Project facilitator: Prashant Bhide
(Overall contribution is 55%)

Project partner: Anilkumar Narayanam
(Overall contribution is 45%)

Document all the EDA, Model Building and Model Evaluation steps

1. Top five rows of the data set at the beginning of the analysis

From the [non_event_data] or device_brand dataset:

```
display(df_device_brand)
df_device_brand.info()
```

	0	1	2	3	4	5
0	-1819925713085810000	F	23	F0-24	OPPO	N1 Mini
1	3670076507269740000	M	33	M32+	Meizu	menote1 2
2	5333872006968810000	M	34	M32+	Xiaomi	xnote
3	4216041491117040000	M	60	M32+	Ishi	ihv1
4	-3441149835823130000	M	30	M25-32	Huawei	è□£è€€ç•…çŽ©5X
5	600258969813393000	M	42	M32+	Meizu	é…è"□2

From the [event_data] dataset:

```
display(df_events)
df_events.info()
```

	0	1	2	3	4	5	6	7
0	-1000369272589010000	F	26	F25-32	\N	\N	\N	\N
1	-1000572055892390000	F	27	F25-32	\N	\N	\N	\N
2	-1000643208750510000	M	29	M25-32	\N	\N	\N	\N
3	-1001337759327040000	M	30	M25-32	2774404	2016-05-07 09:14:24.0	119.61	29.7
4	-1001337759327040000	M	30	M25-32	3065018	2016-05-04 10:26:14.0	120.29	30.42
5	-1001337759327040000	M	30	M25-32	3230344	2016-05-04 10:04:42.0	120.3	30.41

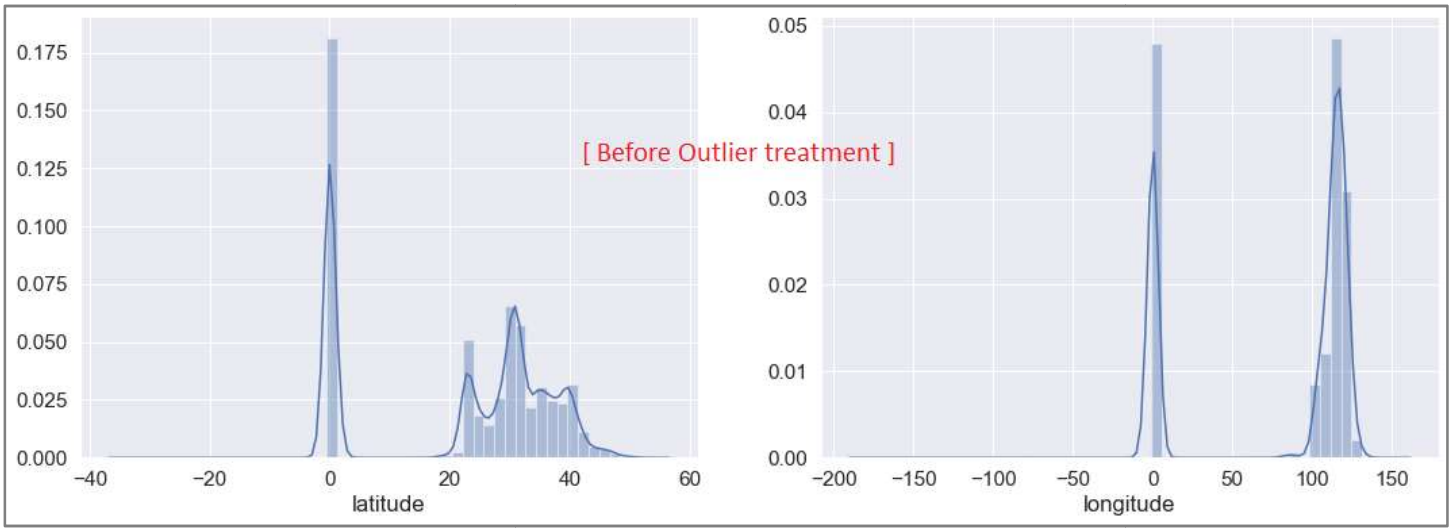
From the [app_data] dataset:

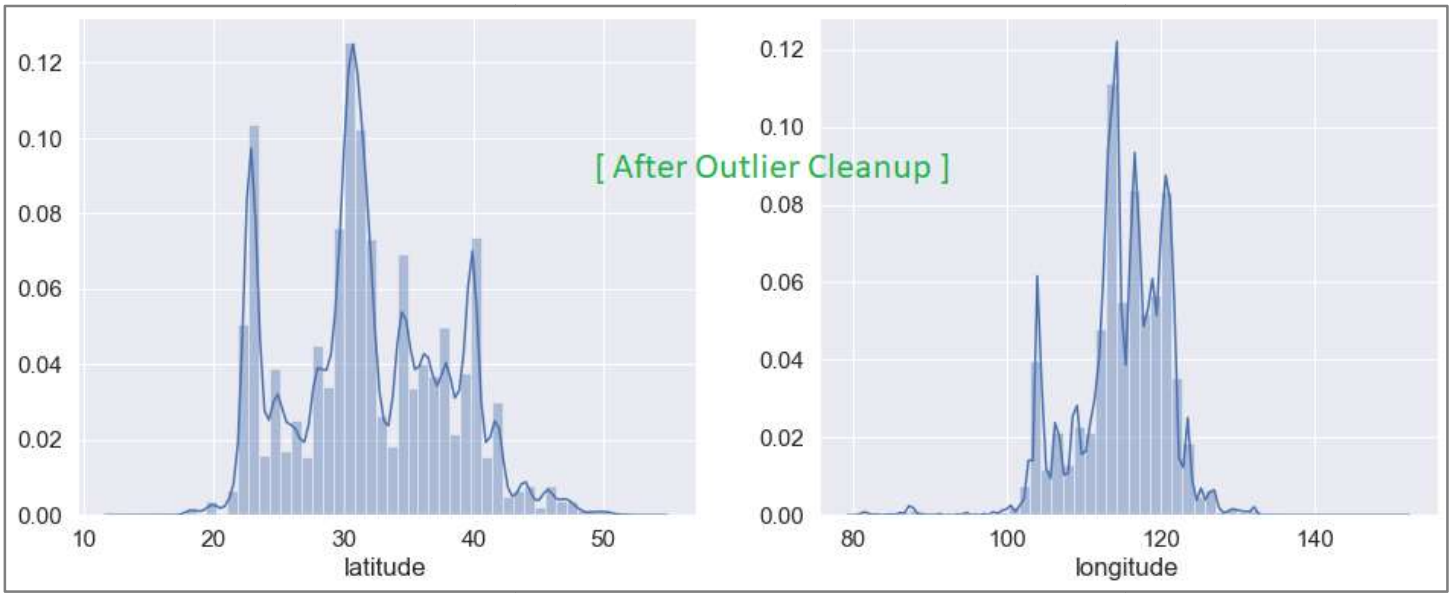
```
display(df_app_data_0)
df_app_data_0.info()
```

	event_id	app_id	is_active	category
0	3231904	-1000044012126765960	0	Casual puzzle categories
1	3231904	-1000044012126765960	0	Custom label
2	3231904	-1000044012126765960	0	game
3	3069897	-1000044012126765960	0	Casual puzzle categories
4	3069897	-1000044012126765960	0	Custom label

2. List of data cleaning techniques applied such as missing value treatment, etc.

- We replaced the negative values with ‘NaN’. From the summary stats and distribution plots, we see a significant number of latitude and longitude values fall around zero and need to be cleaned up along with other negative value outliers.





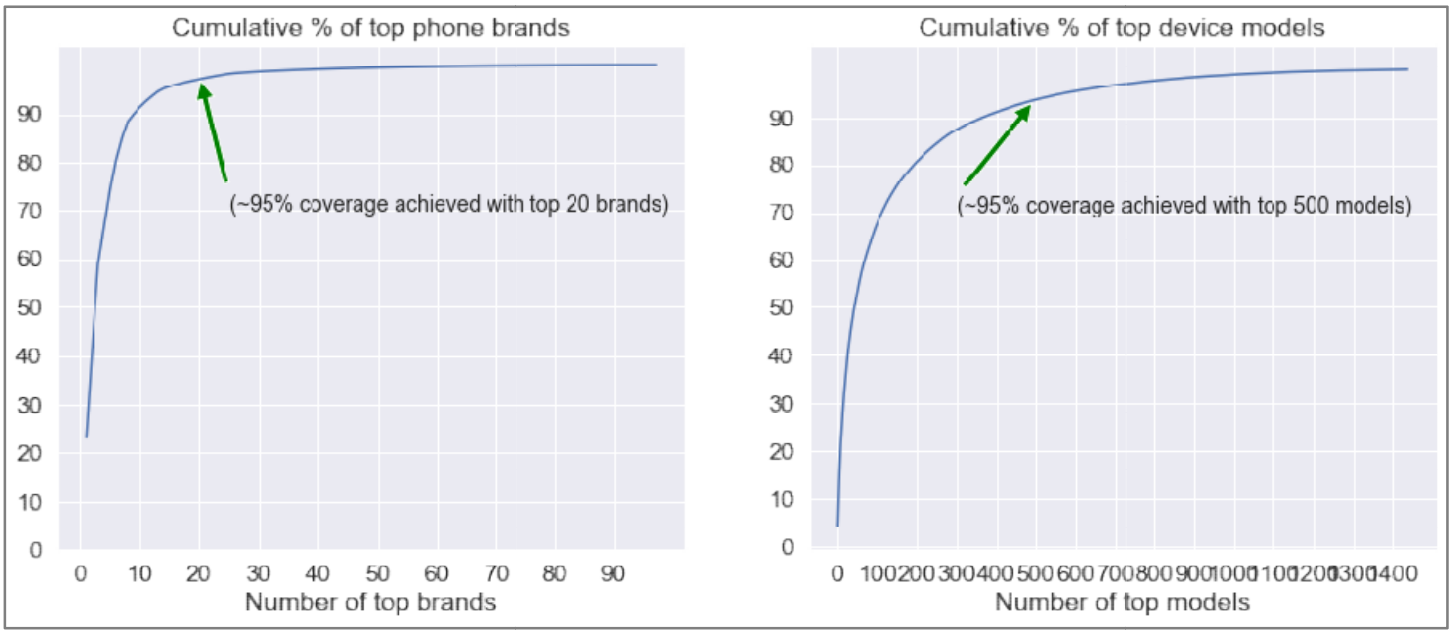
- Checked the missing values in all the datasets.

3. Feature engineering techniques that were used along with proper reasoning to support why the technique was used

Device Brand dataset:

- It is important to fetch hour and day of the week information to perform the event distribution analysis for the data. We will first derive new features 'hour' and 'dayofweek' from event_time
- We reduced the cardinality by mapping the low-count brands and models to a generic 'others' category, as we can see that the 95% coverage marks can be achieved using just the top 20 (out of 97) phone brands and top 500 (out of 1438) device models

- We will also drop the 4-binned 'age_group' column since there will be different age group bins in subsequent tasks



Event Data set:

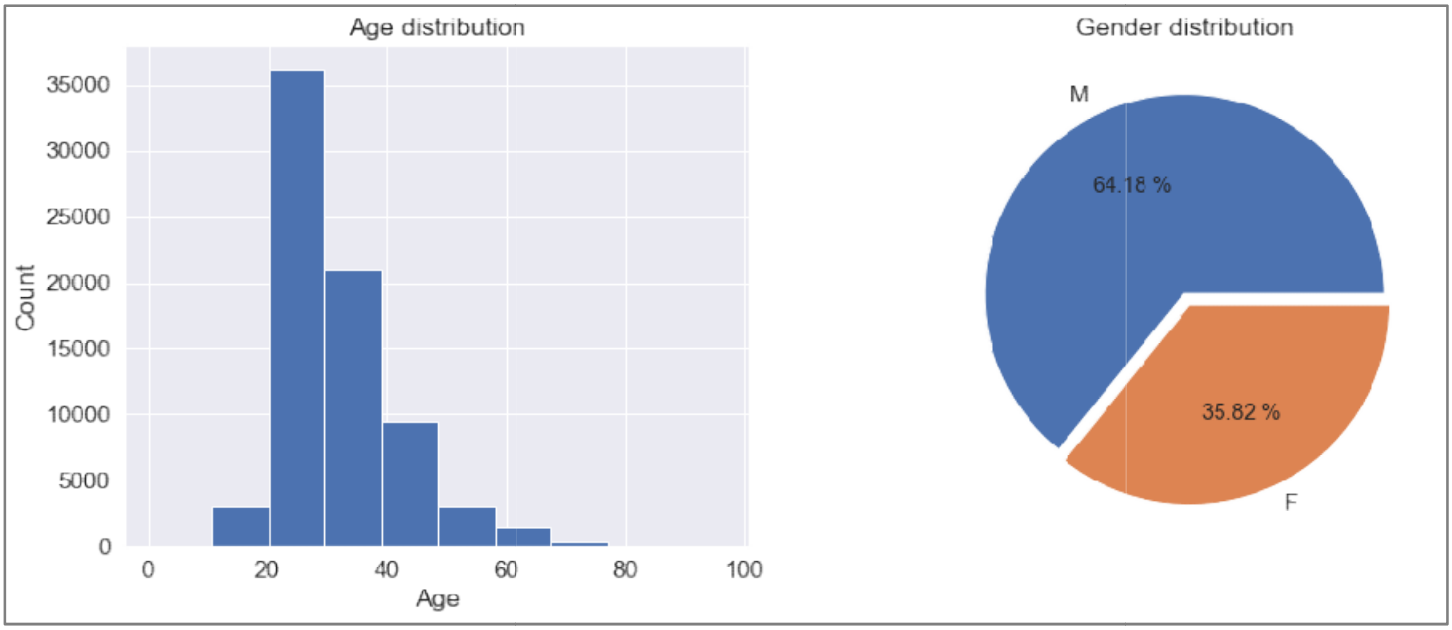
- We will re-bin the age column into 3 bins from 4 earlier as required for the analysis.
- Create a feature called Average Events, which can give an estimate of how long the users' mobile phones are active

App Data set:

- We did string-join the 'category' values and convert string values to a cleaned set of key words. Applied the join, lower functions and removed all punctuations.
- List all the categories that have more than 1lakh events to ‘super categories’ and use one-hot encoding to for all the super categories

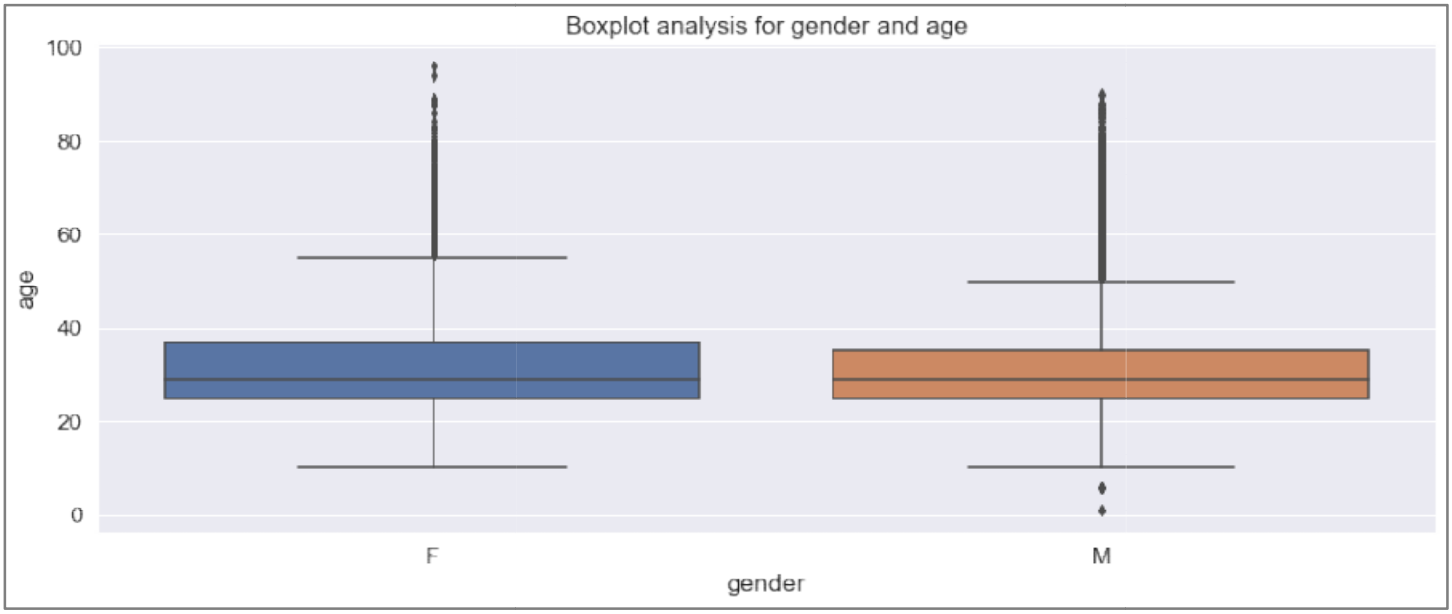
4. Outputs to the various EDA and Visualization codes along with the corresponding results and the insights gathered from each EDA and visualization

1. Plot appropriate graphs representing the distribution of age and gender in the data set [univariate]



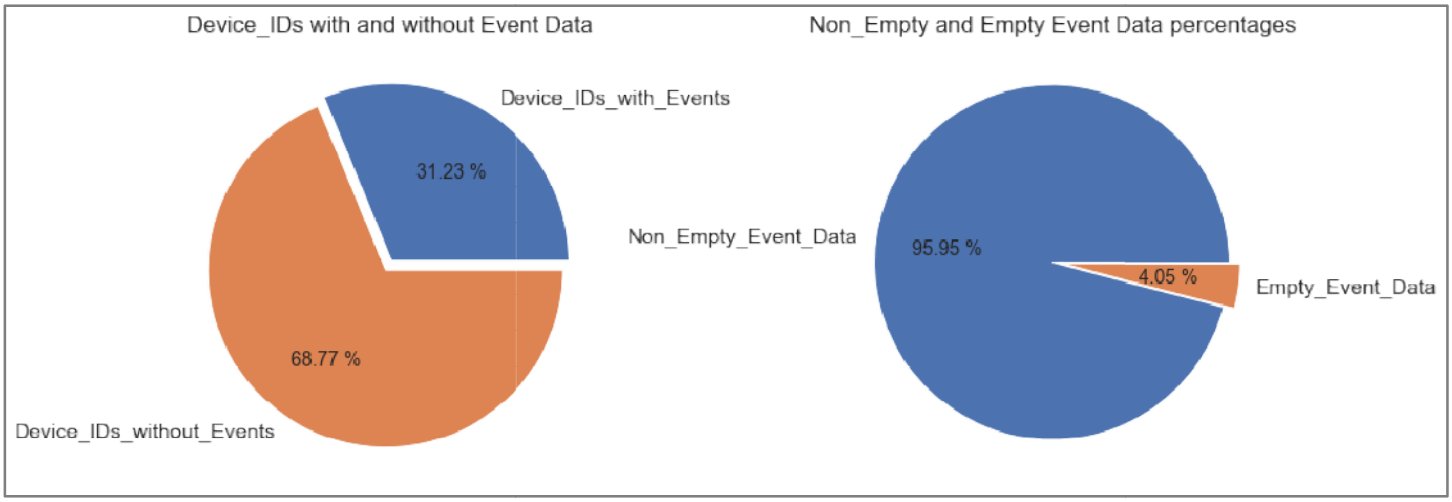
From the Age distribution we can see that the dominant ranges are 20-30 & 30-40 years with much lower counts in the extreme ranges. From the Gender distribution we see a dominance of Males at 64% compared to Females at 36%.

2. Boxplot analysis for gender and age [bivariate]



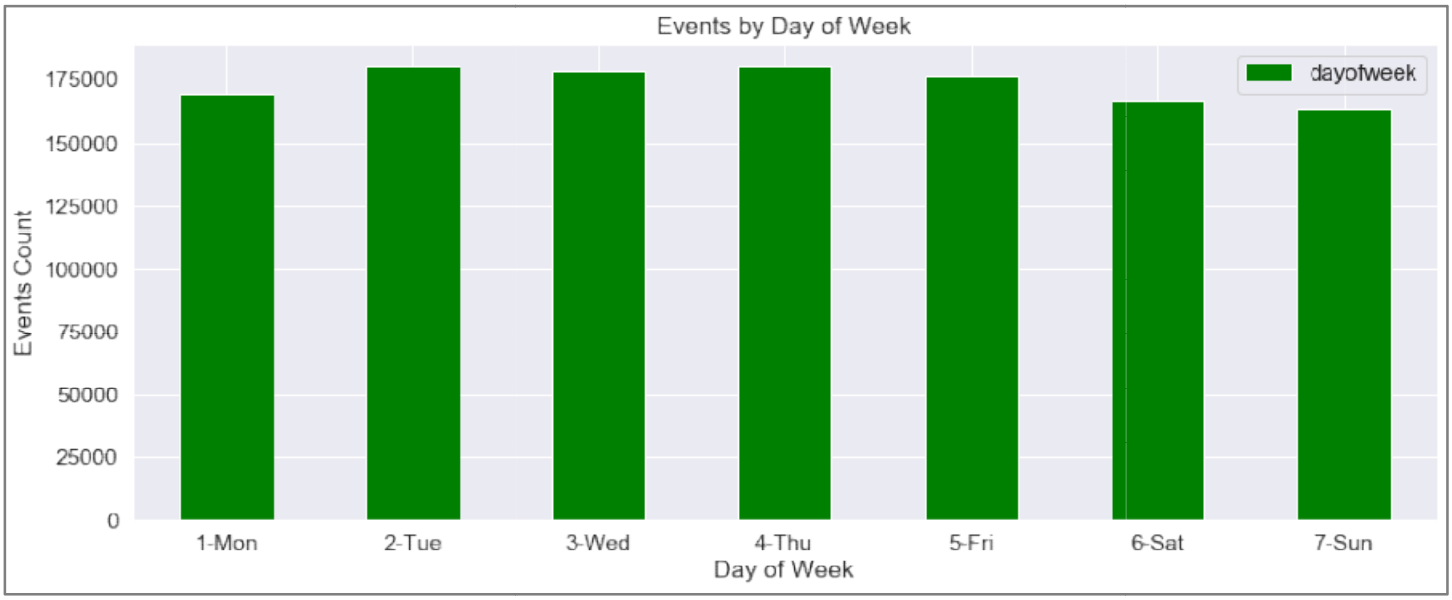
From the Boxplot distribution, we can see that the Female and Male Age distributions are very similar around the lower quartile and median ranges. However, the Female age distribution is higher compared to Male for the upper quartile and outliers beyond that, while the Male distribution shows a few outliers under the lower quartile.

3. Plot the percentage of the device_ids with and without event data



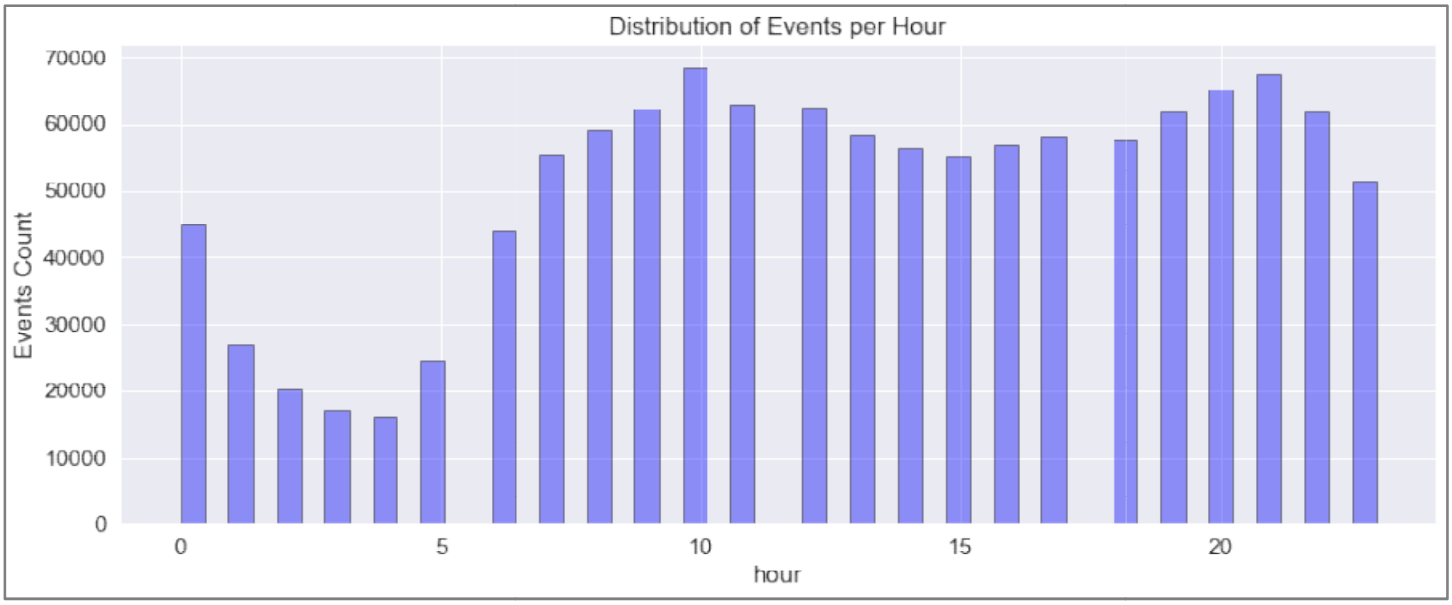
We can see that only 31% of the Device_IDs have event data. However, this event data makes up close to 96% of the df_events size with the null events making up just 4% of the dataset size.

4. Plot a graph representing the distribution of events over different days of a week



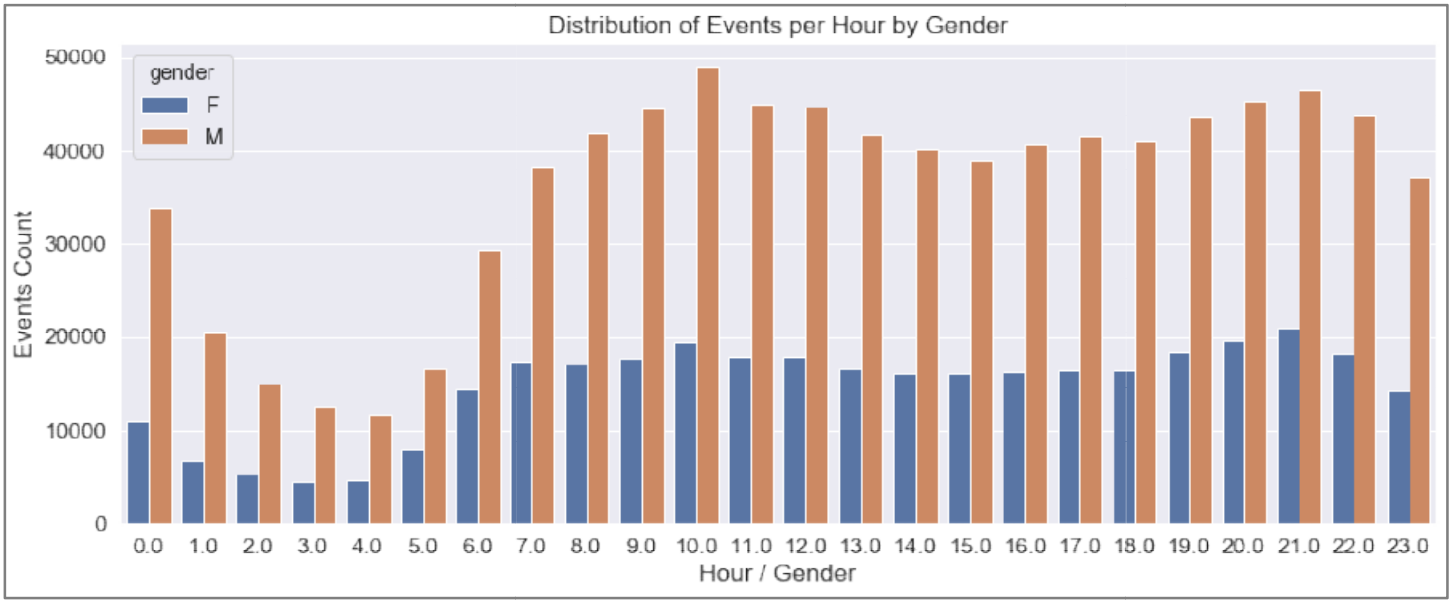
We can see that the day of week event counts are fairly consistent without any significant variation.

5. Plot a graph representing the distribution of events per hour



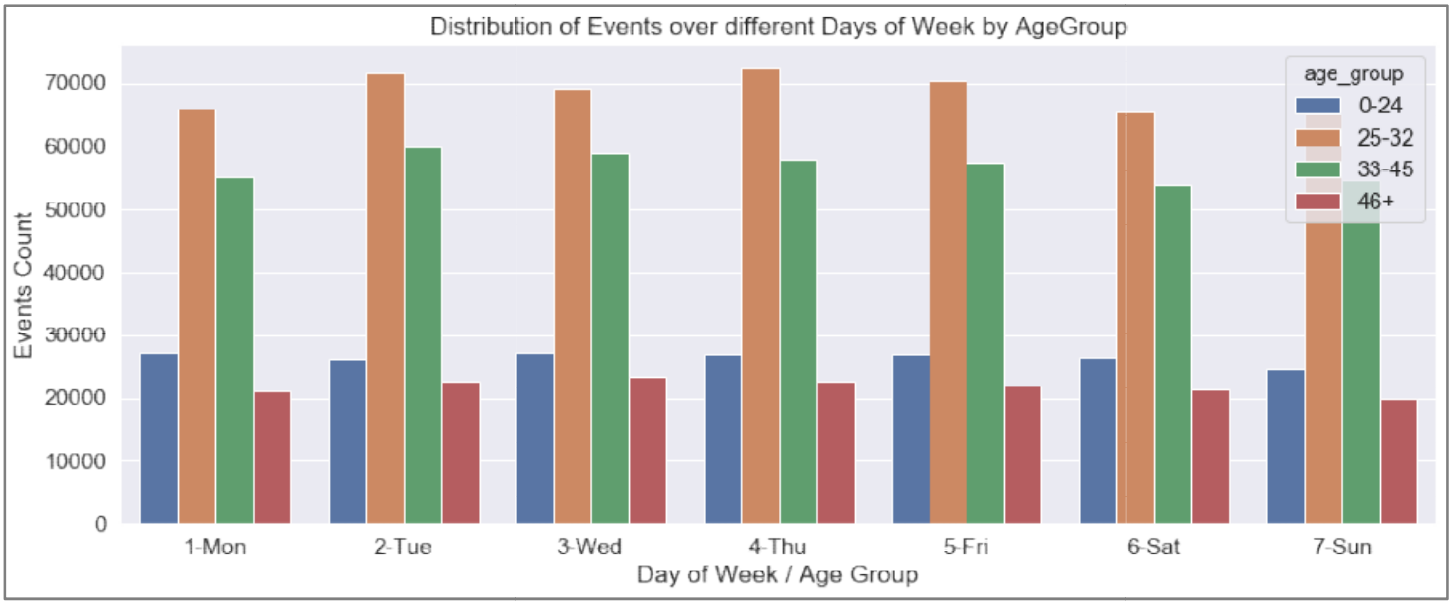
From the Events per Hour distribution, we can see that the event counts are low from post midnight to early morning hours and pick up over the daytime peaking around 10 AM and 9 PM.

6. Plot the difference in the distribution of events per hour for Male and Female consumers



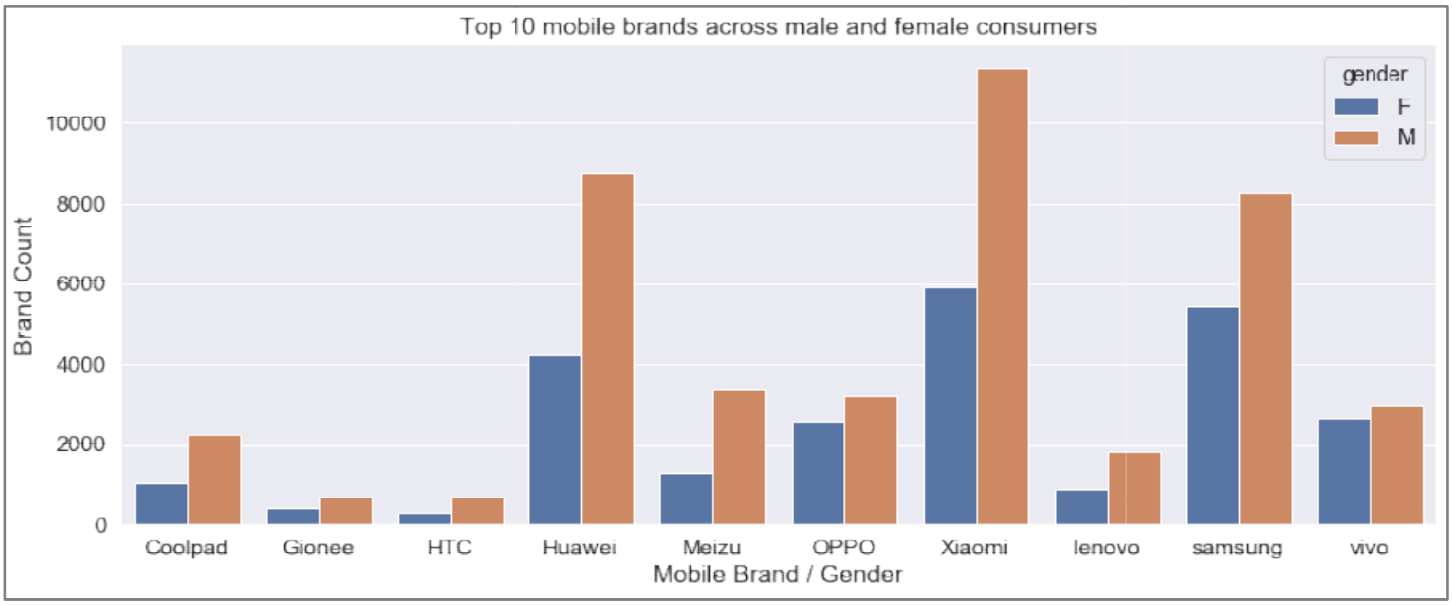
From the Events per Hour by Gender distribution, we can see that the event counts are low from post midnight to early morning hours and pick up over the daytime peaking around 10 AM and 9 PM. Also the counts are significantly higher for Males compared to Females, which indicates that higher event averages could be good predictors for Male gender.

7. Is there any difference in the distribution of Events for different Age Groups over different days of the week? [Consider the following age groups: 0–24, 25–32, 33–45, and 46+]



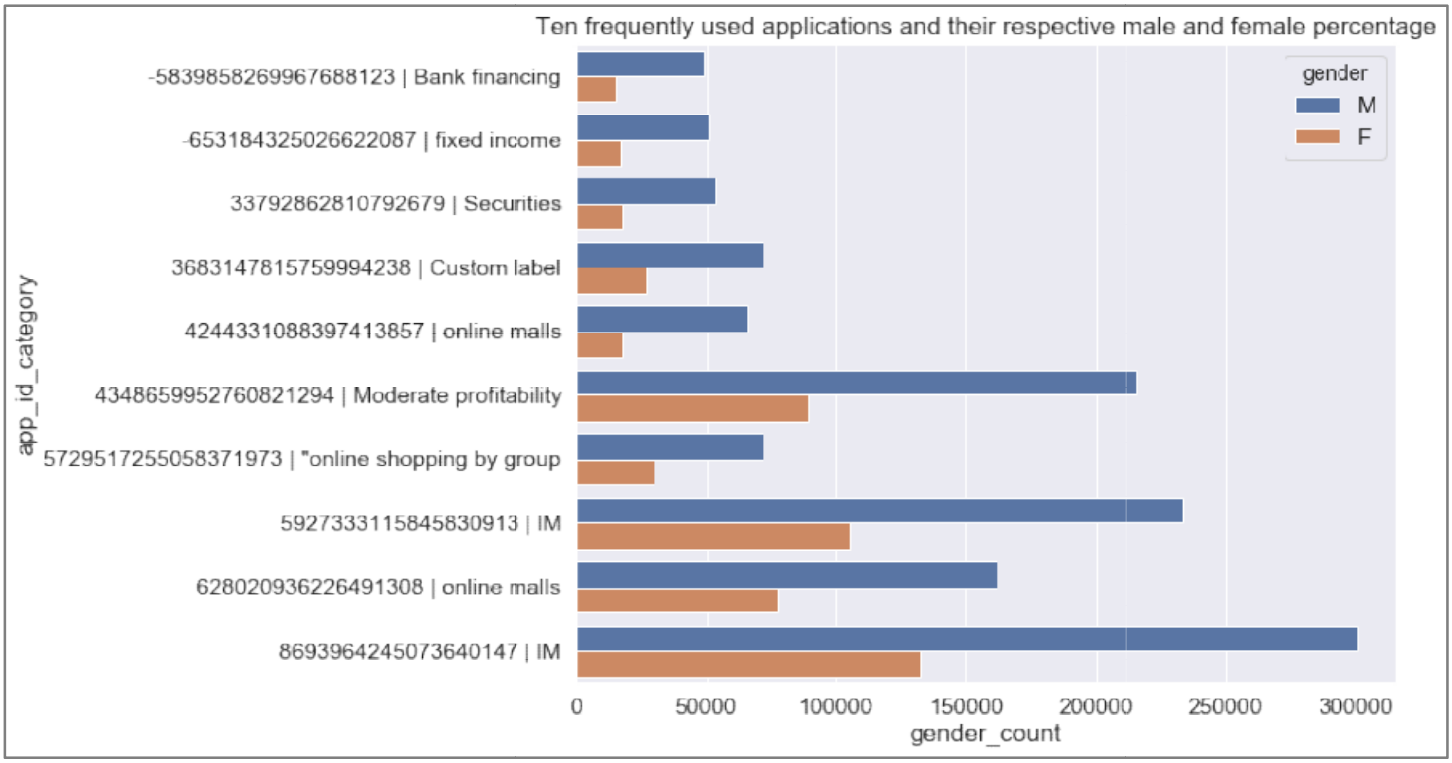
From the Events per Days of Week by Age Group distribution, we do not see any significant variation within each Age Group in terms of events count across the different Days of Week. We can also see that the event counts are low for the 0-24 and 46+ age groups as can be expected. This indicates that higher event averages could be good predictors for the middle Age Groups.

8. Stacked bar chart for the top 10 mobile brands across male and female consumers

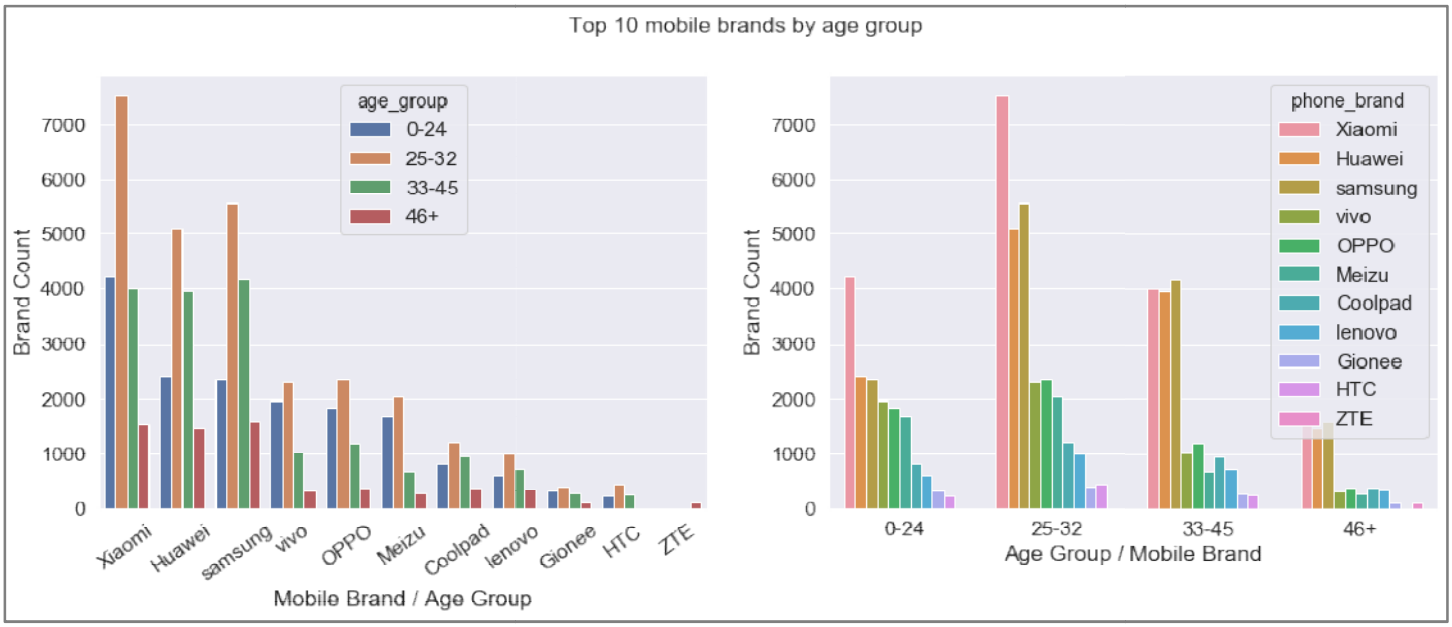


We can see that most of the top brands have a roughly 1:2 Female-to-Male ratio, with the exception of 'OPPP' and 'vivo' brands which have an almost balanced gender ratio.

9. Prepare a chart representing the ten frequently used applications and their respective male and female percentage



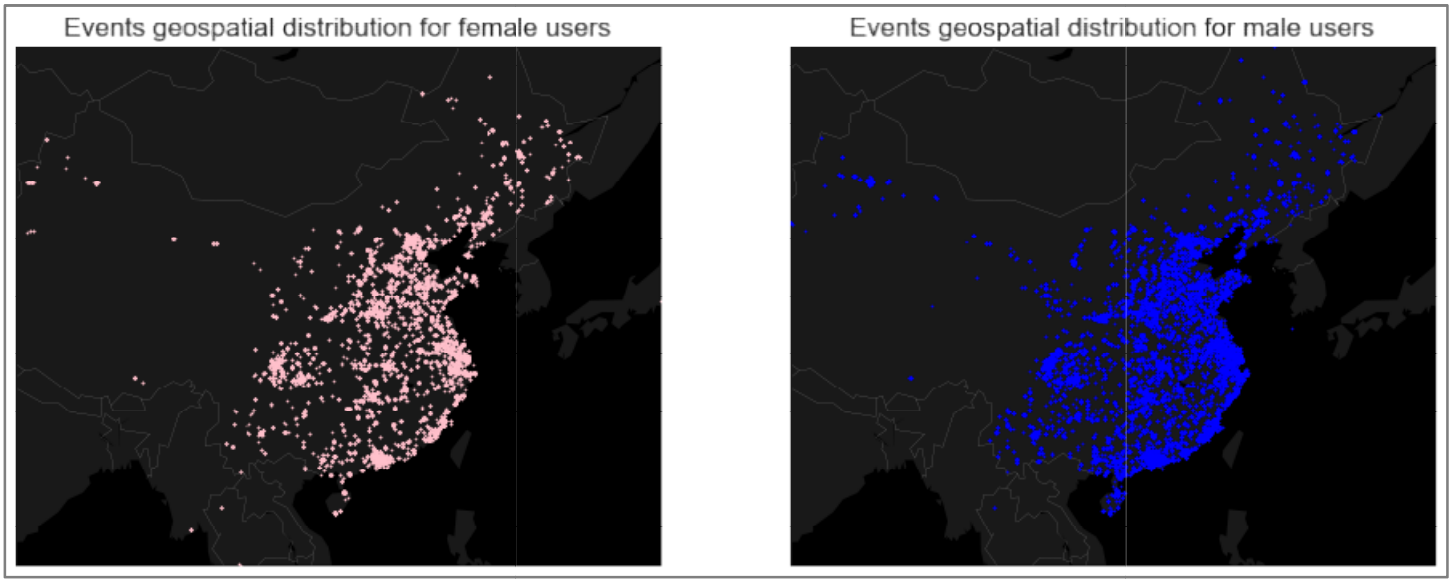
10. List the top 10 mobile phone brands bought by customers by age groups. [Consider the following age groups: 0–24, 25–32, 33–45, 46+]



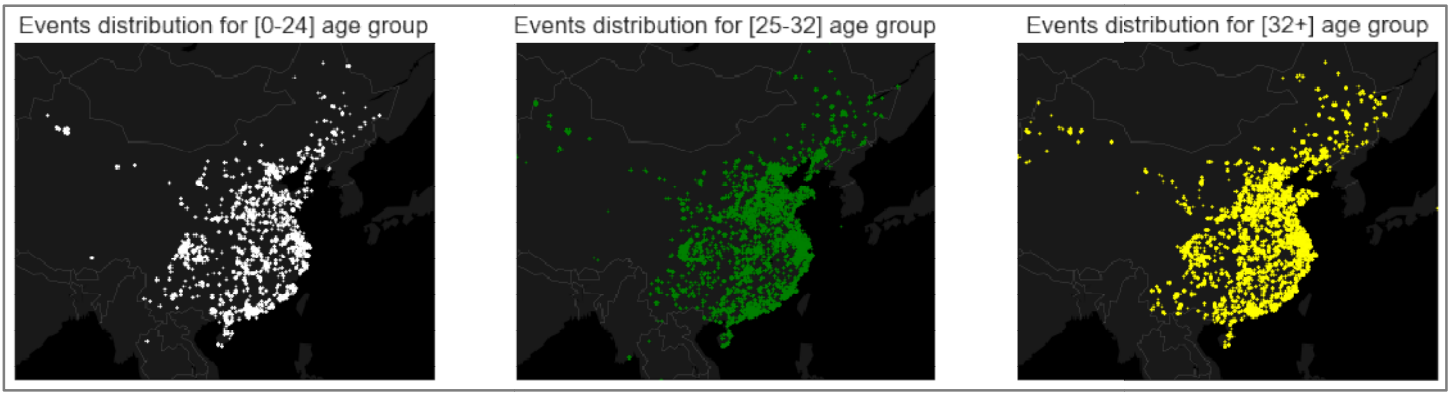
We can see that the same brands are popular across the age groups, with the exception of 'ZTE' brand as an outlier for the 46+ age group. This indicates that this specific mobile brand could be a good predictor for the 46+ age group.

5. Geospatial visualizations along with the insights gathered from this visualization

From the global geospatial distribution, we can see that almost all the events are from the China region. So for the subsequent visualization plots, we will zoom in to the China region bounded by the respective lat long min/max values.



We can see that the gender-wise events have a similar geospatial distribution.



We can see that the [0-24] and [25-32] age group events have a similar geospatial distribution, while the [32+] age group events have a more spread-out distribution at the outer geospatial regions. This indicates that any density-based clusters formed at the outer regions could be good predictors for the [32+] age group.

6. Results interpreting the clusters formed as part of DBSCAN Clustering and how the cluster information is being used

- DBSCAN clustering using the 'ball_tree' algorithm with 'haversine' metric on the unique lat long dataset. We see 127 clusters.
- Convert the cluster labels into data frame and map with our data set.
- The cluster ID's represent the latitude and longitude values. So we can drop latitude and longitude data as it will reduce the size of the dataset.

7. A brief summary of any additional subtask that was performed and may have improved the data cleaning and feature generation step

- We will optimize the memory usage by downcasting the column dtypes to int8/int16 based on the min/max values.

8. All the data preparation steps that were used before applying the ML algorithm

- We will first merge the event_data and app_data datasets
- Next we will merge the master and device_brand datasets
- We will load top 10 rows and take a look at the Capstone project provided train_test_split
- We will read the entire dataset with the right column names and dtypes. We will also ignore the 'gender', 'age' and 'group' column since those are not required for the join with master dataset
- Next we will merge the master and train_test_split datasets
- Finally we will persist the master dataset to local storage for retrieval and processing in subsequent tasks

9. Documentation of all the machine learning models that were built along with the respective parameters that were used (e.g., DBSCAN, XGBoost, Random Forest, GridSearchCV, etc.)

We will use 3 models each for the Scenario1 and Scenario2 datasets for gender and age predictions.

LogisticRegression model:

- We will opt for `Age Prediction as a Classification Problem` to enable a direct Age-related Campaigns mapping during Deployment
- Accordingly we will use (multi_class='auto') as the LogisticRegression variable selection for the Binary Gender prediction and (multi_class='multinomial') for the Multiclass Age prediction

XGBoost model with HyperParameterTuning (HPT):

- We will use (objective='binary:logistic', eval_metric='logloss') as the XGBClassifier variable selection for the Binary Gender prediction and (objective='multi:softprob', eval_metric='mlogloss') for the Multiclass Age prediction
- The sample HPT param grid provided on the platform results in training time of over 50 hours. So for performance reasons, we will settle for a much smaller HPT param grid
- We will extract the XGBClassifier best parameters and best estimators identified by the GridSearch HyperParameterTuning

StackingCVClassifier model:

- We will use LogisticRegression and RandomForestClassifier as the Base Models, XGBClassifier as the Meta Learner, and StackingCVClassifier as the Cross-Validation Stacking Classifier
- We will also use (objective='binary:logistic', eval_metric='logloss') as the XGBClassifier variable selections for the Binary Gender prediction and (objective='multi:softprob', eval_metric='mlogloss') for the Multiclass Age prediction

10. The reason for using regression or classification for age prediction

- We will opt for Age Prediction as a Classification Problem to enable a direct Age-related-Campaigns mapping during Deployment.

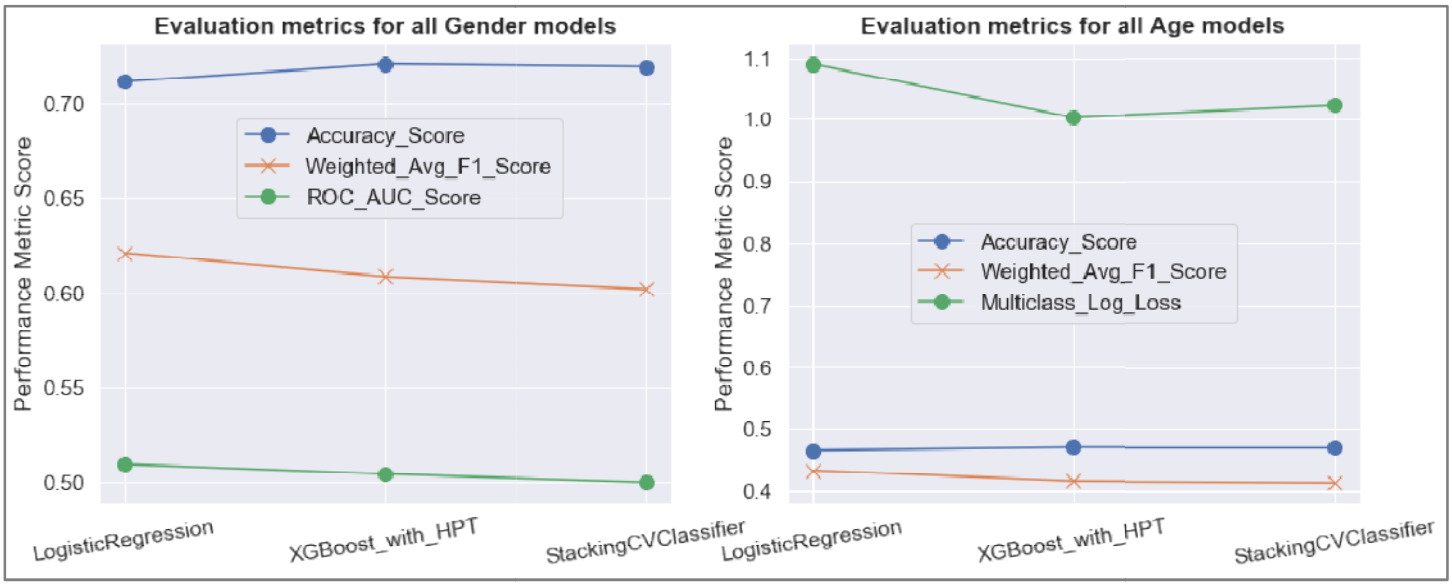
11. The outcomes of the evaluation metrics (results for both Scenario 1 and Scenario 2 must be shown separately)

Scenario1:

- For the Binary Gender prediction evaluation, we will calculate and log the AccuracyScore, Confusion Matrix (F1 Score, Precision, Recall), Weighted_Avg_F1_Score and ROC_AUC_Score metrics
- For the Multiclass Age prediction evaluation, we will calculate and log the AccuracyScore, Confusion Matrix (F1 Score, Precision, Recall), Weighted_Avg_F1_Score and Multiclass-Log-Loss metrics
- The final models to be selected for deployment are expected to display a balanced 'Female' / 'Male' Gender prediction, as well as a balanced [0-24] / [25-32] / [32+] Age group prediction, without any specific class bias

- Accordingly, we will opt for the "weighted avg f1-score" as the evaluation metric, since that corresponds to the harmonic balance between Precision and Recall of all the class labels
- **As per this metric, the winning model for the Scenario1 dataset is LogisticRegression for both Gender and Age predictions !!!**

Scenario1 – Gender Prediction Metrics			
Model	**Train accuracy**	**Test accuracy**	**f1-score weighted average**
LogisticRegression	0.7146	0.7113	**0.6209**
XGBoost	0.7076	0.7208	0.6085
StackingCVClassifier	0.7060	0.7194	0.6020
Scenario1 – Age Prediction Metrics			
Model	**Train accuracy**	**Test accuracy**	**f1-score weighted average**
LogisticRegression	0.5075	0.4662	**0.4330**
XGBoost	0.5011	0.4701	0.4157
StackingCVClassifier	0.5000	0.4701	0.4136



Scenario2:

- For the Binary Gender prediction evaluation, we will calculate and log the AccuracyScore, Confusion Matrix (F1 Score, Precision, Recall), Weighted_Avg_F1_Score and ROC_AUC_Score metrics
- For the Multiclass Age prediction evaluation, we will calculate and log the AccuracyScore, Confusion Matrix (F1 Score, Precision, Recall), Weighted_Avg_F1_Score and Multiclass-Log-Loss metrics
- The final models to be selected for deployment are expected to display a balanced 'Female' / 'Male' Gender prediction, as well as a balanced [0-24] / [25-32] / [32+] Age group prediction, without any specific class bias
- Accordingly, we will opt for the "weighted avg f1-score" as the evaluation metric, since that corresponds to the harmonic balance between Precision and Recall of all the class labels

- **As per this metric, the winning model for the Scenario2 dataset is LogisticRegression for both Gender and Age predictions !!!**

Scenario2 – Gender Prediction Metrics			
Model	**Train accuracy**	**Test accuracy**	**f1-score weighted average**
LogisticRegression	0.6439	0.6377	**0.5386**
XGBoost	0.6417	0.6403	0.5156
StackingCVClassifier	0.6439	0.6381	0.5334
Scenario2 – Age Prediction scores			
Model	**Train accuracy**	**Test accuracy**	**f1-score weighted average**
LogisticRegression	0.4407	0.4237	**0.3877**
XGBoost	0.4308	0.4176	0.3276
StackingCVClassifier	0.4348	0.4210	0.3453