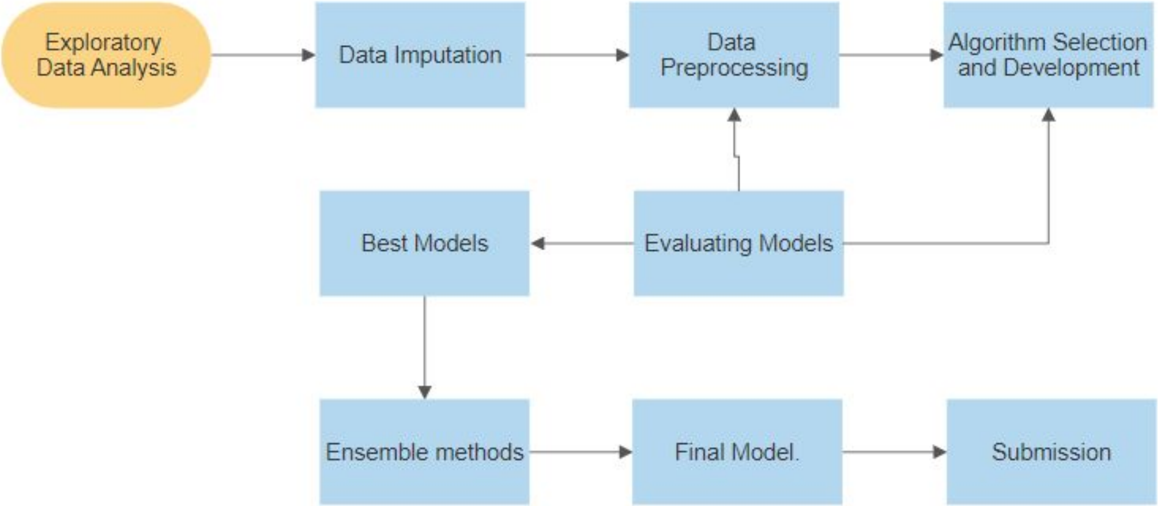
# Using Data to Bring Customers Home

Rohith Nagabhyrava

https://github.com/rnagabhyrava/wayfair-competition

# WorkFlow



Exploratory Data Analysis

Data Imputation

Data Preprocessing

Algorithm Selection and Development

Evaluating Models

Best Models

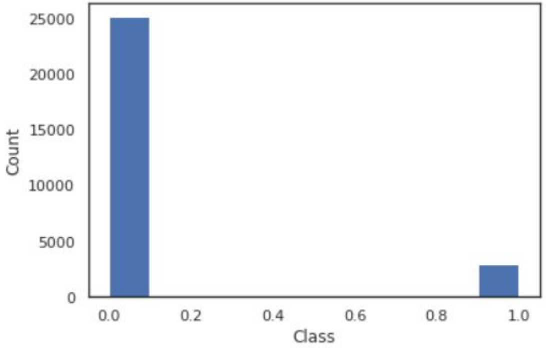
Ensemble methods

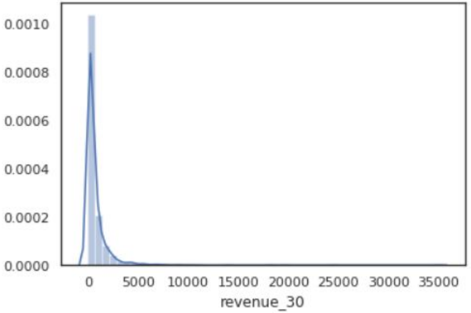
Final Model.

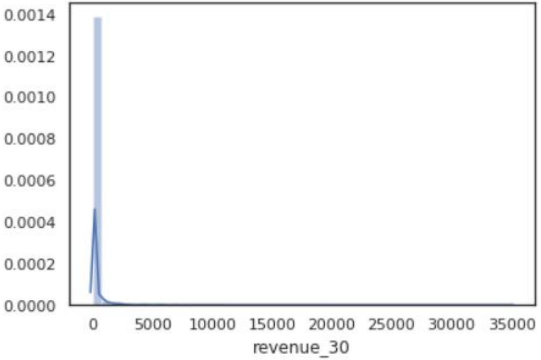
Submission

# Exploratory Data Analysis

- Target Variables are imbalanced.
- Only 8.6% are actually converted
- More than 90% of cases resulted in revenue close to zero.
- No surprise because more than 90% of cases didn't convert
- Most of the observations resulted in less than 5000 (99% cases)







Distribution of Revenue_30

# Data Inspection and Imputation

- Data contains 180 feature variables and 2 prediction variables.
- More than 150 variables contain missing values .
- After careful inspection
  - Missing values in Maxnps, Minnps and avgnps were imputed with mean.
  - All the missing values in other columns were imputed with zeros.

# Classification

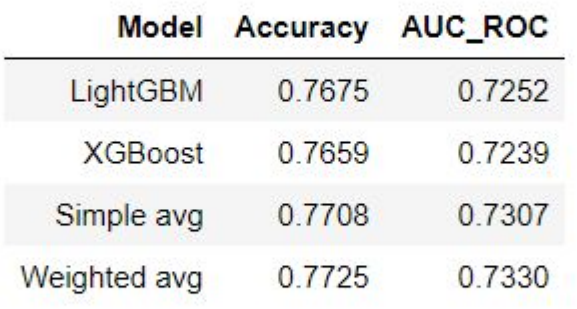
- Data is heavily imbalanced. This problem is approached in two methods.
    - Approach 1: SMOTE, Random Over Sampling, Random UnderSampling.
    - Approach 2: Used models which support class weight.
- SMOTE performed better compared to other sampling techniques.
- But, failed to generalize minority class in testing.
- Metrics:
    - As there is a heavy class imbalance used AUC ROC instead of accuracy.

# Classification

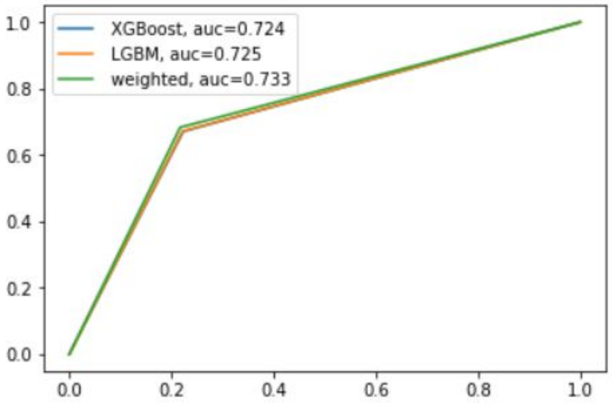
- **Feature Selection:** Variables which are highly correlated with each other were removed.
  - Threshold used >0.90
- Classifiers were fitted on full variables and on data after removing correlated variables and compared.
- Removing the correlated variables helped improve the AUC and Accuracy of the models.
- Classifiers  were tuning with RandomizedsearchCV on various parameter settings.
- Top performing Models: XGBoost, LightGBM.

# Classification results

- Summary of scores from models



| Model | Accuracy | AUC_ROC |
|---|---|---|
| LightGBM | 0.7675 | 0.7252 |
| XGBoost | 0.7659 | 0.7239 |
| Simple avg | 0.7708 | 0.7307 |
| Weighted avg | 0.7725 | 0.7330 |



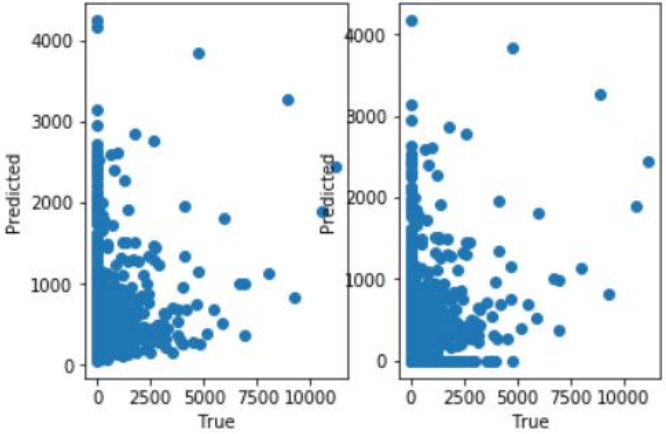
- To further improve the accuracy base LightGBM and XGBoost were weighted averaged. This led to an increase of about 0.01 Accuracy and 0.01 AUC.

# Regression

- Regression was performed only on cases where revenue_30 is greater than 0.
- Models were tuned with and without removing outliers, with and without removing highly correlated variables.
- Best models were than stacked to further improve the RMSE.
- Best model setting was stacking LassoLarsCV with LightGBMRegressor using all the variables.
- To further improve the RMSE used the classifier to first predict the test set and the predicted revenue and made revenue zero where prediction for conversion is zero.
- This decreased the RMSE by about 20%.
- Best RMSE achieved: 417.947

# Regression Results.

- RMSE of **523.258** was achieved by stacking LasoLarsCV and LGBMRegressor
- RMSE was further reduced to **417.94** by replacing revenue with zero where predicted convert_30 is zero
- Model is not so good at predicting small and medium range values but when it comes to values greater than 1000 model fails to predict. Reason is that we don't have much data for extreme values.
- Also most of the zeros were predicted wrongly because our classifier is not much accurate.



# Further Improvements

- Careful selection of variables might help in improving the performances of models.
- Using a CNN to extract classes might help.