Ecom Churn Prediction

By Murali Kadambi

The project is to build and fine tune a model that can predict if a user who adds a product item to his/her cart will proceed to purchase the item from an ecommerce website.

Assumption:

Label 0 (Abandoned Cart) is considered if the user has added an product to the cart and there is no Purchase event associated with the product, irrespective of the session id

Data Analysis

There are about 42M rows in the dataset.

Top 5 products sold

In [21]:
```
# 5 most popular products sold
df.select('event_type', 'product_id')\
.filter(df['event_type'] == 'purchase')\
.groupBy('product_id').count().sort('count', ascending=False).show(5)
```

```
+----------+-----+
|product_id|count|
+----------+-----+
|   1004856|28944|
|   1004767|21806|
|   1004833|12697|
|   1005115|12543|
|   4804056|12381|
+----------+-----+
only showing top 5 rows
```

Top 5 Brands viewed

In [23]:
```python
# 5 most popular brands viewed apart from None
temp_df = df.select('event_type', 'brand')\
        .filter(isnull(df['brand']) == False)\
        .groupBy('brand').count().toPandas()

temp_df.sort_values('count', ascending=False, inplace=True)

print('Top 5 brands viewed:')
temp_df.head(5)
```

```
Top 5 brands viewed:
```

Out[23]:

	brand	count
1732	samsung	5282775
486	apple	4122554
3230	xiaomi	3083763
2417	huawei	1111205
2040	lucente	655861

Number of Unique users

There are about 3M unique users in the dataset

In [25]:
```python
# Number of unique users
df.select('user_id').distinct().count()
```

Out[25]:
```
3022290
```

Most Active User

The user 512475445 has had about 7.5k sessions and is the most active use

In [26]:
```python
# The most active user on the platform
#
temp_df = df.select('user_id', 'user_session').groupBy('user_id')\
.count().sort('count', ascending=False).limit(10)
temp_df.head(1)
```

Out[26]:
```
[Row(user_id='512475445', count=7436)]
```

Top Category Items Sold

In [36]:
```
# Displaying the top 10 category 0 values being sold
df.select('category_0').filter((isnull(df['category_0'])==False) & (df['event_type']=='purchase'))\
                        .groupBy('category_0').count()\
                        .sort('count', ascending=False).show(10)
```

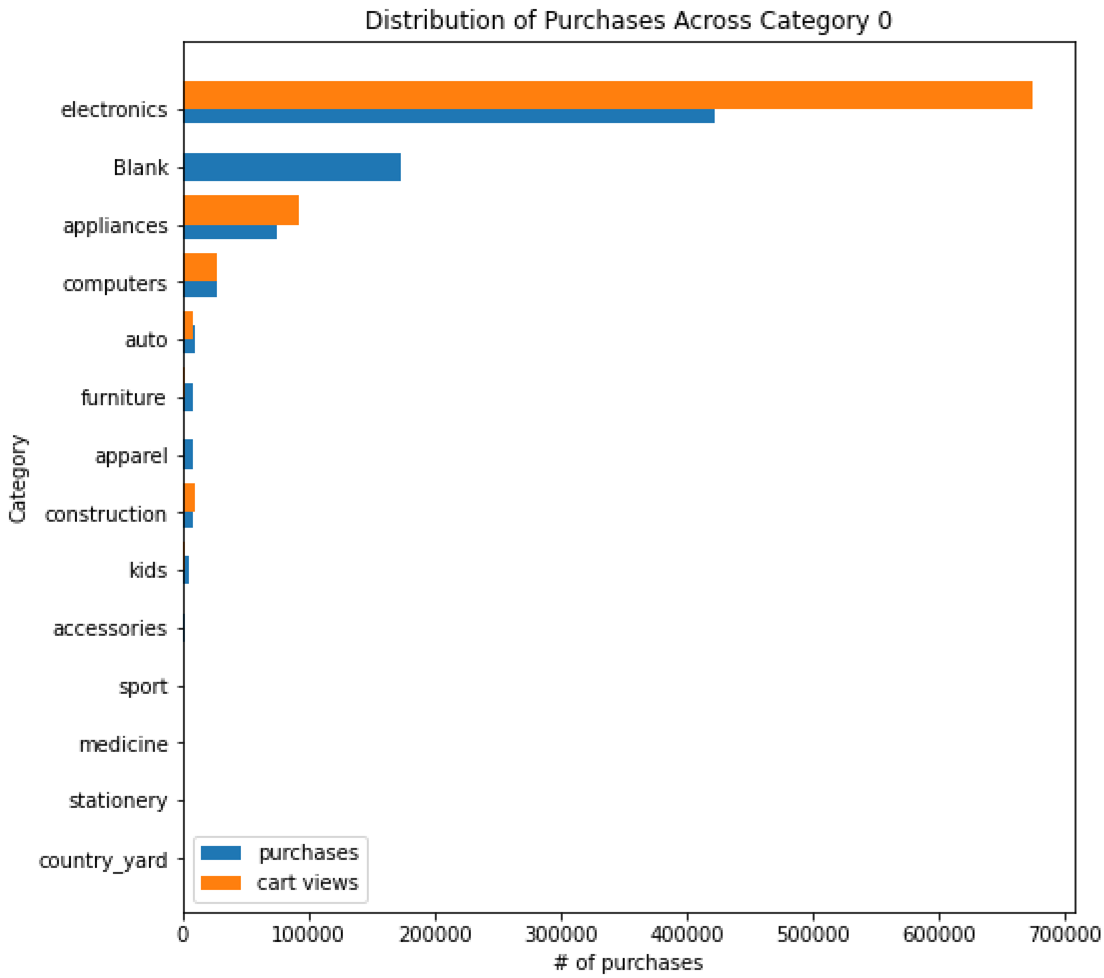
```
+------------+------+
|  category_0| count|
+------------+------+
| electronics|423028|
|  appliances| 74996|
|   computers| 27855|
|        auto| 10620|
|   furniture|  8301|
|     apparel|  8002|
|construction|  7801|
|        kids|  5482|
| accessories|  1587|
|       sport|  1236|
+------------+------+
only showing top 10 rows
```

Avg and Max Smartphone Price

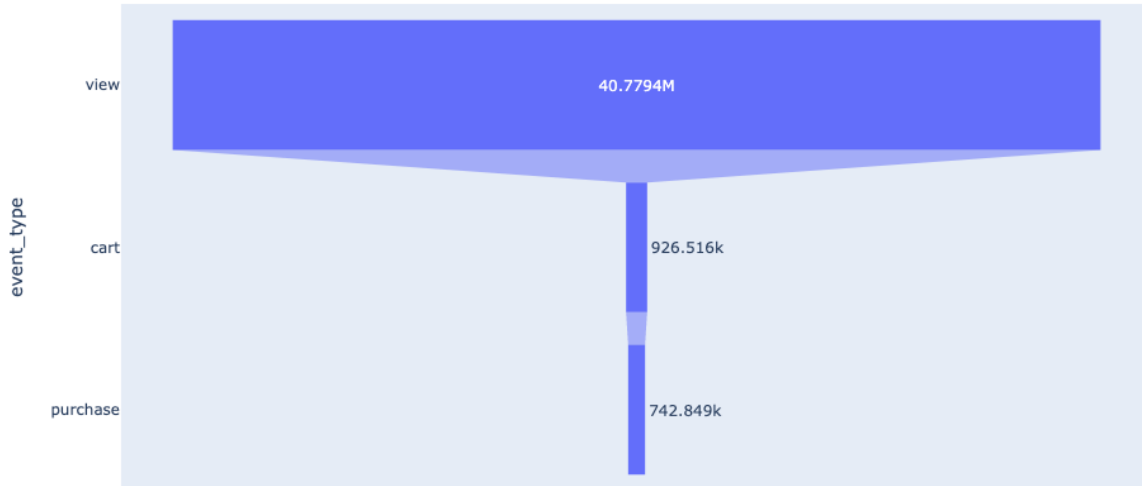
In [37]:
```
# Average and Maximum price for smartphones purchased by the customers
df.filter((df['event_type'] == 'purchase')&(df['category_1'] == 'smartphone'))\
  .select('price').agg(F.mean('price').alias('Average Smartphone Price'),
                       F.max('price').alias('Max Smartphone price')
                      ).show(truncate=False)
```

```
+------------------------+--------------------+
|Average Smartphone Price|Max Smartphone price|
+------------------------+--------------------+
|464.61911297894596      |2110.45             |
+------------------------+--------------------+
```

Distribution of views vs. buys in Primary Category



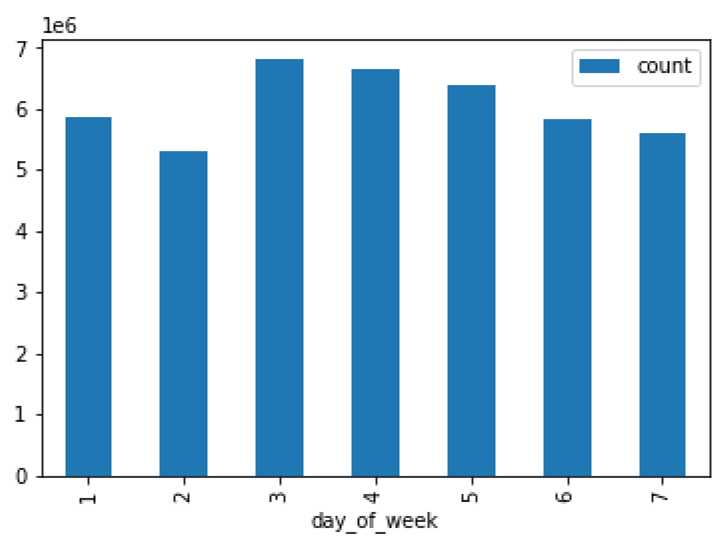
Funnel Diagram



Traffic Distribution

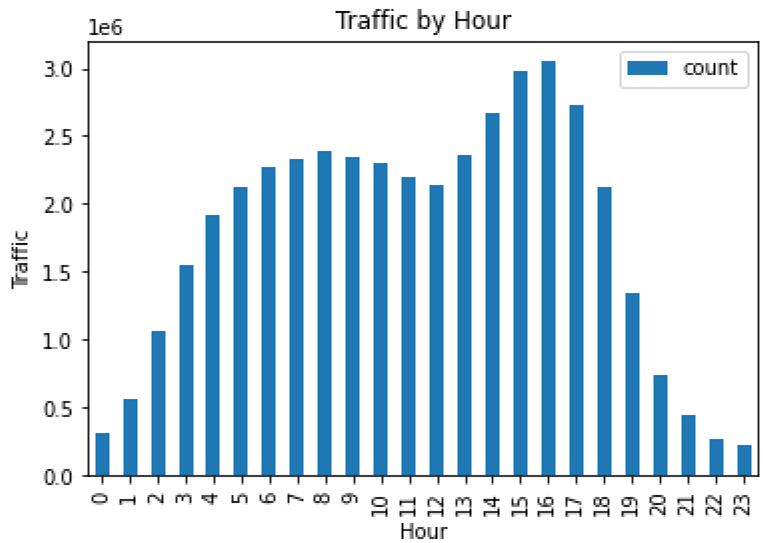
Traffic by Days of Week

Tuesday seems to have the most traffic in a week



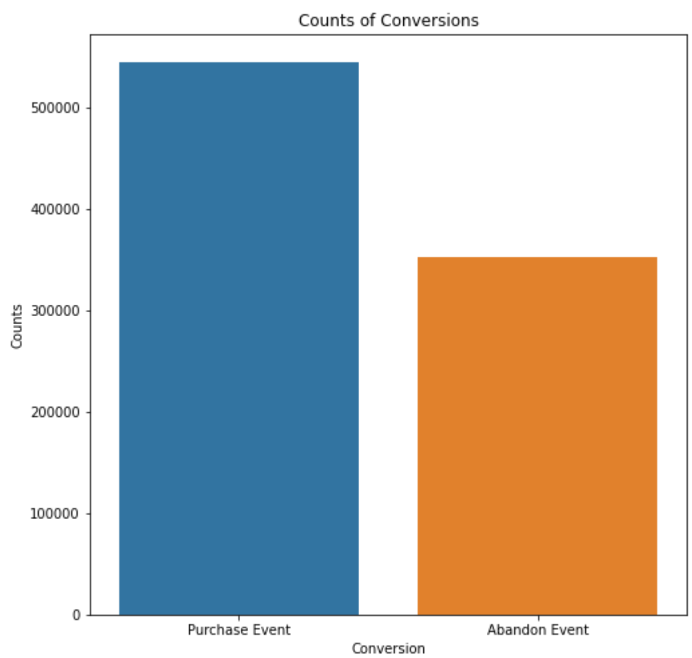
Traffic by Hour

As can be seen below, peak traffic is around 4pm

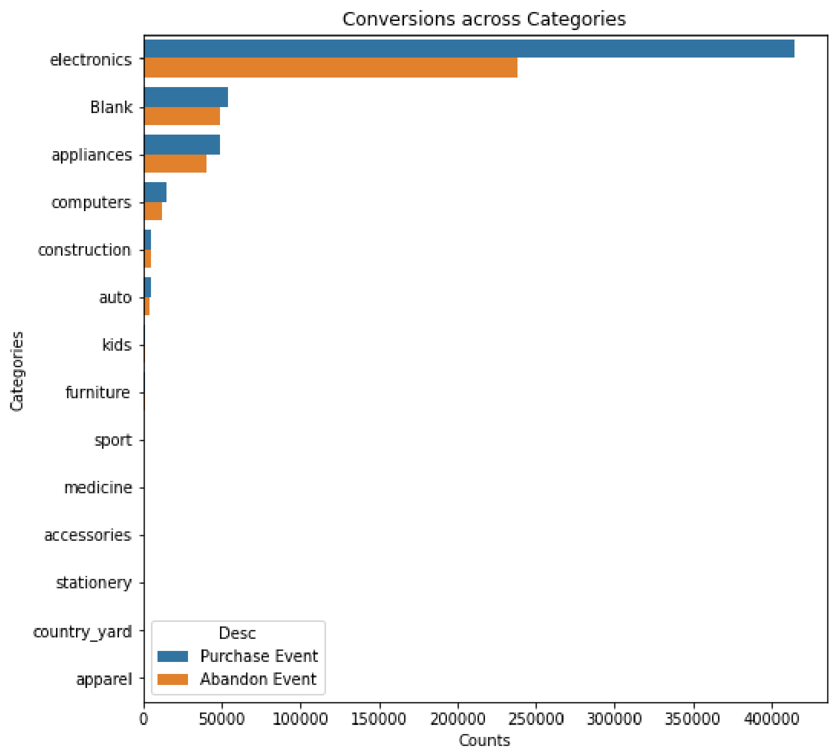


Target Label Distribution

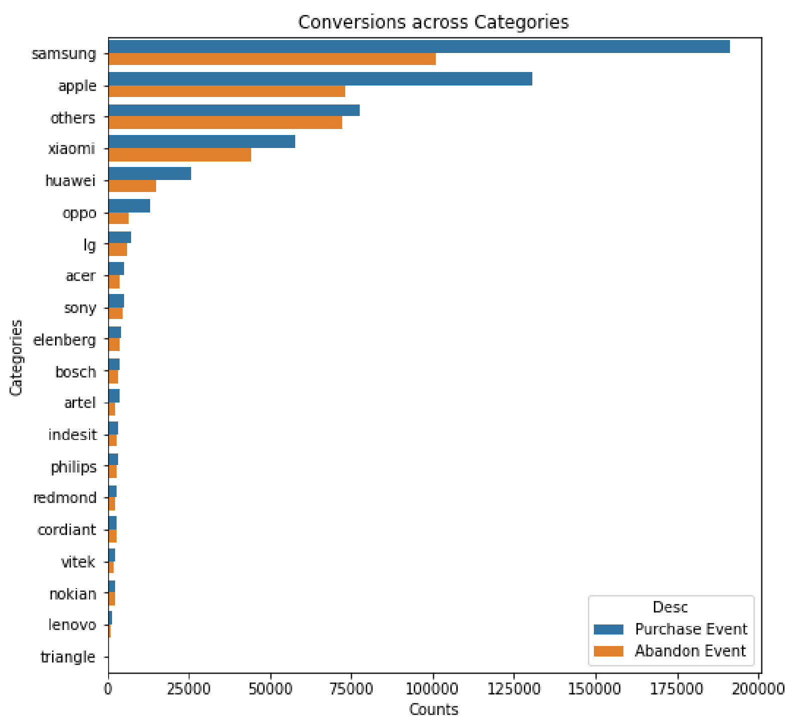
Label 0 is considered as Abandoned cart, whereas label 1 is considered as cart that proceeded to the Purchase flow
As can be seen below there are higher purchase events than abandoned events



Most of the abandoned carts are having electronics products, which also is the category with the most sales



Most of the abandon events happen Samsung brand products



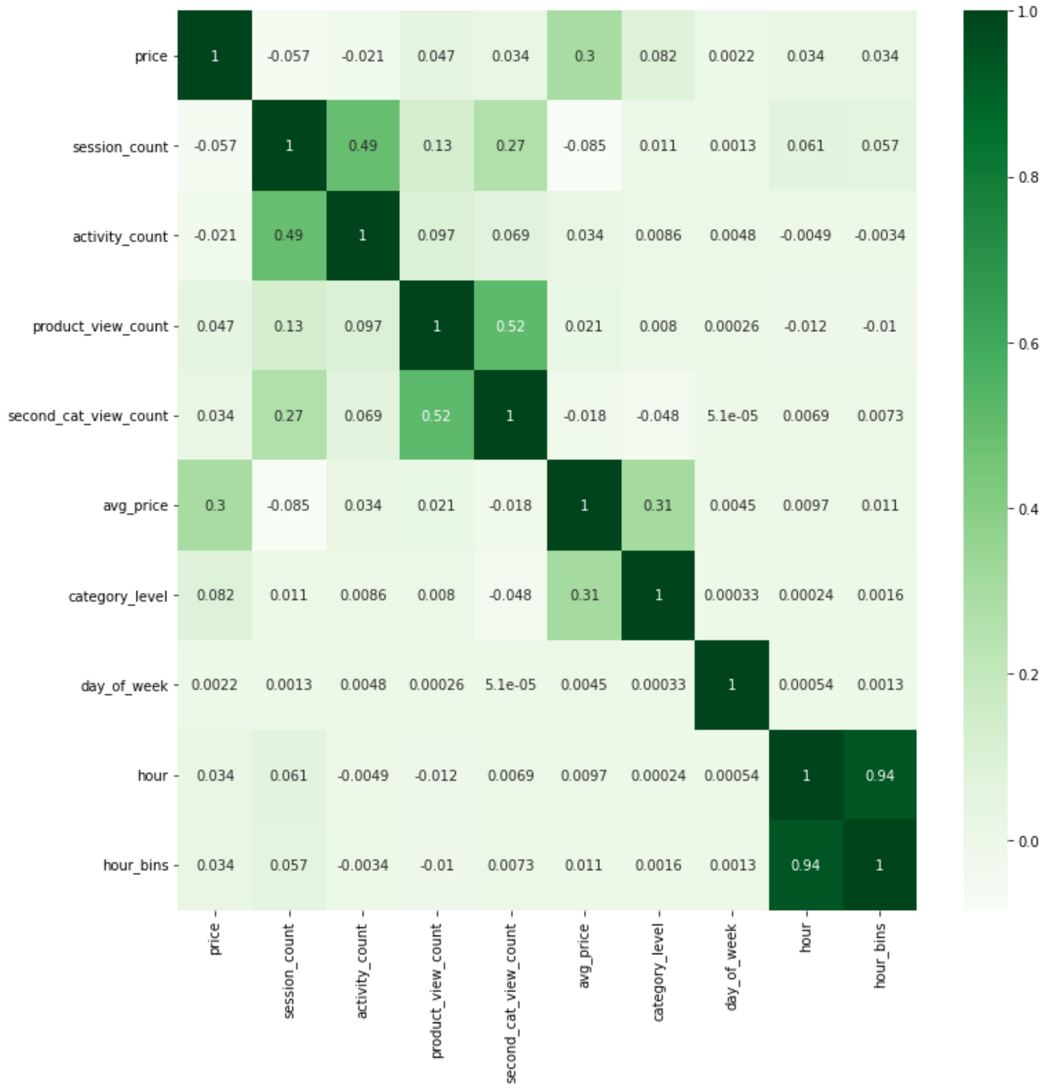
Modeling Approach

Data was modeled with 3 models, Logistic Regression, Decision Trees, Random Forest.

Modeling Criteria: Higher Recall was preferred so that we can minimize the False Negatives, to avoid missing predicting any abandoned carts

Feature Selection

Numerical Features were selected using a correlation map. Categorical Features were selected using a Chi Square Selector using p-value of 0.05. Here hours column was dropped because of high correlation with hours_bin column



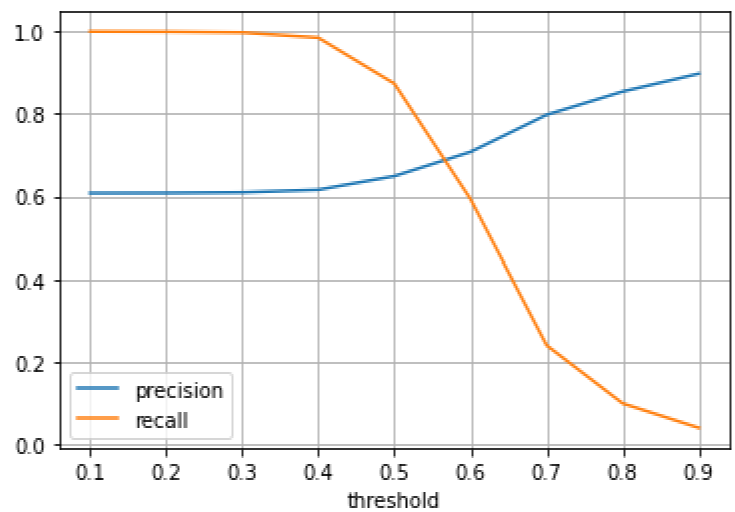
```
In [22]: print('order of importance of features = ')
         np.array(vector_input_cols)[chi_sq_model.selectedFeatures]

         order of importance of features =

Out[22]: array(['cat_0_cln_ix', 'cat_1_cln_ix', 'brand_cln_ix', 'category_level',
                'day_of_week', 'hour', 'hour_bins'], dtype='<U14')
```

Logistic Regression

Logistic Regression model was found to give optimal decision at a threshold of 0.55



Decision Tree

Optimal Decision Tree was found by performing a grid search on the hyper parameters. The tuned model had the following hyperparameter values

- Impurity: gini
- maxBins: 5
- Max Depth of tree: 7
- minInstancesPerNode: 100

Random Forest

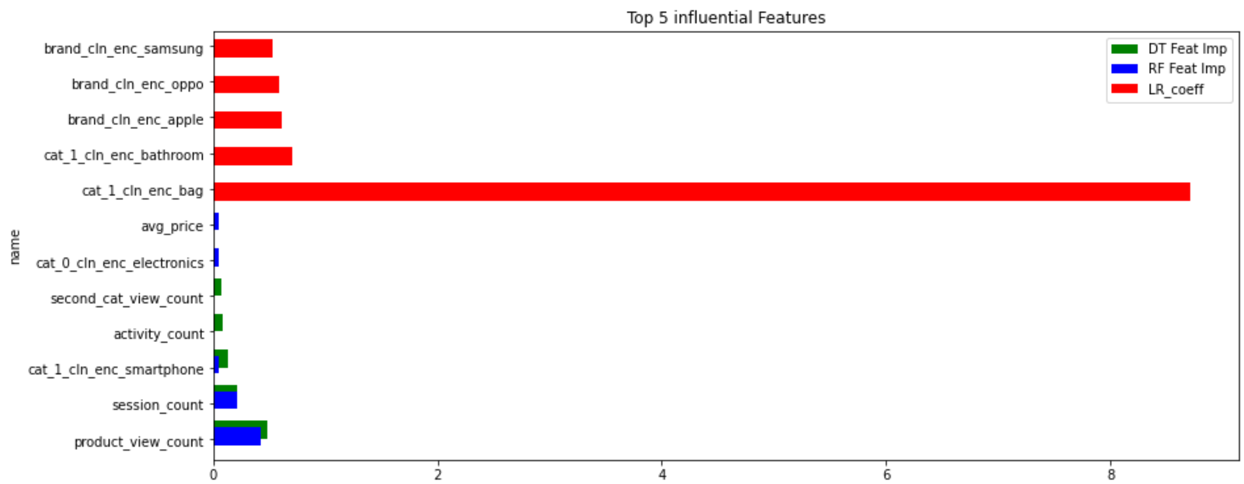
Optimal Random Forest was found by performing a grid search. The tuned model had the following hyperparameter values

- featureSubsetStrategy: sqrt
- Impurity: entropy
- maxBins: 20
- maxDepth: 5

- minInstancesPerNode: 5
- numTrees: 30

Feature Importance

Below are the Top 5 influential features of each of the models



Performance Metrics

Performance metrics of the models are as below:

Model/ Metrics	Precision	Recall	AUC	F1 score
Logistic Regression, threshold=0.55	0.6749	0.7548	0.6522	0.7126
Decision Trees	0.6715	0.8487	0.6035	0.7498
Random Forest	0.6459	0.9198	0.5703	0.75895

The Random Forest model has the highest Recall with 0.9198, hence this shall be selected