FLIGHT DELAY PREDICTION WITH MACHINE LEARNING MODELS

SPRINGBOARD CAPSTONE PROJECT I BY CUTHBERT LO

INTRODUCTION

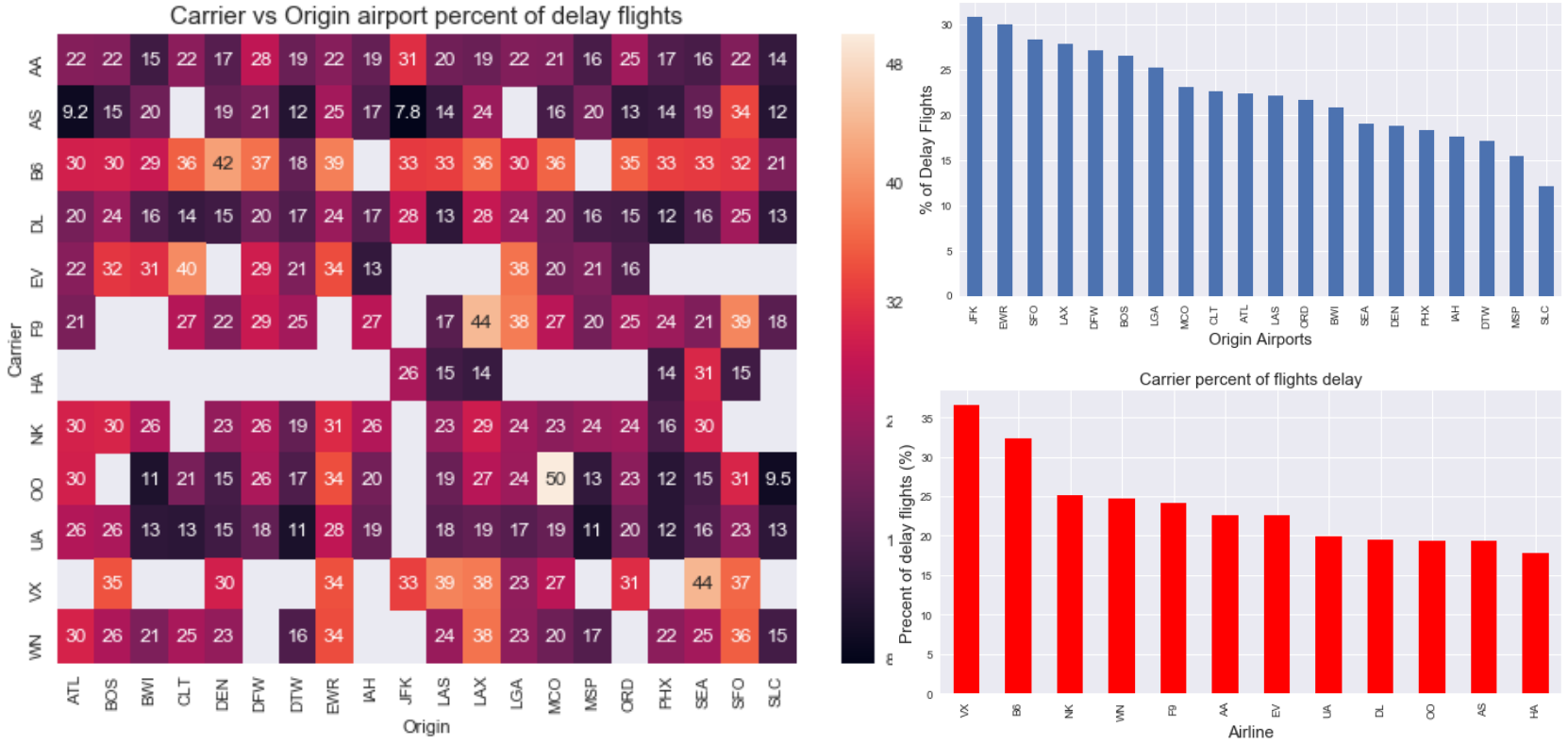
- Flight Delay is unavoidable but we can plan for it maximizing the resource allocation and minimize the impact.
- Air travelers, airlines, airport operators, ATC

DATASETS

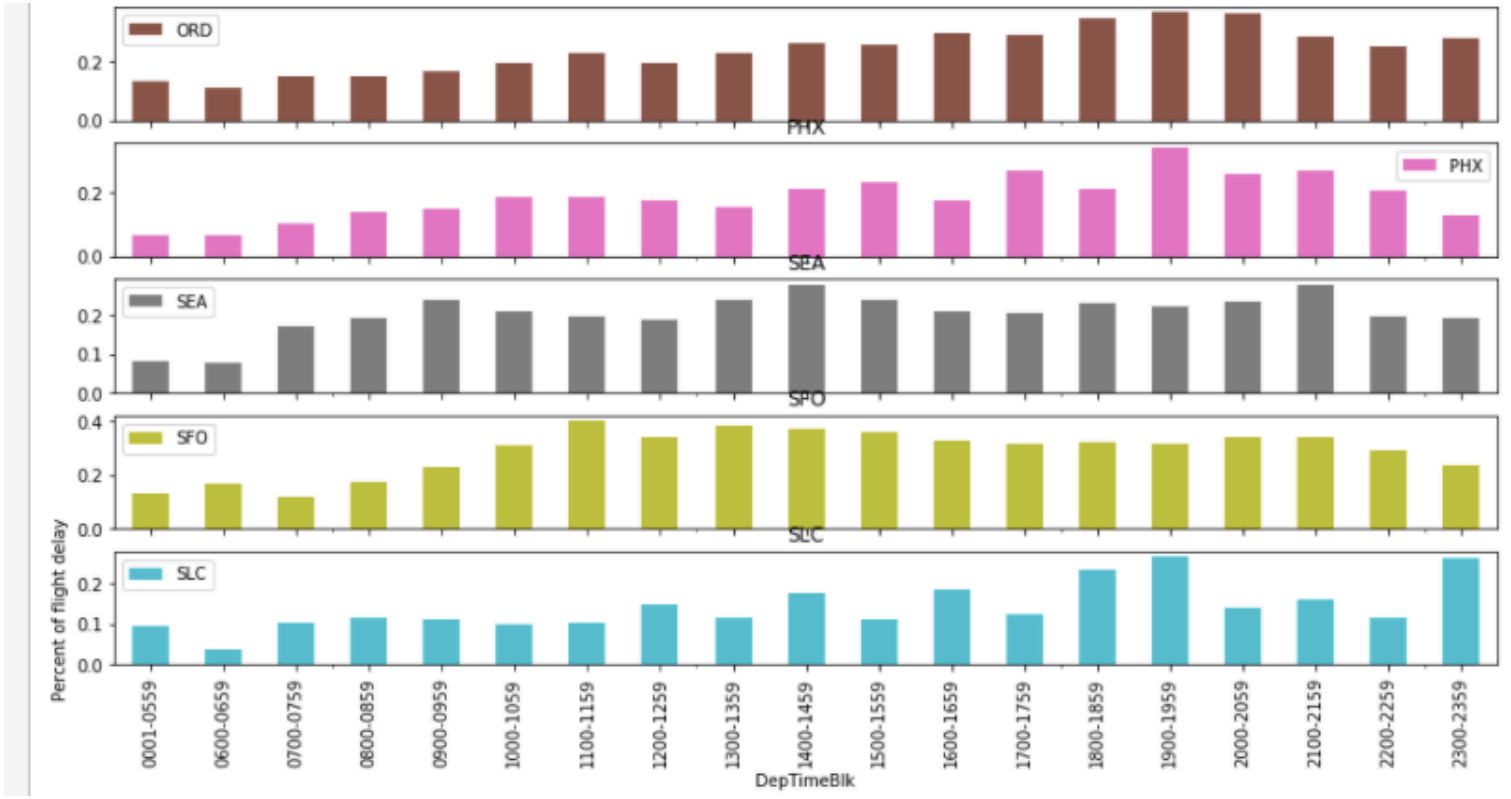
- Data collected from two public sources
  - Airline On-Time Performance Data by Bureau of Transportation Statistics
    - This table contains on-time arrival data for non-stop domestic flights by major air carriers, and provides such additional items as departure and arrival delays, origin and destination airports, flight numbers, scheduled and actual departure and arrival times, cancelled or diverted flights, taxi-out and taxi-in times, air time, and non-stop distance.
  - Automated Surface Observing System (ASOS) Network by Iowa Environmental Mesonet
    - The Automated Surface Observing System (ASOS) is considered to be the flagship automated observing network. Located at airports, the ASOS stations provide essential observations for the National Weather Service (NWS), the Federal Aviation Administration (FAA), and the Department of Defense (DOD). The primary function of the ASOS stations are to take minute-by-minute observations and generate basic weather reports.

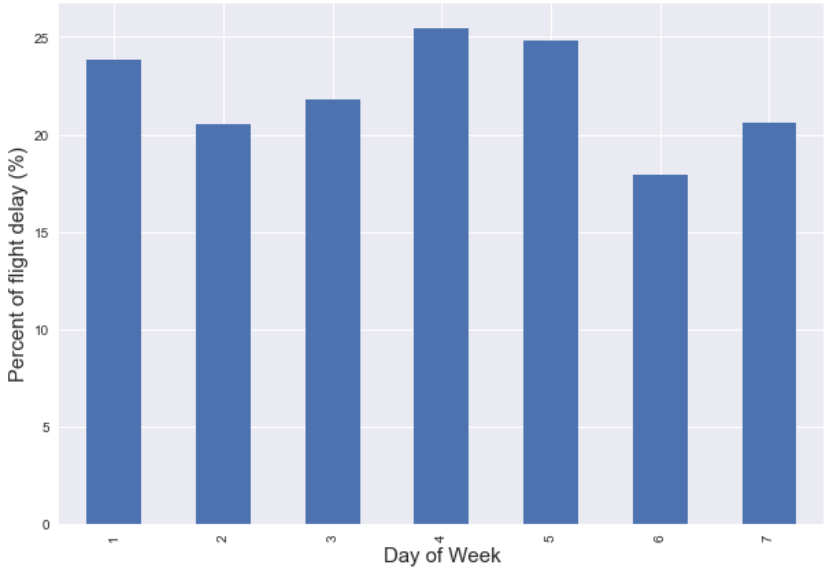
EXPLORATORY ANALYSIS

- Data shows delay at JKF, EWR and DFW are the worst and airlines VX (Virgin America) and B6 (JetBlue) are having lots of delays
- Different airports have it peak hours varies
- Saturdays has less delay among others



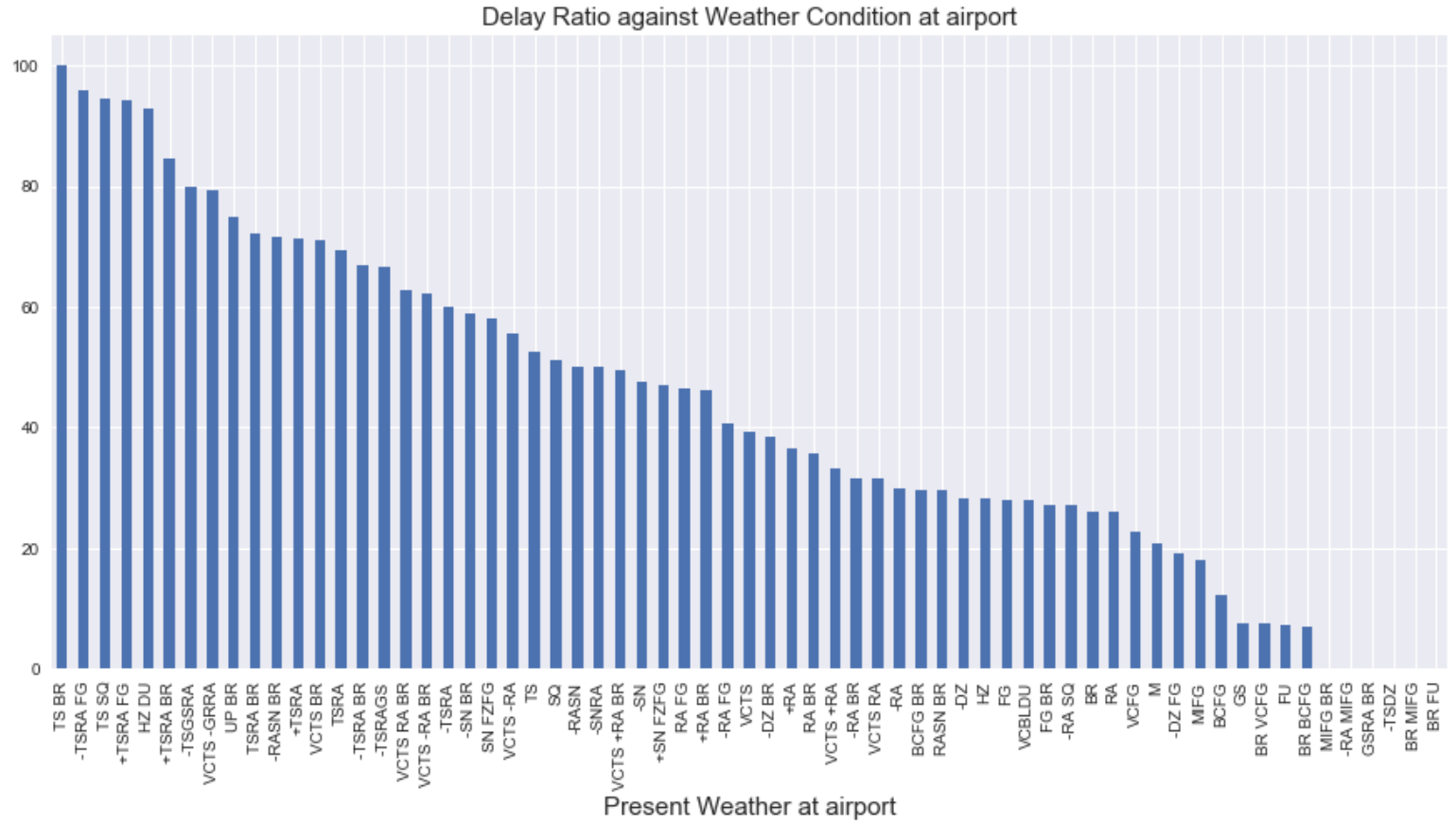
EXPLORATORY ANALYSIS





EXPLORATORY ANALYSIS

- Thunderstorm and Mist (TS BR), the delay ratio is reaching 100%



FEATURES ENGINEERING

- Removed features that no relevant to flight delay like any post flight information: actual arrival time, actual taxi time, etc.
- Added two new features namely
  - Total delayed flight at airport on previous day
  - Total no. of flight at airport today
- Reduced features from 93 to 40

```
In [199]: %%time

df_merged.drop(['FlightDate','TailNum','FlightNum','Flights',
                'CarrierDelay','WeatherDelay','NASDelay','SecurityDelay','LateAircraftDelay',
                'CancellationCode','FirstDepTime','TotalAddGTime','UniqueCarrier','AirlineID',
                'OriginAirportID','OriginAirportSeqID','OriginCityMarketID','OriginStateFips',
                'OriginStateName','OriginCityName','OriginTimeZone',
                'DestAirportID','DestAirportSeqID','DestCityMarketID','DestStateFips',
                'DestStateName','DestCityName',
                'DepTime','DepDelay','DepDelayMinutes','DepDel15','DepartureDelayGroups',
                'ArrDelay','ArrDelayMinutes','ArrDel15','ArrivalDelayGroups',
                'ArrTime','ActualElapsedTime','AirTime',
                'WheelsOn','TaxiIn','TaxiOut','WheelsOff',
                'Cancelled','Diverted','UTCFlightDateTime',
                'valid','station','lat','lon',' mslp','DateHr',' metar'],
               axis=1, inplace=True)

CPU times: user 361 ms, sys: 646 ms, total: 1.01 s
Wall time: 1.98 s
```

DATA PROCESSING

- Label Encoding
  - Convert categorical features in to labels
- Data splitting
  - Training set 70%, Test Set 30%
- Hyperparameters Tuning
  - Grid Search for each model

```
In [8]: from sklearn.preprocessing import LabelEncoder
        le = LabelEncoder()
        tmp = df[cat_list].apply(le.fit_transform)
```

```
# Setup the hyperparameter grid
param_grid = {'n_estimators': [10,100,200,500,1000,2000],
              'max_depth': [3,4,5,6],
              'learning_rate': [0.0001, 0.001, 0.01, 0.1, 0.2, 0.3]}

# instantiate model
gbrt = GradientBoostingClassifier()

# Instantiate the GridSearchCV object: gbrt_cv
gbrt_cv = GridSearchCV(gbrt, param_grid, cv=5)
```

MODELING

- Supervised learning binary classification
- 78% is Class 0 (no delay) and 22% is Class 1 (delay) of three months of data
- models are trained using 70% data and the reminding 30% is used for prediction and evaluation of models' performance.
- Compared 8 algorithms
- Metrics Selection
  - Optimize for Precision to minimize false positive prediction

MODELING

- Baseline model – Dummy Classifier
  - The rate of successfully guessing a delay is 22%

	precision	recall	f1-score	support
Not Delay	0.78	0.78	0.78	18290
Delay	0.22	0.22	0.22	5253
avg / total	0.65	0.65	0.65	23543

- Best Performing Model – Gradient Boosting
  - Precision of predicting a delay jump to 63%, nearly 3 times increase in performance
  - Overall avg F1 score has 17% increase in perfromance

	precision	recall	f1-score	support
Not Delay	0.81	0.96	0.88	18290
Delay	0.63	0.23	0.34	5253
avg / total	0.77	0.80	0.76	23543

MODELING

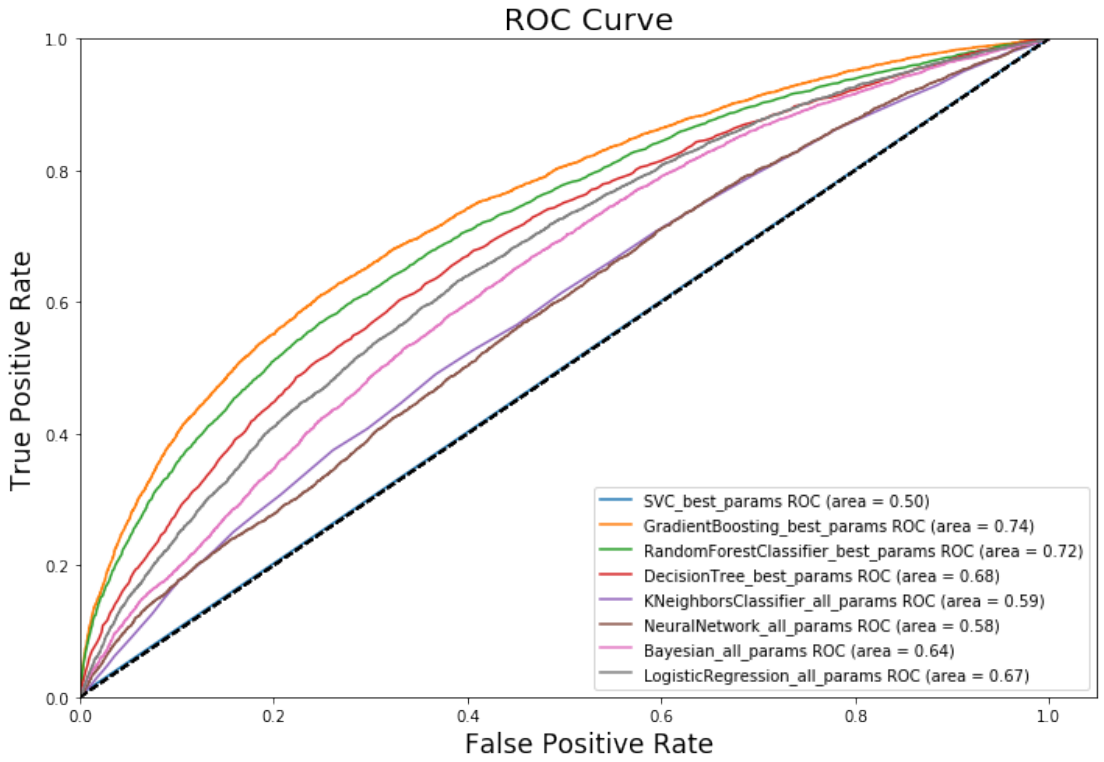
- Gradient Boosting classifier performs the best:
  - 63% for Delay and 81% for Not Delay
  - AUC is 0.74

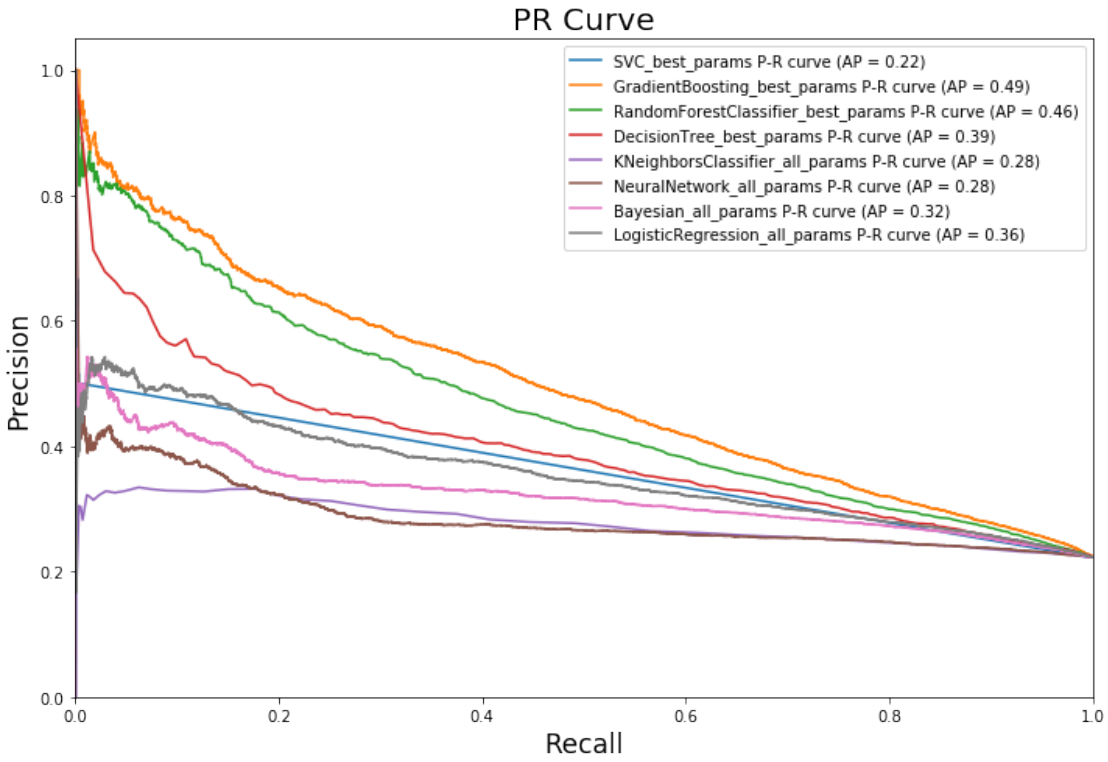
	precision	recall	f1-score	support
Not Delay	0.81	0.96	0.88	18290
Delay	0.63	0.23	0.34	5253
avg / total	0.77	0.80	0.76	23543

- Random Forest is second best:
  - 63% for Delay and 81% for Not Delay
  - AUC is 0.72

	precision	recall	f1-score	support
Not Delay	0.81	0.96	0.88	18290
Delay	0.61	0.20	0.30	5253
avg / total	0.76	0.79	0.75	23543

MODELING


ROC Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
SVC_best_params ROC (area = 0.50)
GradientBoosting_best_params ROC (area = 0.74)
RandomForestClassifier_best_params ROC (area = 0.72)
DecisionTree_best_params ROC (area = 0.68)
KNeighborsClassifier_all_params ROC (area = 0.59)
NeuralNetwork_all_params ROC (area = 0.58)
Bayesian_all_params ROC (area = 0.64)
LogisticRegression_all_params ROC (area = 0.67)


PR Curve
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0
SVC_best_params P-R curve (AP = 0.22)
GradientBoosting_best_params P-R curve (AP = 0.49)
RandomForestClassifier_best_params P-R curve (AP = 0.46)
DecisionTree_best_params P-R curve (AP = 0.39)
KNeighborsClassifier_all_params P-R curve (AP = 0.28)
NeuralNetwork_all_params P-R curve (AP = 0.28)
Bayesian_all_params P-R curve (AP = 0.32)
LogisticRegression_all_params P-R curve (AP = 0.36)

CONCLUSION AND IMPROVEMENT

- Using various performance evaluation metrics, we found that the Gradient Boosting classifier gives the the best model performance.
- 3 times better performance than guessing
- We achieved the ROC AUC to be about 0.74. The AUC for PR was not great (about 0.49) but the precision is reaching 0.63 in positive class and 0.81 in negative class.
- Adding some perspective like airlines operations information e.g. passenger counts, aircraft maintenance history; and historical air traffic control information which would help the model to be more generalize in solving the flight delay prediction problem.