PROJECT MC-NULTY SUMMARY
CLASSIFICATION OF FORD GO BIKE TRAFFIC IN SAN FRANCISCO

-Hiranya Krishna Kumar

DOMAIN

In Project Mc-Nulty, I classified FordGo bike share in San Francisco into three classes by creating a target label , Flux calculated as:

0: Normal Traffic ,bike flow <0.15*dock capacity of station
1: Cautionary inflow/Surplus>0.15 *capacity
2:Cautionary outflow/Shortage of bikes>0.15*capacity

The data for bike traffic was collected from Bike Share Data for the year 2017 from July-December Station Information and status data was collected by making API calls to FordGo bike. Weather data like temperature, wind speed, summary, visibility, cloud cover for the same period was collected from Dark Sky API Dark Sky API by using python Forecast.io wrapper forecast.io

DATA

Station_Information.json

Field Name	Required	Defines
stations	Yes	Array that contains one object per station in the system as defined below
- station_id	Yes	Unique identifier of a station. See Field Definitions above for ID field requirements
- name	Yes	Public name of the station
- short_name	No	Short name or other type of identifier, as used by the data publisher
- lat	Yes	The latitude of station. The field value must be a valid WGS 84 latitude in decimal degrees format. See: http://en.wikipedia.org/wiki/World_Geodetic_System, https://en.wikipedia.org/wiki/Decimal_degrees

PROJECT MC-NULTY SUMMARY

CLASSIFICATION OF FORD GO BIKE TRAFFIC IN SAN FRANCISCO

-Hiranya Krishna Kumar

- lon	Yes	The longitude of station. The field value must be a valid WGS 84 longitude in decimal degrees format. See: http://en.wikipedia.org/wiki/World_Geodetic_System, https://en.wikipedia.org/wiki/Decimal_degrees
- address	Optional	Valid street number and name where station is located. This field is intended to be an actual address, not a free form text description (see "cross_street" below)
- cross_street	Optional	Cross street of where the station is located. This field is intended to be a descriptive field for human consumption. In cities, this would be a cross street, but could also be a description of a location in a park, etc.
- region_id	Optional	ID of the region where station is located (see system_regions.json)
- post_code	Optional	Postal code where station is located
- capacity	Optional	Number of total docking points installed at this station, both available an

Station region.json

Field Name	Required	Defines
regions	Yes	Array of region objects as defined below

PROJECT MC-NULTY SUMMARY
CLASSIFICATION OF FORD GO BIKE TRAFFIC IN SAN FRANCISCO

-Hiranya Krishna Kumar

- region_id	Yes	Unique identifier for the region
- name	Yes	Public name for this region

Trip Data

Attribute	**Type**	**Description**
duration_sec	str	
start_time	float	
start_station_id	str/binary	Heating type
start_station_name		A/C or ceiling fan
start_station_latitude	str/binary	waterfront/non waterfront
start_station_longitude		
end_station_id		
end_station_nam		

DATA CLEANING AND FEATURE ENGINEERING

A Target label was created and model was features were extracted by performing pandas merge and groupby methods.Hourly trip flux data was extracted for each station. Weather data for San Francisco was collected from Dark sky and merged on hour for each station. After data collection, I replaced all missing values with 0 for

PROJECT MC-NULTY SUMMARY
CLASSIFICATION OF FORD GO BIKE TRAFFIC IN SAN FRANCISCO

-Hiranya Krishna Kumar

categorical variables as. All categorical variables were consolidated using patsy design matrices.A heatmap showing fluctuation of bike traffic by station for every hour was plotted.

MODEL BUILDING

Since the data was highly imbalanced, greater than 95% majority class, SMOTE ENN on all classes on the train set to all classes. Stratified 3- Fold CV with grid search was used for knn, decision tree and Random Forest classifiers.Randomized CV was used for Gradient boosting, SVM with linear and radial kernels, Logistic regression.

RESULTS

The final model using Random Forest with parametermin_samples_leaf=9, n_estimators=150 ,max_depth=5. as it had highest recall on all classes for the test set . The objective is to maximize recall and minimize false negatives on surpluss and shortage of bike classes.The results are tabulated below.

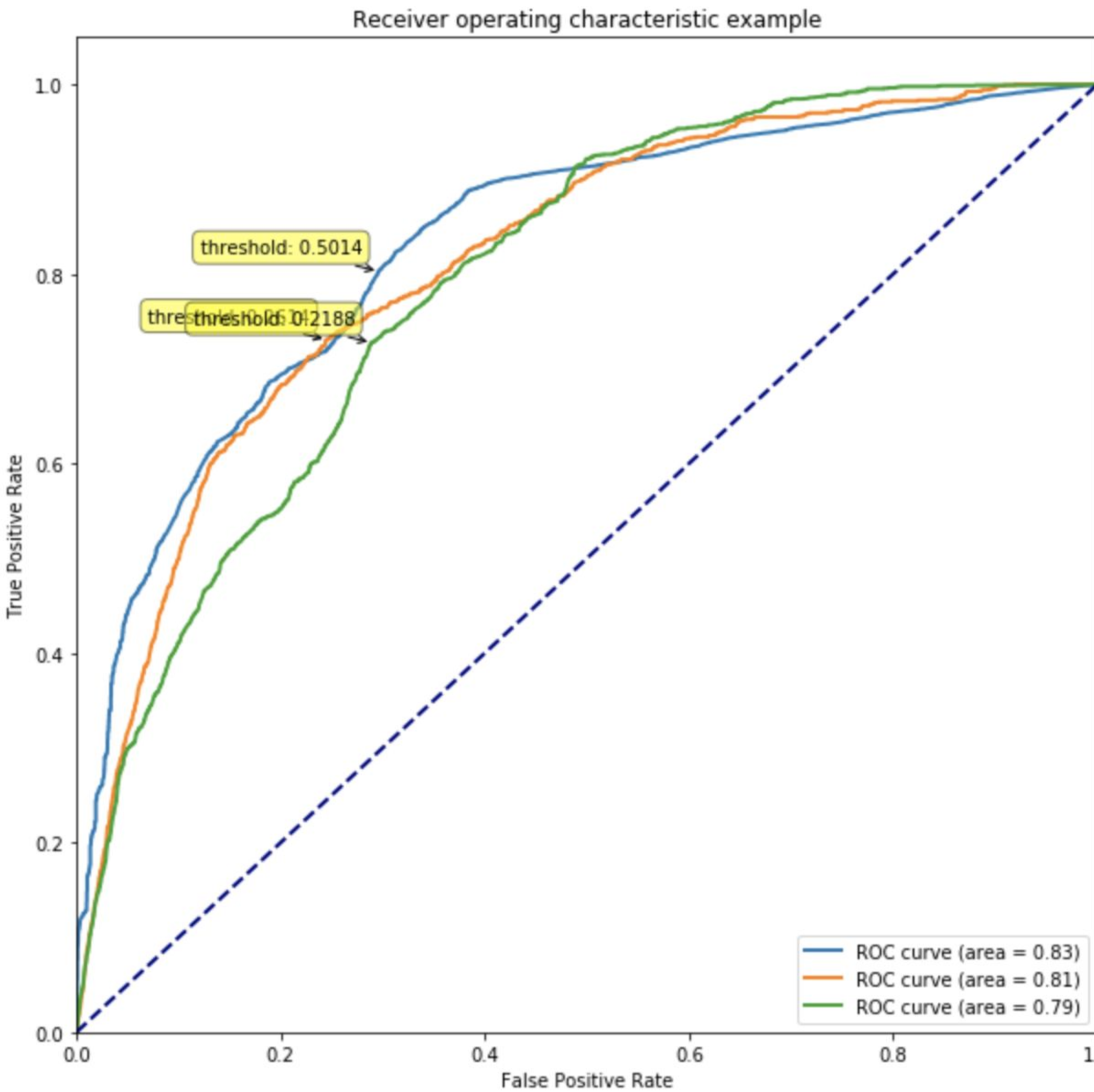
CLASS	**PRECISION**	**RECALL**	**F1**
Normal	0.96	0.90	0.93
Cautionary Surplus	0.13	0.38	0.19
Cautionary Shortage	0.23	0.29	0.26

CONCLUSIONS:

The Thresholds for the model was adjusted by calculating the optimum threshold by minimizing the distance metric between (fpr,tpr) and (0,1) to maximize tpr. An roc curve was plotted using one vs rest classifier by binarizing class labels and resetting the threshold value to optimum values of each class.

PROJECT MC-NULTY SUMMARY
CLASSIFICATION OF FORD GO BIKE TRAFFIC IN SAN FRANCISCO

-Hiranya Krishna Kumar



PROJECT MC-NULTY SUMMARY
CLASSIFICATION OF FORD GO BIKE TRAFFIC IN SAN FRANCISCO

-Hiranya Krishna Kumar

FINAL REPORT AFTER RESETTING THRESHOLDS

```
Accuracy Scores for class 0 after resetting thresholds 0.7955960416803198
CR for class 0 after resetting thresholds
              precision    recall  f1-score   support

           0       0.22      0.70      0.33      3324
           1       0.97      0.80      0.88     42453

   micro avg       0.80      0.80      0.80     45777
   macro avg       0.60      0.75      0.61     45777
weighted avg       0.92      0.80      0.84     45777

Accuracy Scores for class 1 after resetting thresholds 0.48824737313498046
CR for class 1 after resetting thresholds
              precision    recall  f1-score   support

           0       0.99      0.76      0.86     44589
           1       0.07      0.73      0.13      1188

   micro avg       0.75      0.75      0.75     45777
   macro avg       0.53      0.74      0.50     45777
weighted avg       0.97      0.75      0.84     45777

Accuracy Scores for class 2 after resetting thresholds 0.5628663593798924
CR for class 2 after resetting thresholds
              precision    recall  f1-score   support

           0       0.98      0.71      0.82     43641
           1       0.11      0.73      0.19      2136

   micro avg       0.71      0.71      0.71     45777
   macro avg       0.55      0.72      0.51     45777
weighted avg       0.94      0.71      0.80     45777
```

FUTURE WORK

1. Gather realtime-data for bike availability to calculate flux
2. Use Poisson regression to predict bike counts
3. Further tune hyperparameters for random forest , gradient boosting model
4. Build a flask app to classify hourly bike traffic for every station

5. Try other models like adaptive gradient boosting, Xgboost