Donor Classification

Using AWS SageMaker

Alex Ilyin, Ashish Gupta, Bhawna Gupta

INDEX

Introduction

- To classify if an individual earns more than $50k accurately
- To request optimum donation amounts from individuals based on their income
- Identify attributes of those whore are most likely to donate

Data Structure

Dataset
“Adult” dataset
found on
UCI ML Repo

UCI

Columns:

- Age
- Workclass
- Education
- Education-num
- Marital-status
- Occupation
- Relationship
- Race
- Sex
- Capital-gain
- Capital-loss
- Hours-per-week
- Native-country
- Income

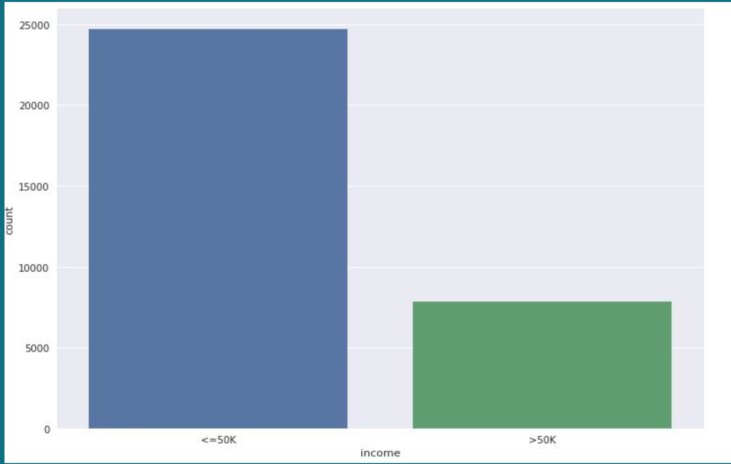
Data Preprocessing :

Dealing with missing values

Transformations on highly skewed features like capital gains/losses

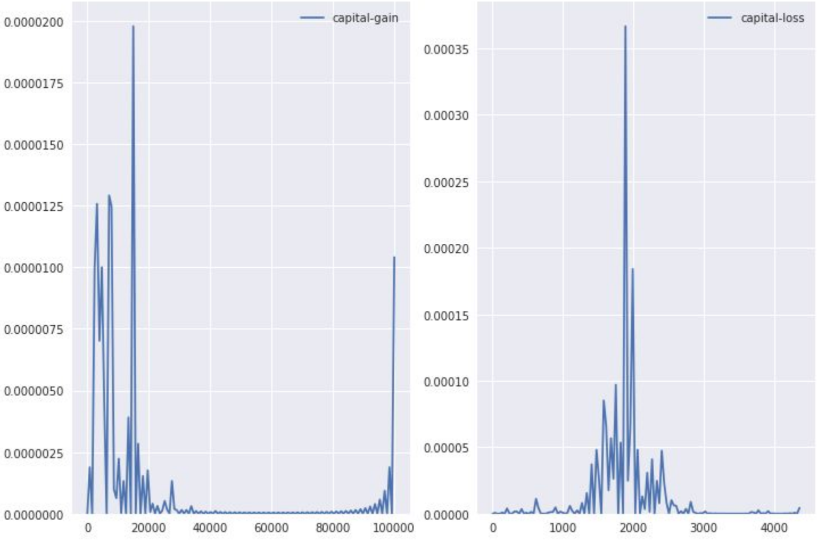
Scaling numeric features and one-hot encoding of categoricals

Income Distribution

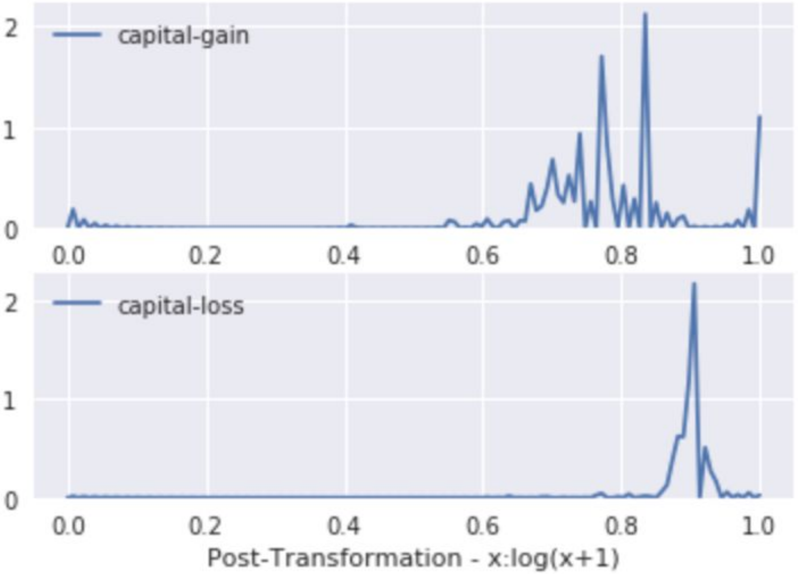


Transformation

- We slightly increment the value and take a logarithmic transformation to spread the data.
- We constrict the data between (0,1) for improving model performance

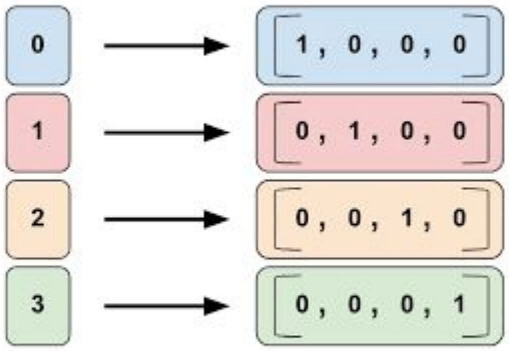


- Skewness in pre-transformed capital gains and loss features

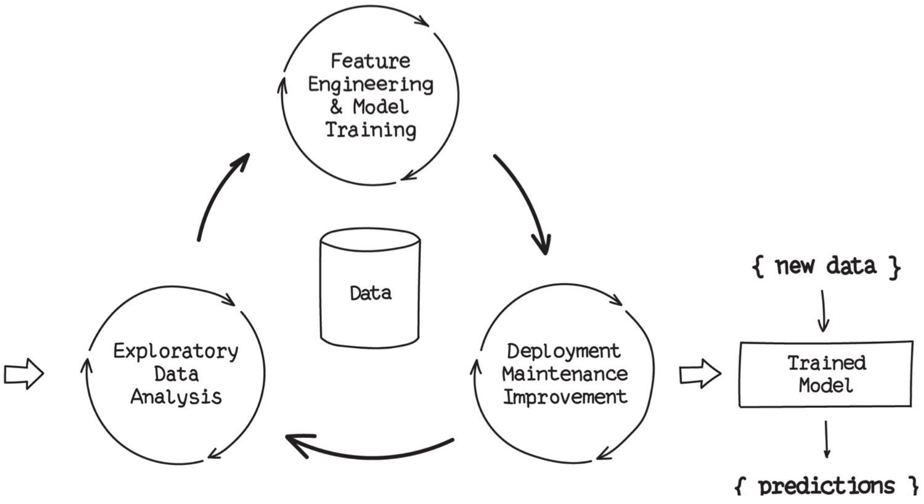


Transformation: OHE

- Before training the model, we have to convert categorical variables to One-Hot Encoded variables
- This is done so the model interprets categorical variables as a vector of numeric values

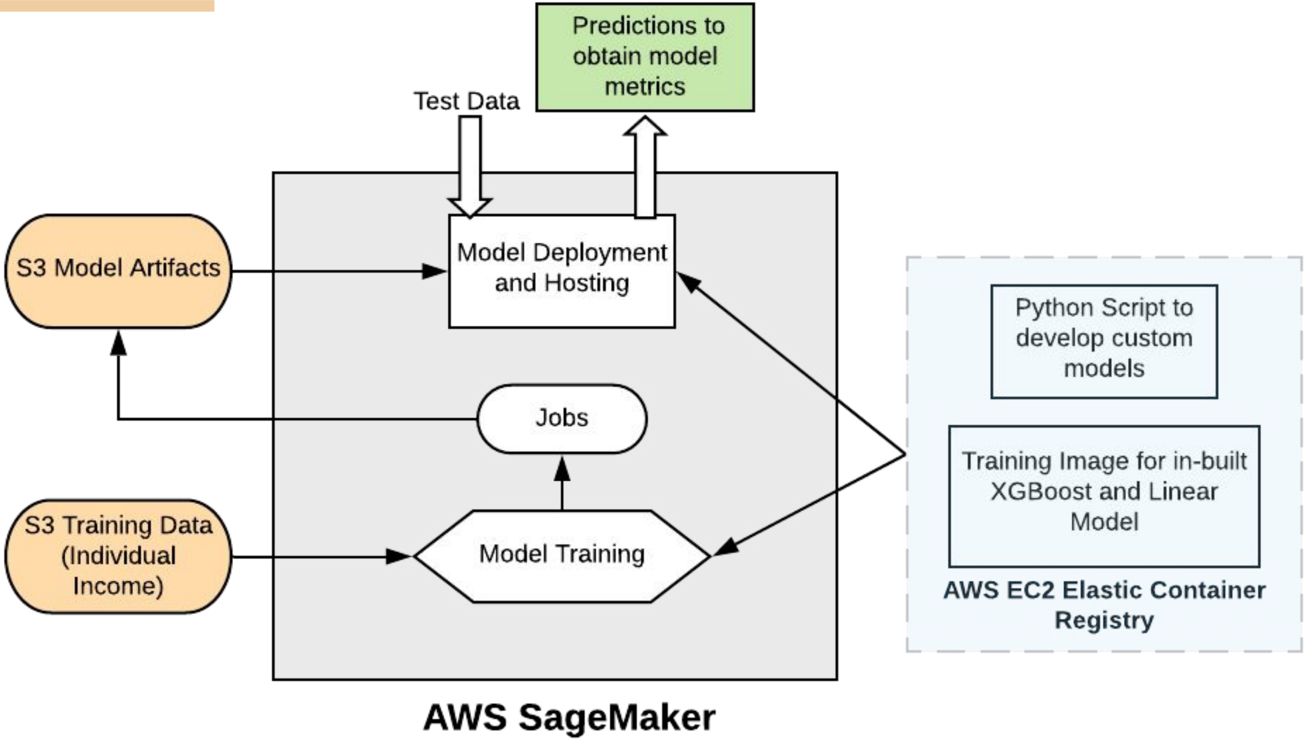


Approaches - General Structure



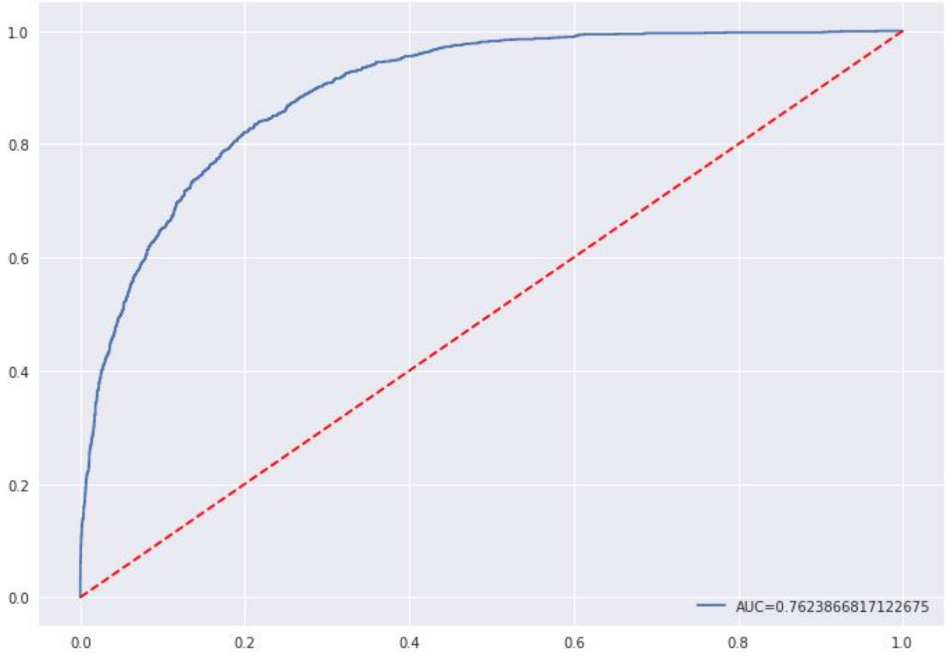
- Models chosen: Logistic Regression, Random Forest, XGBoost
  - Step 1: Generate base model using static hyperparameters
  - Step 2: Use hyperparameter tuning to improve model
  - Step 3: Compare tuned model to base model
  - Step 4: Compare models based on following metrics:
    - Precision, Recall, F1 Score, AUC
  - Step 5: Use best model to generate customer insights

Approaches - Architecture



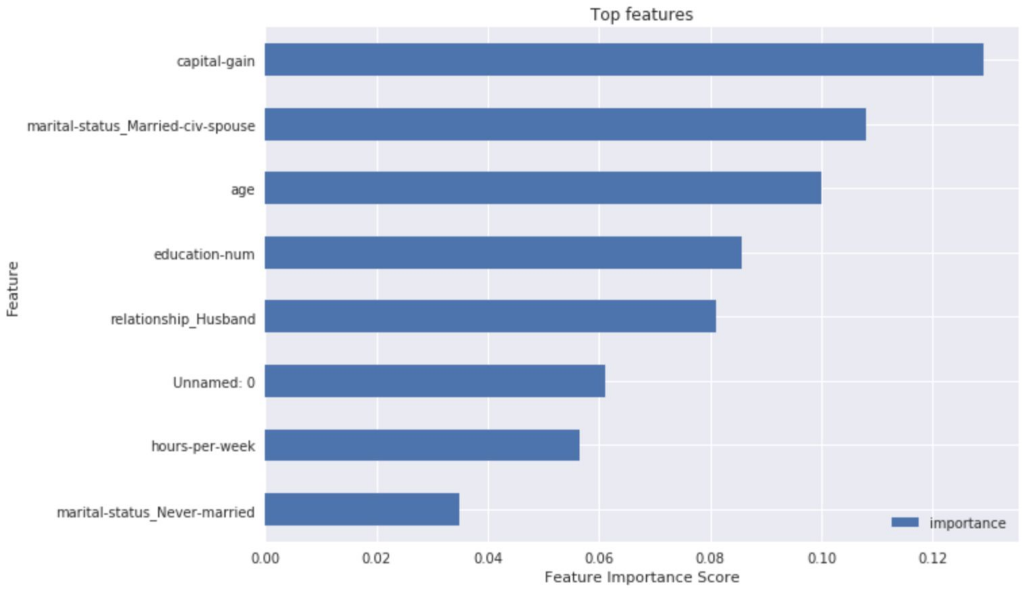
Approaches - Logistic Regression

- Used Sagemaker Linear-Learner and a binary_classifier predictor type
- Challenge: Limited hyperparameters, complicate to extract feature weights
- Best model:
    - L1 = 0.0627
    - Learning_rate = 0.0117
    - Positive Sample Wght = 30.727

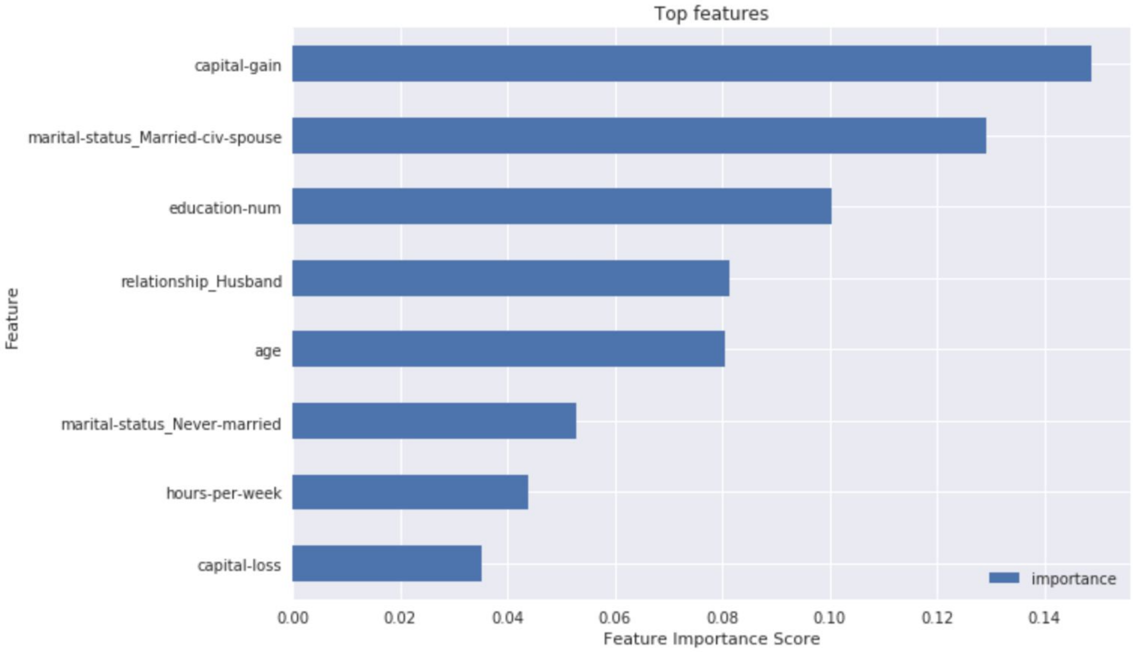


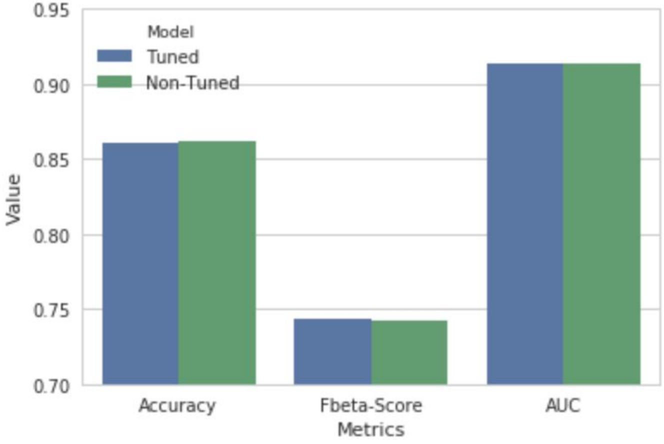
Approaches - RandomForest

- Implemented using Sklearn RandomForestClassifier
- Script fed as entry point to SageMaker
- Training job parameters:
  Num_estimators = 100
  Min_samples_leaf = 2
- Hyperparameter Tuning:
  Num_estimators = 191
  Min_samples_leaf = 5



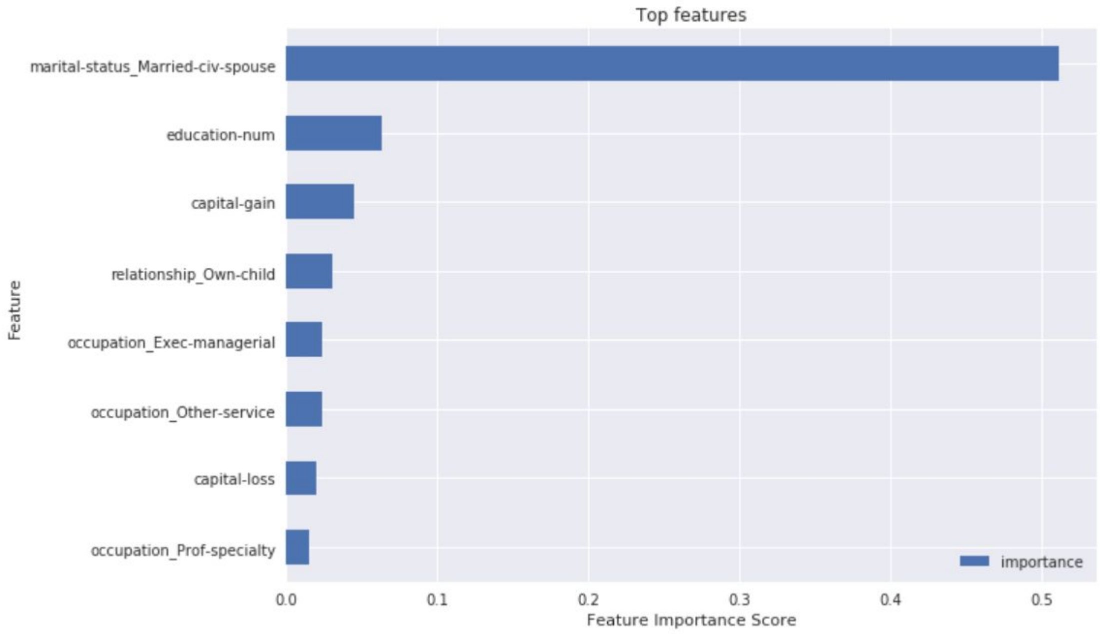
Approaches - Tuned RandomForest





Approaches - XGBoost

- EC2 instance training image is fed into model
- Best model job parameters:
  eta= 0.2,
  gamma = 3,
  max_depth=5,
  min_child_weight=6



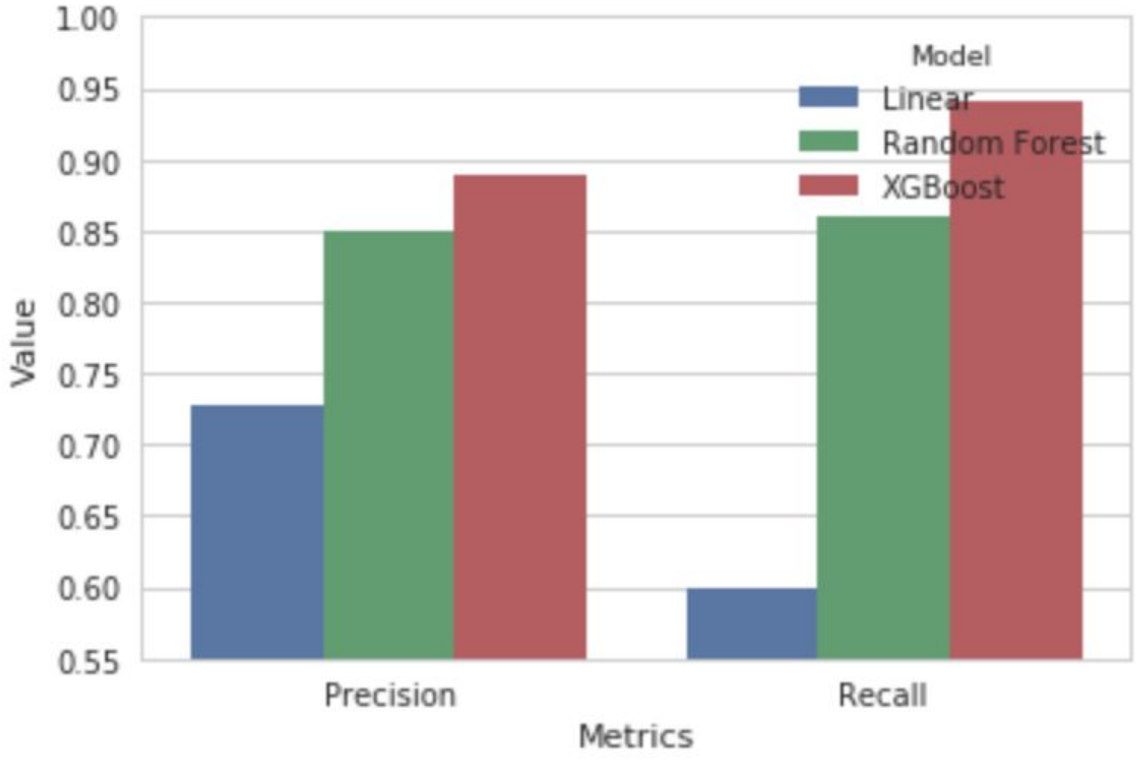
Challenges and Solutions

- Random Forest deployed model endpoint does not allow for predicted probabilities.
  Solution: Extracted saved model using joblib

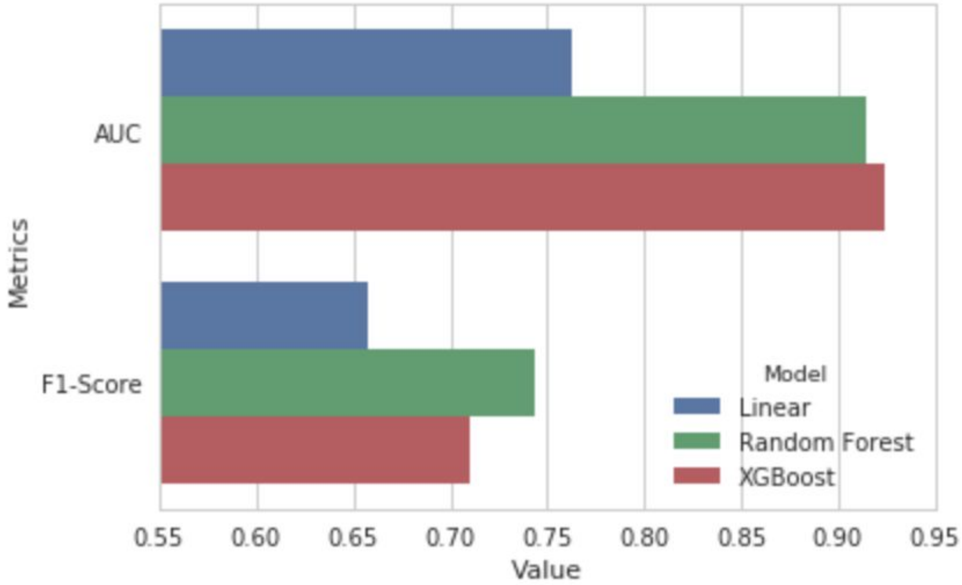
- Poor model performance initially
  Solution: Used Minmaxscaler to scale numeric features

- Relatively poor logistic regression performance
  Solution: Logistic regression was excluded from the final model decision

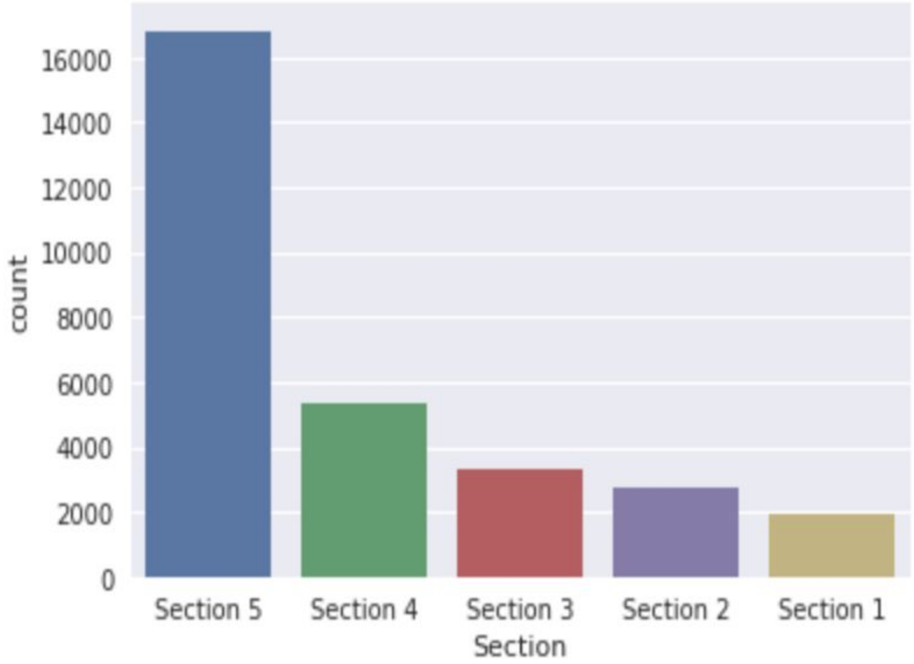
Analysis Results - Model Comparison

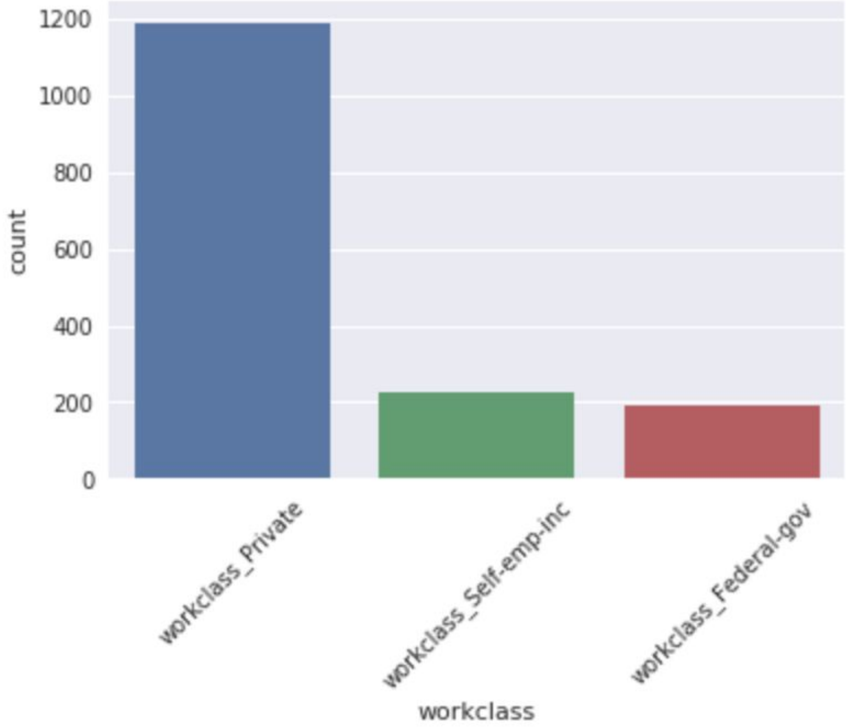


Analysis Results - Model Comparison

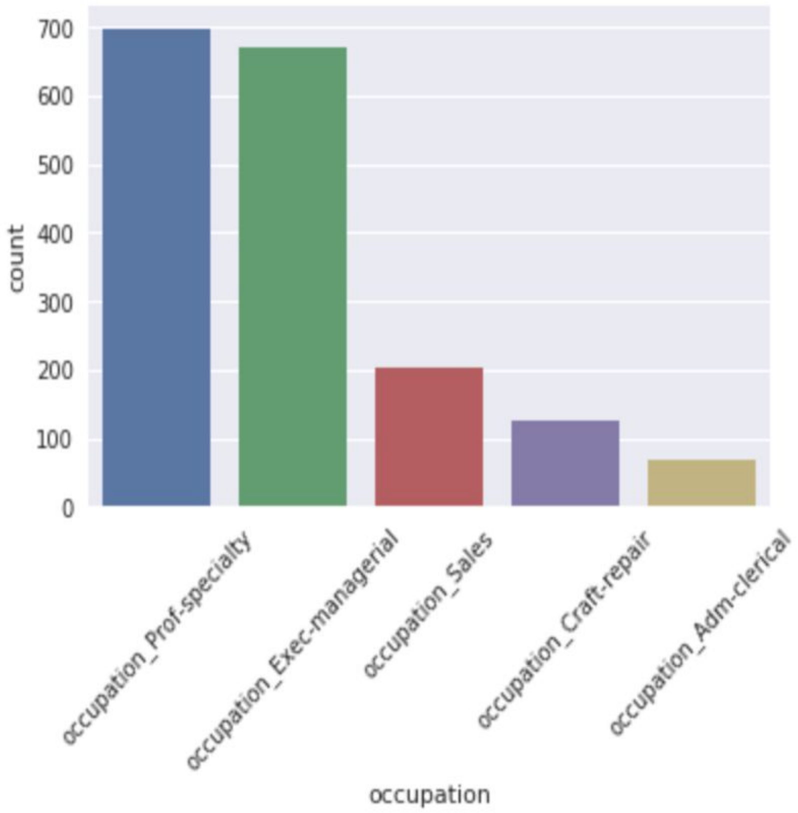


Insights

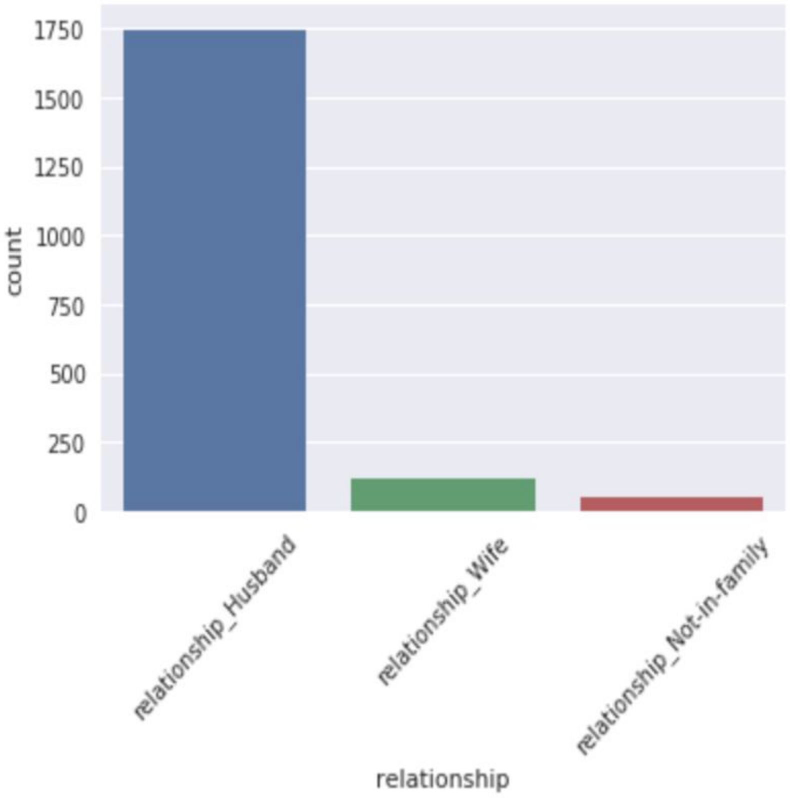

16000
14000
12000
10000
8000
6000
4000
2000
0
count
Section 5
Section 4
Section 3
Section 2
Section 1
Section


1200
1000
800
600
400
200
0
count
workclass_Private
workclass_Self-emp-inc
workclass_Federal-gov
workclass

Insights



700
600
500
400
300
200
100
0
count
occupation_Prof-specialty
occupation_Exec-managerial
occupation_Sales
occupation_Craft-repair
occupation_Adm-clerical
occupation



1750
1500
1250
1000
750
500
250
0
count
relationship_Husband
relationship_Wife
relationship_Not-in-family
relationship

Future Work

- Implementing a recommender engine to match an individual to a donation request in a more granular fashion
- Appending more data points and features to the existing model
- Donation amounts provided by individuals could be incorporated into the model to optimise donation requests

THANK YOU!