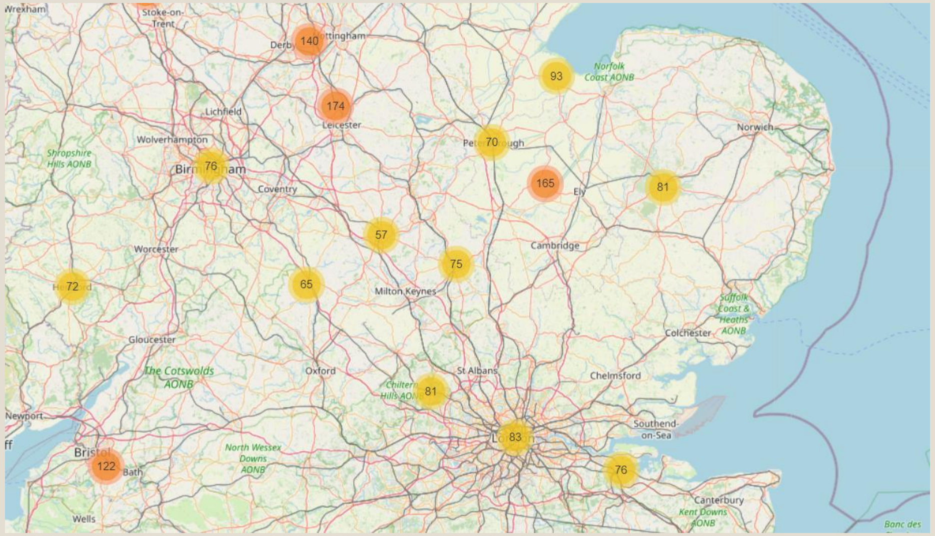
7250 rows × 111 columns

- The survey data was split into training and testing data, the training set is used to train the model and the test set is used to analyse how accurate the model is
- A classification model called K nearest neighbours (KNN) was used, this aims to sort a new data point into its class by looking at its K nearest data points in a high dimensional space
- The parameter K can be varied along with other parameters to optimize the model



Methodology: Interactive map

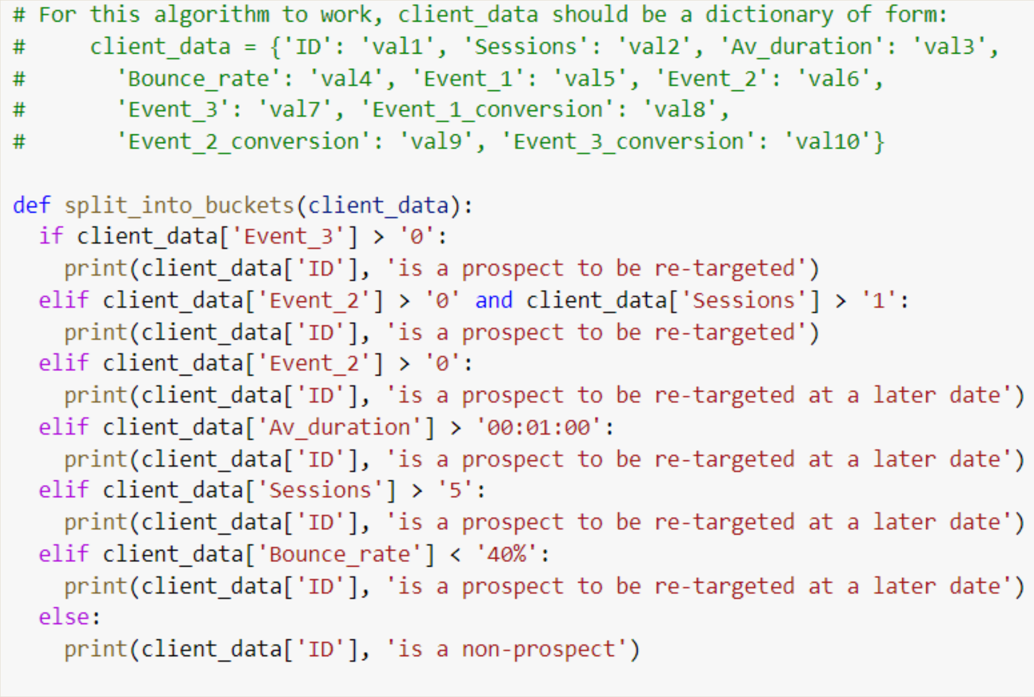
- An interactive map was built using the Python library Folium to give a visual insight into the geographic locations of survey data
- Markers were added to the map which represent the number of users which requested a test drive
- The aim is to use this geographic data along with the Mosaic UK 7 Grand Index to identify a potential buyer profile



Methodology: Analytics algorithm

- A Python algorithm was built which can be used to sort the users from the google analytics data into 3 categories: non prospects, prospects for re-target and prospects to be retargeted at a later date
- The Google analytics data contains data such as average session time, bounce rate, quantity of sessions along with the activity for 3 events
- Event 1: User configured their EV
- Event 2: User visited the test drive sign up page
- Event 3: User visited the test drive sign up page, filled in their details and requested a test drive

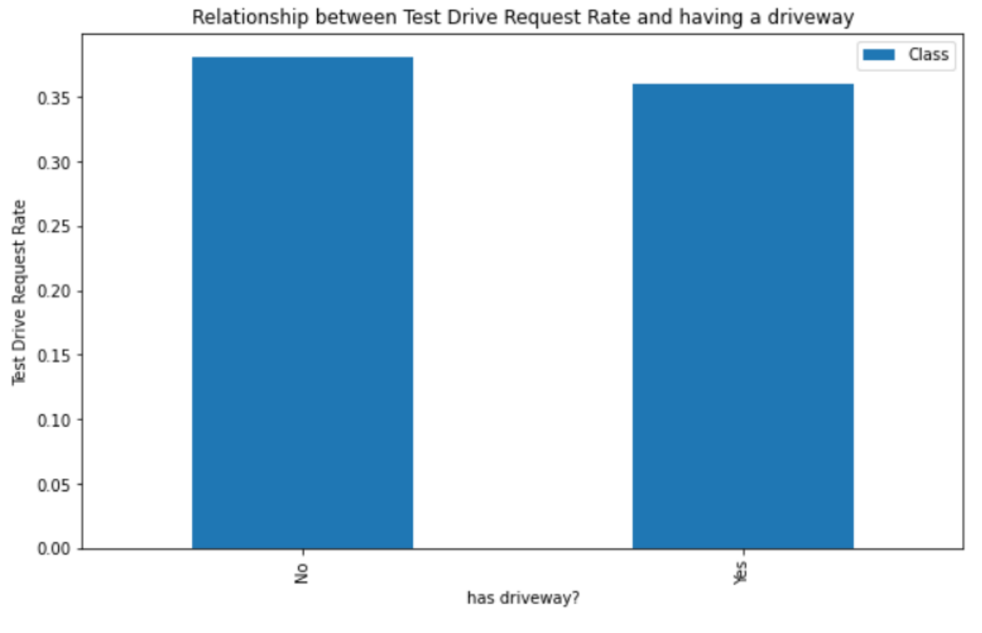
- The algorithm first aims to determine if a visitor is a high engagement user who should be re-targeted
- Completing Event 3 is an immediate indicator along with those who have completed Event 2 and have revisited the site
- Next, several thresholds were made to try and determine users which were high engagement but probably should be retargeted later
- This includes those with a high average session time, high quantity of sessions and a low bounce rate
- If none of the criteria were met, the user were deemed to be non-prospects

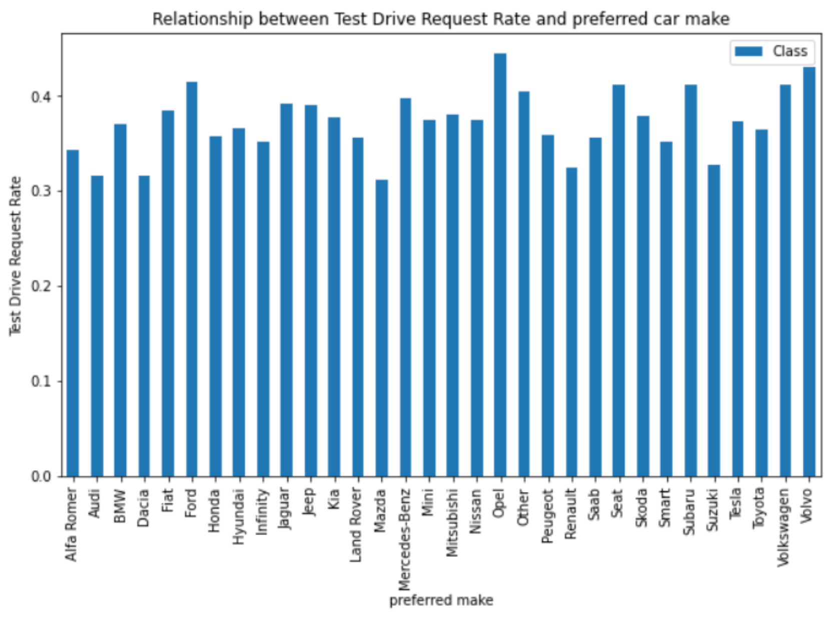


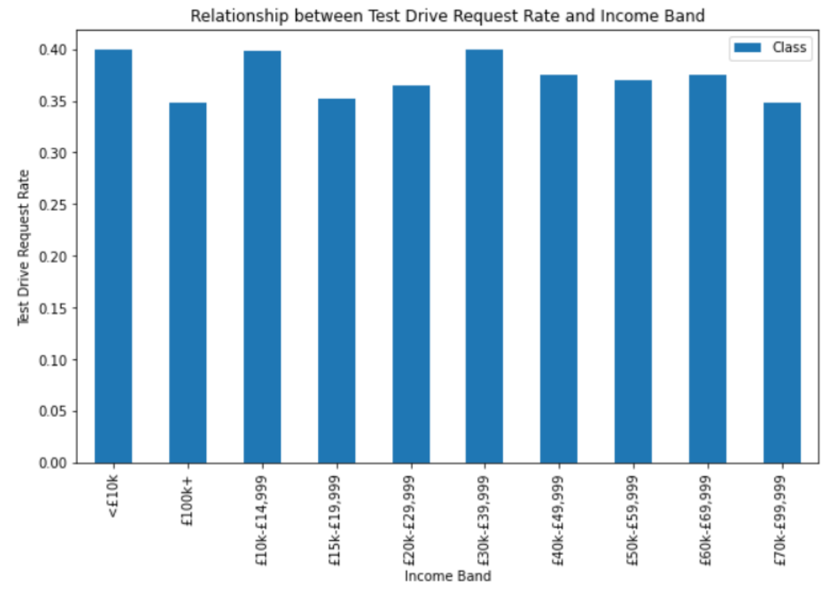
```python
# For this algorithm to work, client_data should be a dictionary of form:
#     client_data = {'ID': 'val1', 'Sessions': 'val2', 'Av_duration': 'val3',
#       'Bounce_rate': 'val4', 'Event_1': 'val5', 'Event_2': 'val6',
#       'Event_3': 'val7', 'Event_1_conversion': 'val8',
#       'Event_2_conversion': 'val9', 'Event_3_conversion': 'val10'}

def split_into_buckets(client_data):
  if client_data['Event_3'] > '0':
    print(client_data['ID'], 'is a prospect to be re-targeted')
  elif client_data['Event_2'] > '0' and client_data['Sessions'] > '1':
    print(client_data['ID'], 'is a prospect to be re-targeted')
  elif client_data['Event_2'] > '0':
    print(client_data['ID'], 'is a prospect to be re-targeted at a later date')
  elif client_data['Av_duration'] > '00:01:00':
    print(client_data['ID'], 'is a prospect to be re-targeted at a later date')
  elif client_data['Sessions'] > '5':
    print(client_data['ID'], 'is a prospect to be re-targeted at a later date')
  elif client_data['Bounce_rate'] < '40%':
    print(client_data['ID'], 'is a prospect to be re-targeted at a later date')
  else:
    print(client_data['ID'], 'is a non-prospect')
```

Results: Exploratory Data Analysis

- BMW have said that from their previous experience potential buyers were likely to live in urban areas with access to driveways and likely to be those who already own a BMW or a similar brand (e.g., Audi or Mercedes-Benz)
- From the survey data, there was a larger proportion of people that requested a test drive without a driveway than with one

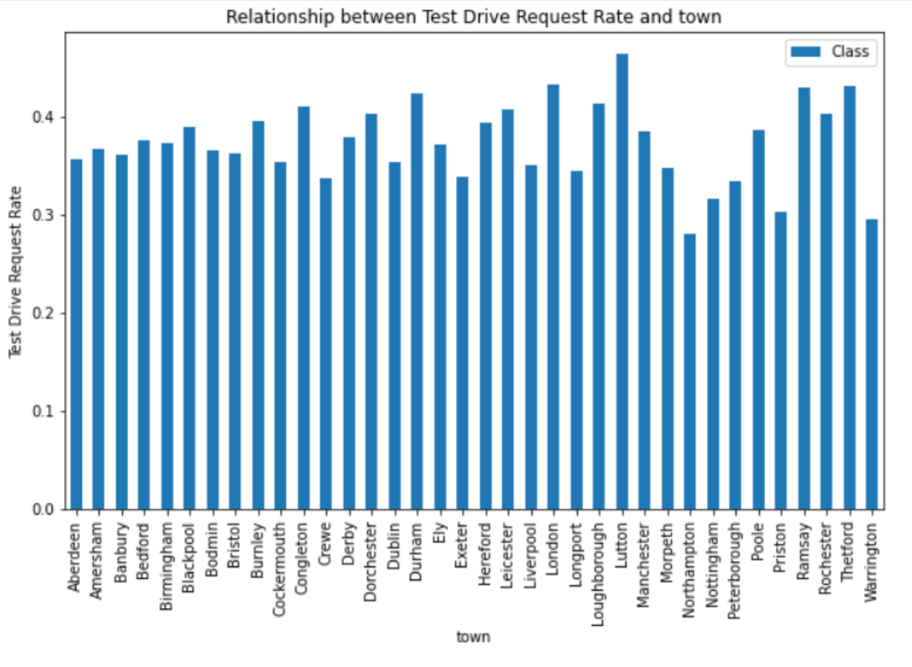




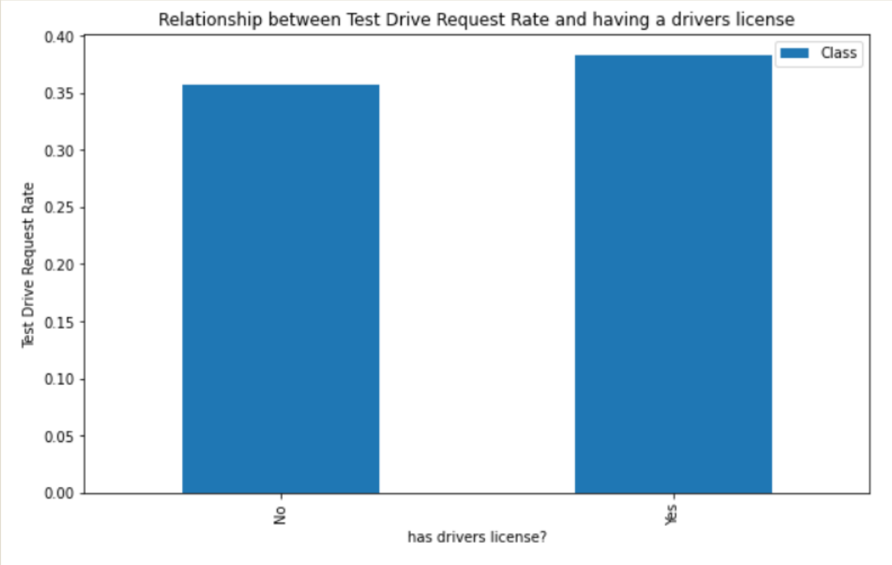


- The preferred car make of the user was significant, the biggest audiences were those who preferred Opel and Volvo cars
- This is perhaps not surprising given these brands have a large selection of EV's on the market
- Contrary to what BMW have suggested, Audi users were not very keen to test drive

- Income band proved to be a key factor in requesting a test drive, the lowest income bands had a surprisingly large test drive request rate, but it is likely they have no intention of buying



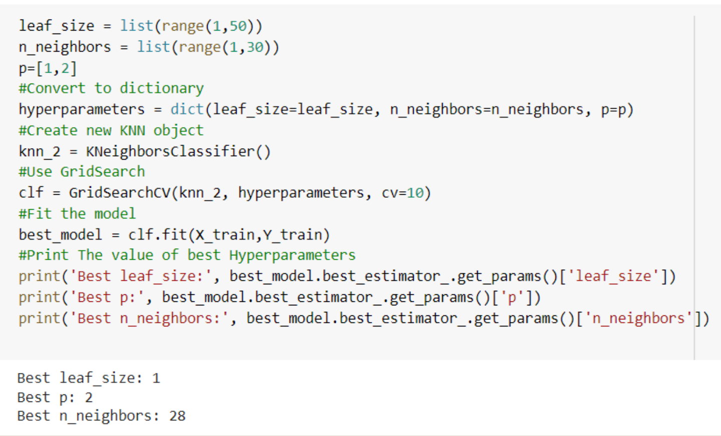
- Location showed variations in test drive request rate but perhaps not in the way expected, London has a large request rate but there were several smaller towns with high request rates too



- Having a driver's license influenced requesting a test drive positively although perhaps not as much as expected
- Many users requested a test drive despite not having a driver’s license

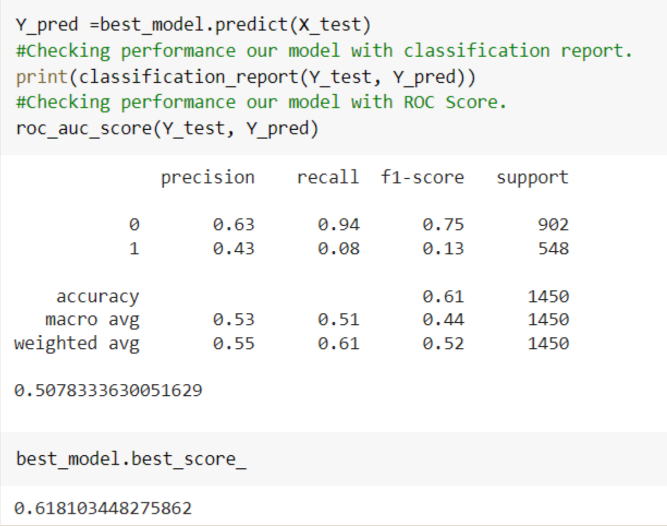
Results: Predictive Modelling

- The parameters for the KNN model was tuned to find the ‘best model’
- The model trained on the training data was tested on the testing data and compared to the actual outcomes
- A classification report was created to find how the model performed
- The model gave a 61.8% accuracy which is not amazing for a binary classification model



```
leaf_size = list(range(1,50))
n_neighbors = list(range(1,30))
p=[1,2]
#Convert to dictionary
hyperparameters = dict(leaf_size=leaf_size, n_neighbors=n_neighbors, p=p)
#Create new KNN object
knn_2 = KNeighborsClassifier()
#Use GridSearch
clf = GridSearchCV(knn_2, hyperparameters, cv=10)
#Fit the model
best_model = clf.fit(X_train,Y_train)
#Print The value of best Hyperparameters
print('Best leaf_size:', best_model.best_estimator_.get_params()['leaf_size'])
print('Best p:', best_model.best_estimator_.get_params()['p'])
print('Best n_neighbors:', best_model.best_estimator_.get_params()['n_neighbors'])
```

```
Best leaf_size: 1
Best p: 2
Best n_neighbors: 28
```



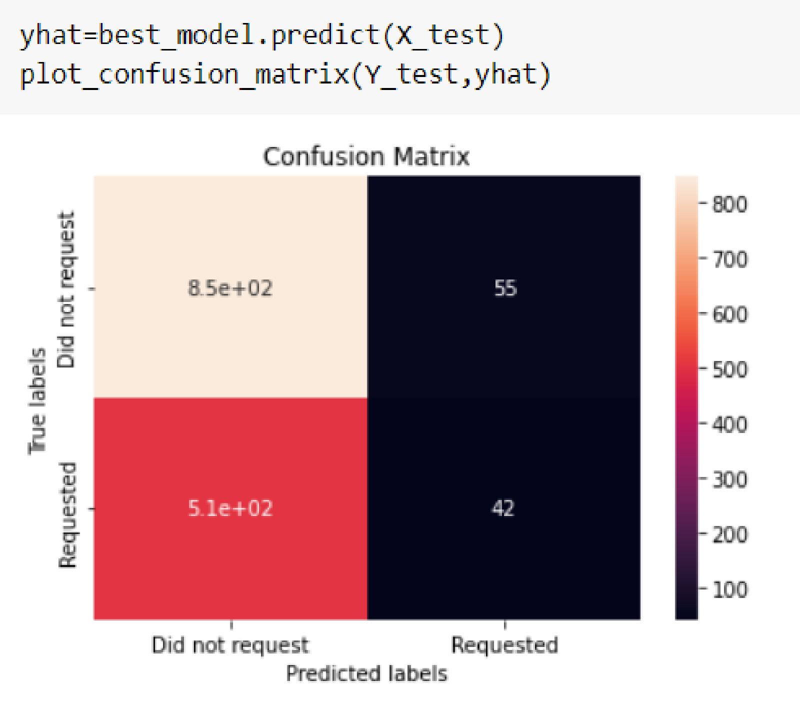
```
Y_pred =best_model.predict(X_test)
#Checking performance our model with classification report.
print(classification_report(Y_test, Y_pred))
#Checking performance our model with ROC Score.
roc_auc_score(Y_test, Y_pred)
```

```
              precision    recall  f1-score   support

           0       0.63      0.94      0.75       902
           1       0.43      0.08      0.13       548

    accuracy                           0.61      1450
   macro avg       0.53      0.51      0.44      1450
weighted avg       0.55      0.61      0.52      1450

0.5078333630051629
```

```
best_model.best_score_
```

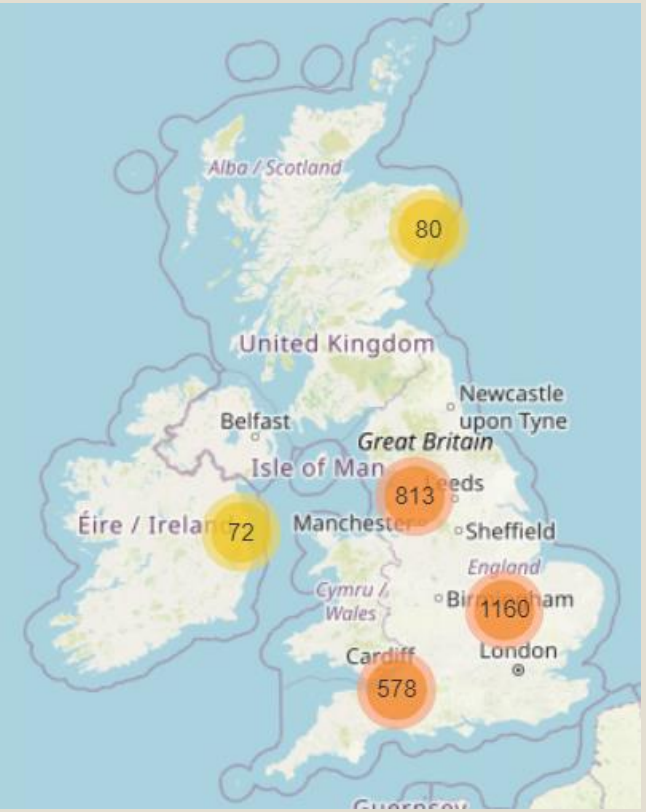
```
0.618103448275862
```

- A confusion matrix was plotted to analyse how the model classified datapoints
- The model almost always predicted that a user did not request a test drive
- As a result, the model suffered from having many false negatives
- This may be due to an overfitting problem or the inclusion of too many variables, many of which were perhaps irrelevant to determining whether a user would request a test drive

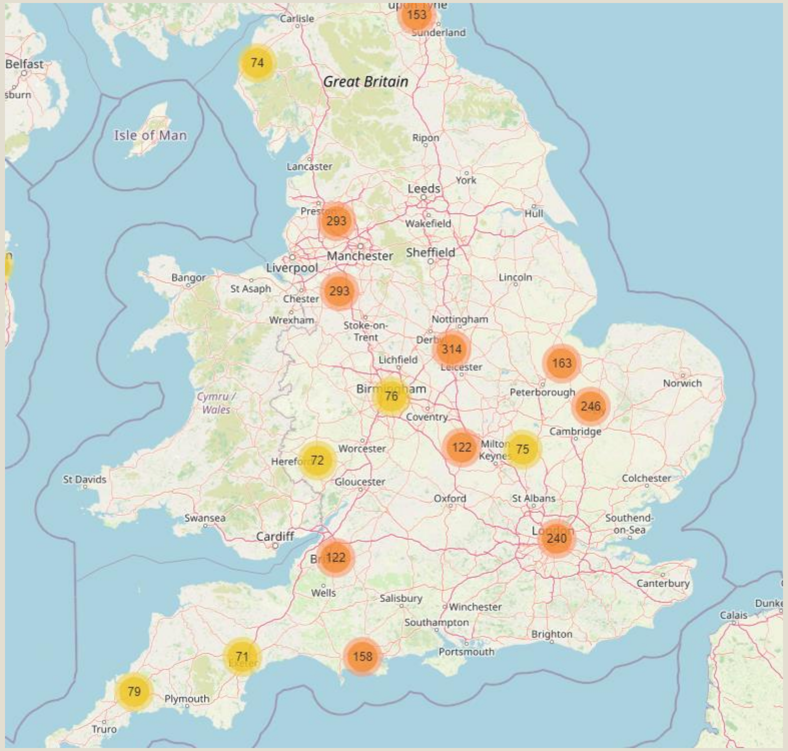


```
yhat=best_model.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```

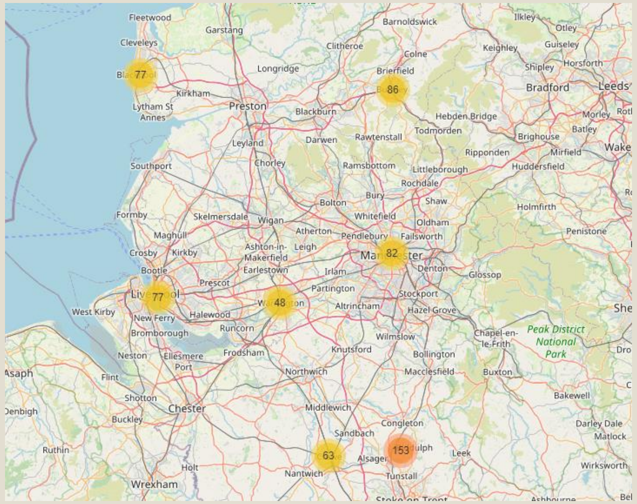
Results: Interactive map

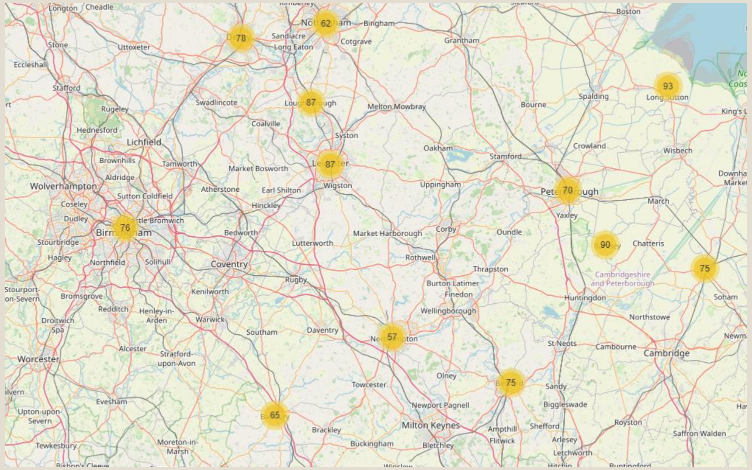


- On a UK wide scale, the largest demand for a test drive was found around London, the Southwest and the Northwest around Manchester
- The demand was low in the Northeast, Scotland and Ireland
- It was evident that the highest demand is concentrated around big cities



- With further zooming, it can be seen that the demand for a test drive is fairly evenly distributed
- Areas around big cities such as Manchester, London, Bristol and Nottingham show large numbers of potential buyers
- The Northwest, East Midlands and London show good potential in particular





Mosaic UK 7 Grand Index

- The results found from our Python analysis of the Survey Data can be compared with the Mosaic dataset to attempt to build a profile of a potential buyer
- The results from the Folium map can be compared to the data in the regional distributions section
- London shows high percentage in the categories City Prosperity (27.96%), Urban Cohesion (18.70%) and Municipal Tenants (16.01%)
- The East Midlands shows a high percentage in the aspiring homemakers (11.88%) and senior security (10.26%) categories
- The Northwest shows a big spread between the categories with family basics the largest (11.58%) but with several other categories such as senior security, aspiring homemakers in the 9-10% range

- The group City Prosperity looks to be a large audience, this group can be described as High-status city dwellers living in central locations and pursuing careers with high rewards
- Some key features of this group are urban areas, high value flats, high income
- Another group is Senior Security, this group is described as Elderly people with assets who are enjoying a comfortable retirement
- Some key features of this group are retired singles and couples, established in community and low internet use
- The Aspiring Homemakers group also looks promising with a large proportion of the East Midlands and Northwest being made up of this category
- Some key features of this group are families with young children and 3 bedrooms

Conclusions

- Contrary to what BMW suggested, having access to driveways and owning a similar car brand were not significant features of people who wanted to test drive the new EV
- However, owners of EV cars such as Volvo and Opel were much more likely to request a test drive
- Income band was somewhat negatively correlated to the request rate of a test drive which seems counter-intuitive but suggests many people with low incomes may be relying on other sources of money or are requesting a test drive with no intention to purchase the product
- Urban areas seemed to be hot zone for potential customers with areas around big cities such as Manchester, London, Bristol and Nottingham showing large numbers of potential buyers
- There were also a lot of people requesting a test drive from London, Northwest and East Midlands which can be characterized as being in the City Prosperity, Urban Cohesion, Senior Security and Aspiring Homemakers groups in the Mosaic index

Appendix

- The Python work was completed on Google Colab notebooks, the links to access the notebooks can be found below
- Exploratory Data Analysis: https://colab.research.google.com/drive/1PckWZOuhSUbvfdYrdlfjhSJYaom2rApV?usp=sharing
- Predictive Analysis: https://colab.research.google.com/drive/1X_vWn8GAi59Wm7QcBDHYMJrXwNJTr8MX?usp=sharing
- Interactive Map:
- https://colab.research.google.com/drive/1Vv5566teAO_gntE3KgOPdUc-BC66IBRu?usp=sharing
- Google Analytics Algorithm:
- https://colab.research.google.com/drive/1ElWxy3Dc-lDR4w-LfVnbYbSlPU7MKav-?usp=sharing