

# CLASSIFICATION OF FORD GO BIKE TRAFFIC IN SAN FRANCISCO



# MOTIVATION



Bay Area Bike share is rapidly growing with over 300 stations and 3600 bikes

Problem: Imbalanced Traffic

Solution:  
Build model to predict cautionary levels of traffic using classification  
Identify bottlenecks and plan for future rebalancing

# DATA COLLECTION

**Ford** GoBike

Over 500,000 trips collected for 2017. Dock capacity data collected using Ford Go station information API and station region API



Dark Sky API

Weather data collected using python-forecast.io wrapper for Dark Sky API using lat and long coordinates for San Francisco

# METHODOLOGY

## DATA CLEANING AND FEATURE ENGINEERING

Created Target Label:Flux

0: Normal Traffic

1: Cautionary

inflow/Surplus>0.15  
capacity

2: Cautionary

outflow/Shortage

## BALANCE CLASSES

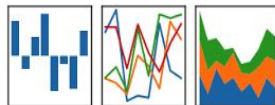
Highly Imbalanced data:

Applied SMOTE ENN to  
balance classes  
(94% majority class)

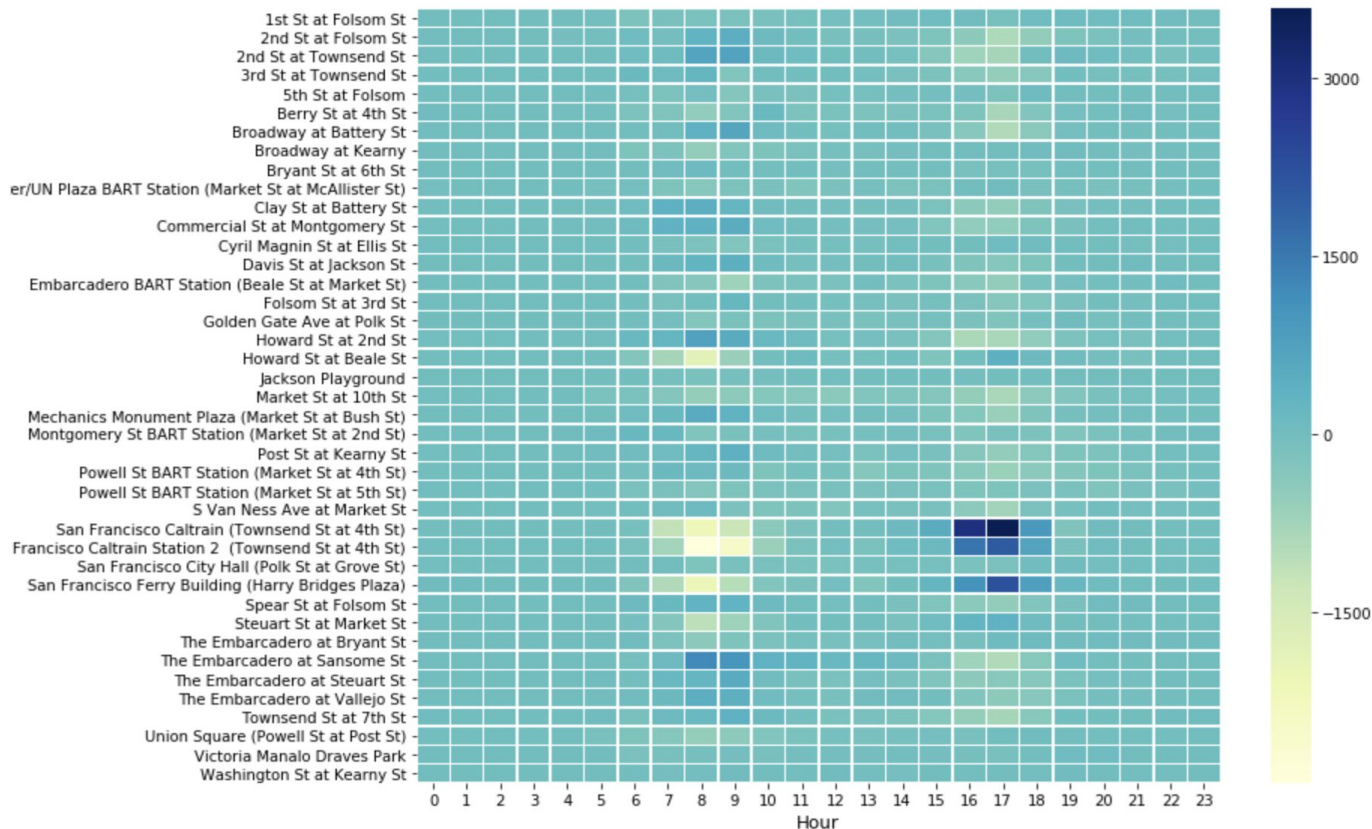
## BEST MODELS

Gradient Boosting

Random Forest

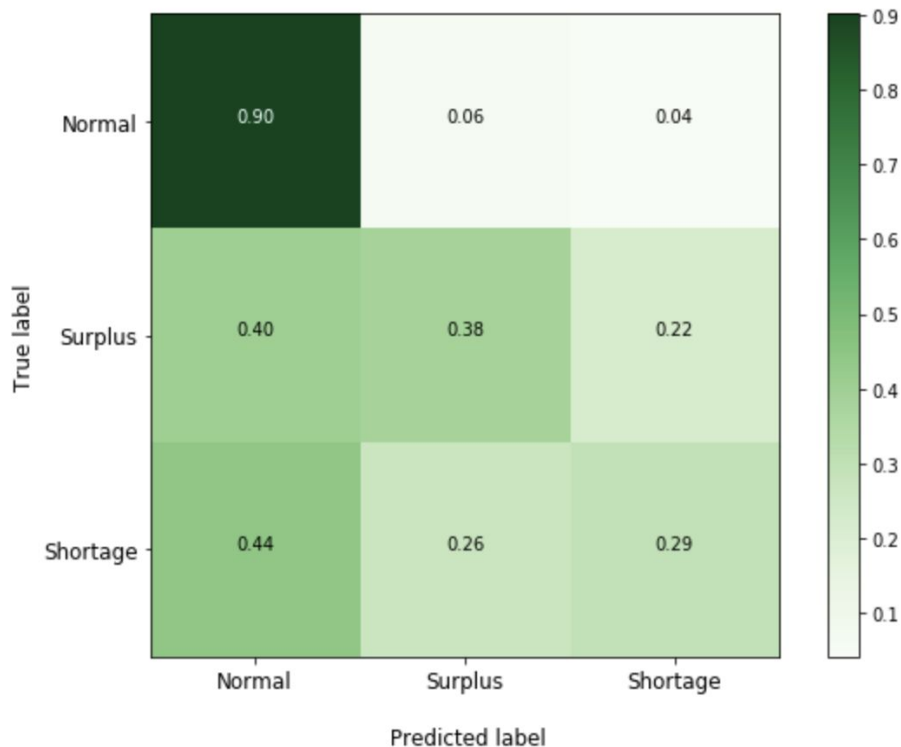


## 2017 Ford Go Bike Stations Traffic by Hour



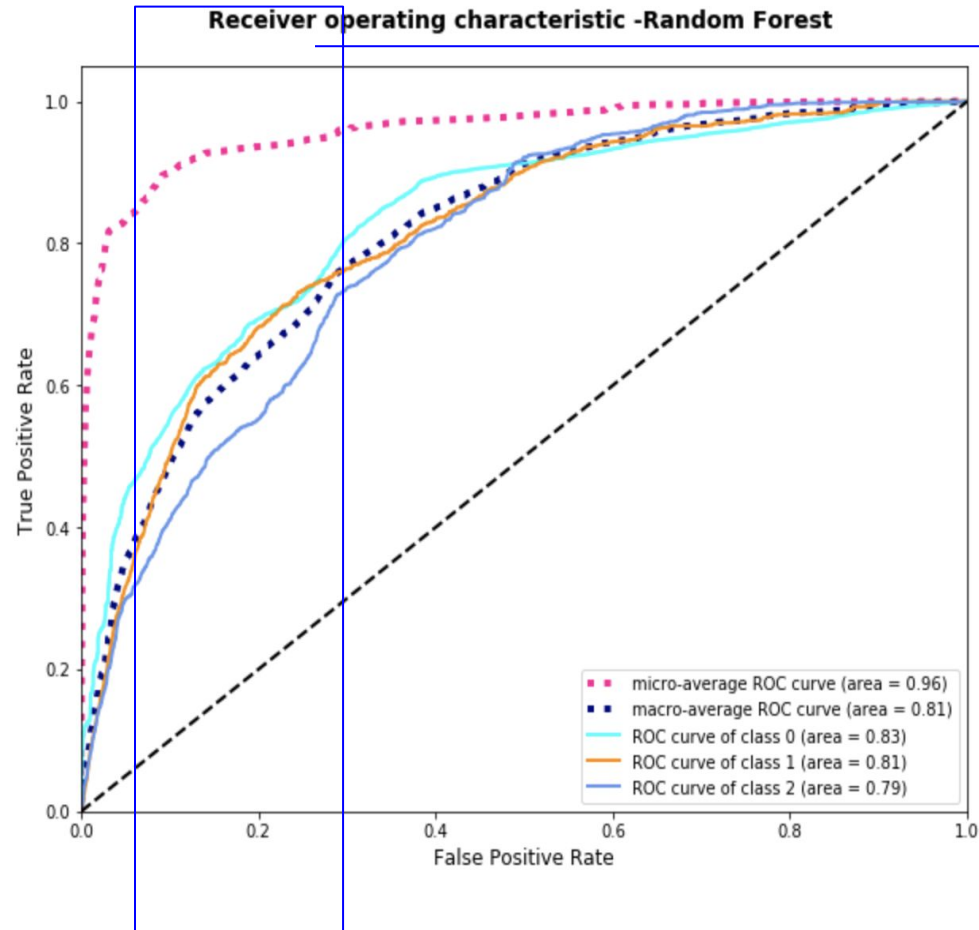
# RESULTS

FINAL MODEL:RANDOM FOREST,STRATIFIED 3 FOLD CV,RANDOMIZED SEARCH



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

# ROC CURVE



Minimize FP for  
Class 0: Normal  
traffic  
Maximize Recall for  
all classes, Minimize  
FN on Class 1 and 2

# FUTURE WORK

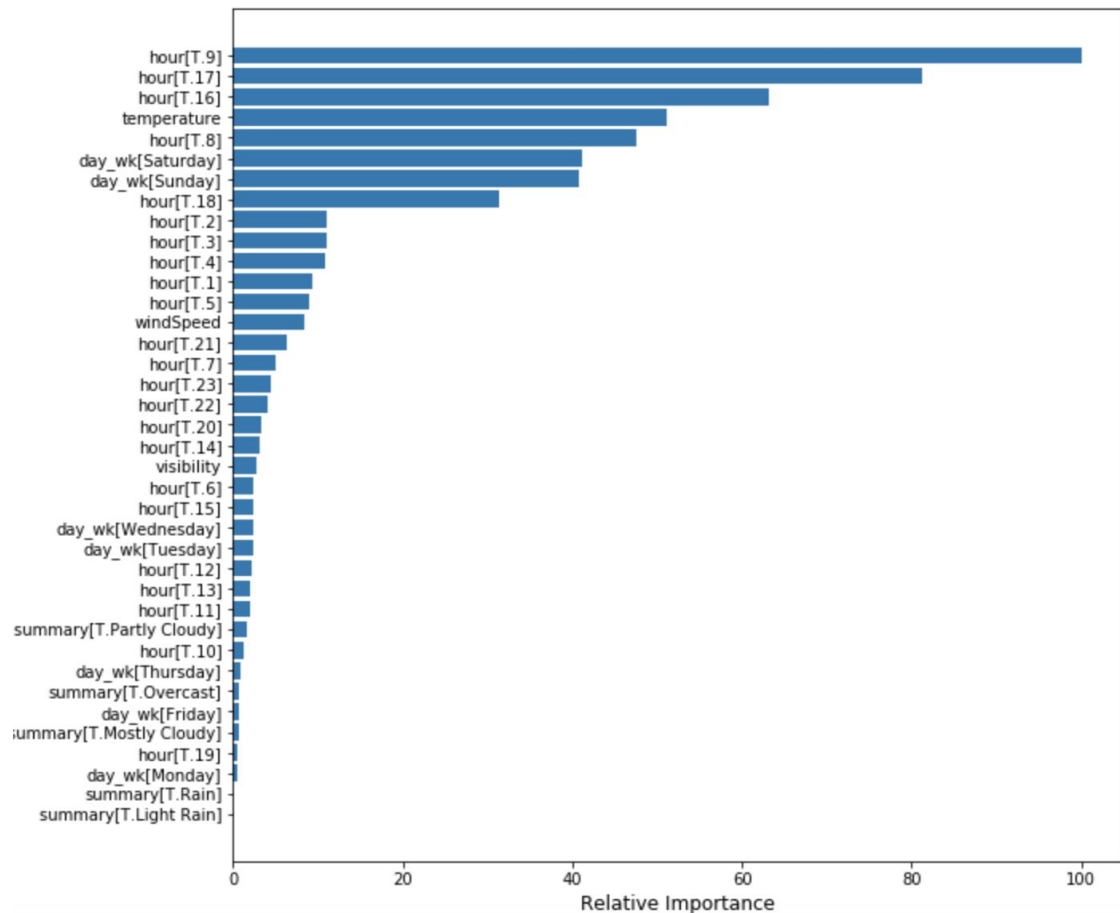
- Gather realtime-data for bike availability to calculate flux
- Use Poisson regression to predict bike counts
- Further tune hyperparameters for random forest , gradient boosting model
- Use only Features with high importance and add more relevant features
- Build a flask app to classify hourly bike traffic for every station
- Try other models like adaptive gradient boosting, Xgboost



THANK YOU!

# APPENDIX

Variable Importance



PARAMS:min\_samples\_l  
eaf=9,n\_estimators=150,  
max\_depth=5

# GRADIENT BOOSTING

	precision	recall	f1-score
0.0	0.97	0.84	0.90
1.0	0.11	0.36	0.17
2.0	0.15	0.35	0.21
micro avg	0.81	0.81	0.81
macro avg	0.41	0.52	0.42