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

   ```
   [Row(device_id=-7548291590301750000, gender='M', age=33,
   group_train='M32+', event_id='2369465', datetimestamp='2016-05-03
   15:55:35', latitude='33.9800000000', longitude='116.7900000000'),
   Row(device_id=-7548291590301750000, gender='M', age=33,
   group_train='M32+', event_id='1080869', datetimestamp='2016-05-03
   06:07:16', latitude='33.9800000000', longitude='116.7900000000'),
   Row(device_id=-7548291590301750000, gender='M', age=33,
   group_train='M32+', event_id='1079338', datetimestamp='2016-05-04
   03:28:02', latitude='33.9800000000', longitude='116.7900000000'),
   Row(device_id=-7548291590301750000, gender='M', age=33,
   group_train='M32+', event_id='1078881', datetimestamp='2016-05-04
   02:53:08', latitude='33.9800000000', longitude='116.7900000000'),
   Row(device_id=-7548291590301750000, gender='M', age=33,
   group_train='M32+', event_id='1068711', datetimestamp='2016-05-03
   15:59:35', latitude='33.9800000000', longitude='116.7900000000')]
   ```

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

   a. Null Values correction
   b. changing the values to right datatype,
   c. removing the “Null” string marked values.

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

   a) One-Hot Encoding: This technique is used to convert categorical variables into binary vectors. Each category is represented by a binary feature, where the presence or absence of the feature indicates the category. One-hot encoding is used when the categorical variable doesn't have an ordinal relationship, and we want to avoid introducing any arbitrary numerical relationship between categories.
   b) Label Encoding: Label encoding is used to transform categorical variables into numerical labels. Each category is assigned a unique integer value. Label encoding is suitable when the categorical variable has an ordinal

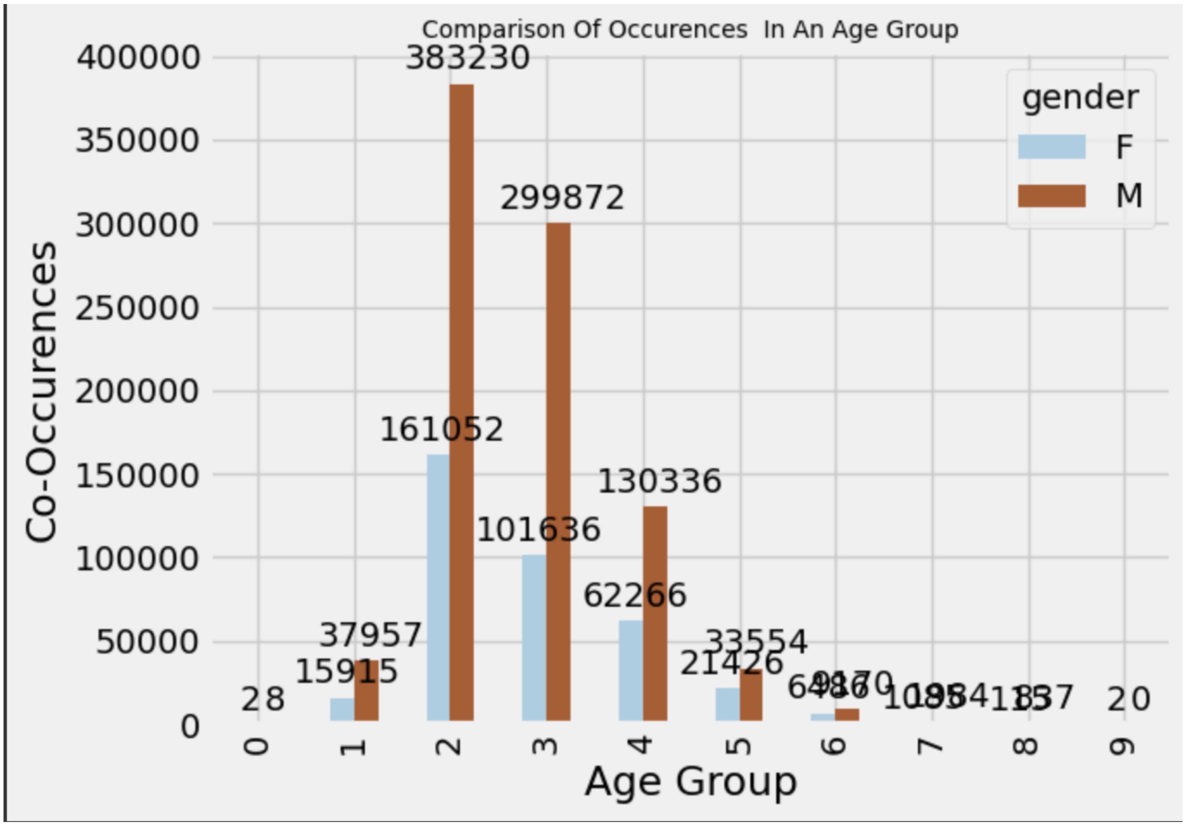
relationship, and the order of the categories matters. For example, "low," "medium," and "high" can be encoded as 0, 1, and 2, respectively.

c) Scaling: Scaling techniques like Standardization (Z-score normalization) or Min-Max Scaling are used to bring numerical features to a similar scale. This is important when features have different ranges or units. Scaling ensures that no particular feature dominates the learning process based on its magnitude alone, and it can improve the convergence of gradient-based models.

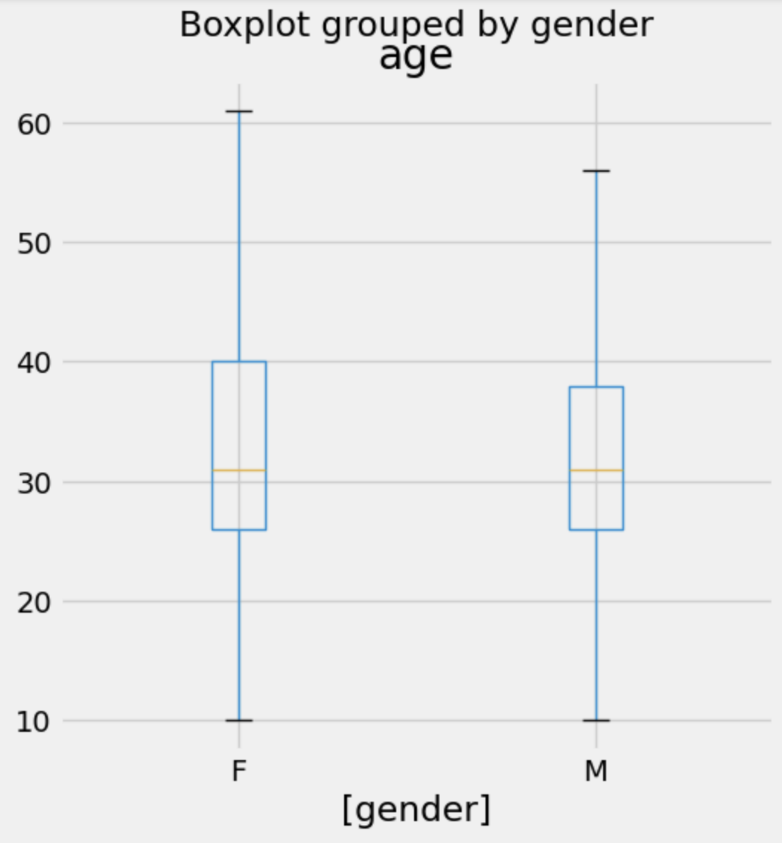
d) Polynomial Features: Polynomial feature generation involves creating new features by combining the existing ones through multiplication or exponentiation. This technique is useful when there is a nonlinear relationship between the features and the target variable. By introducing polynomial features, we can capture higher-order interactions and potentially improve the model's ability to fit complex patterns.

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

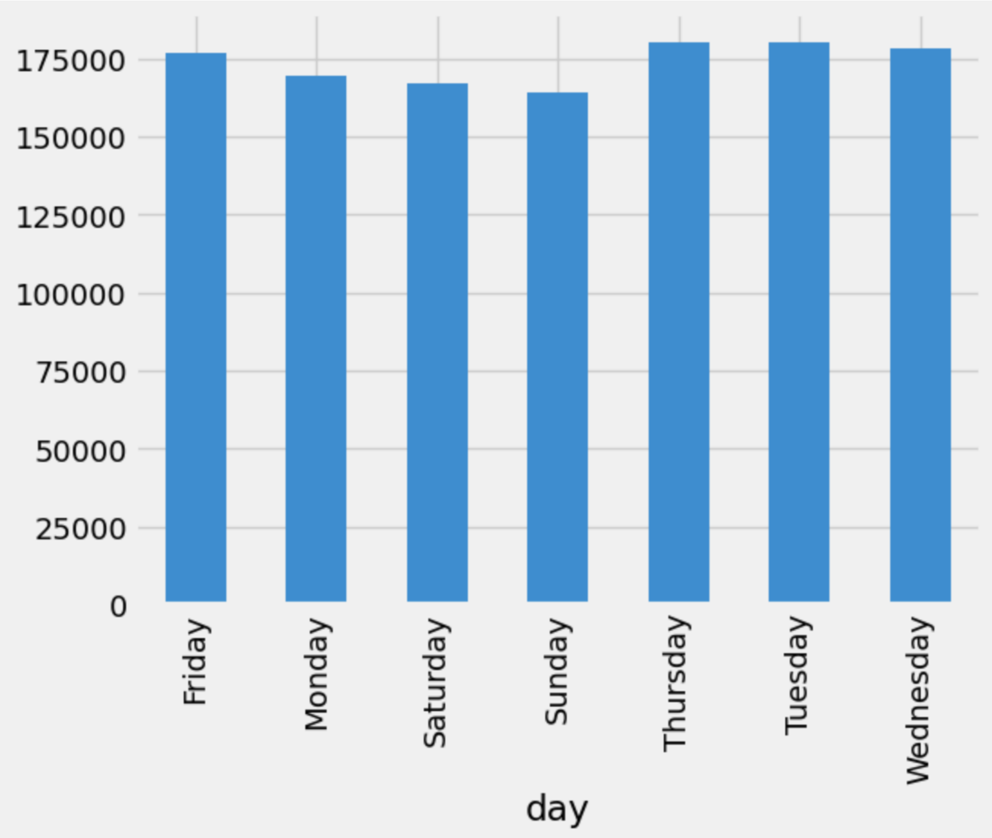
Age group & event co-occurrence graph



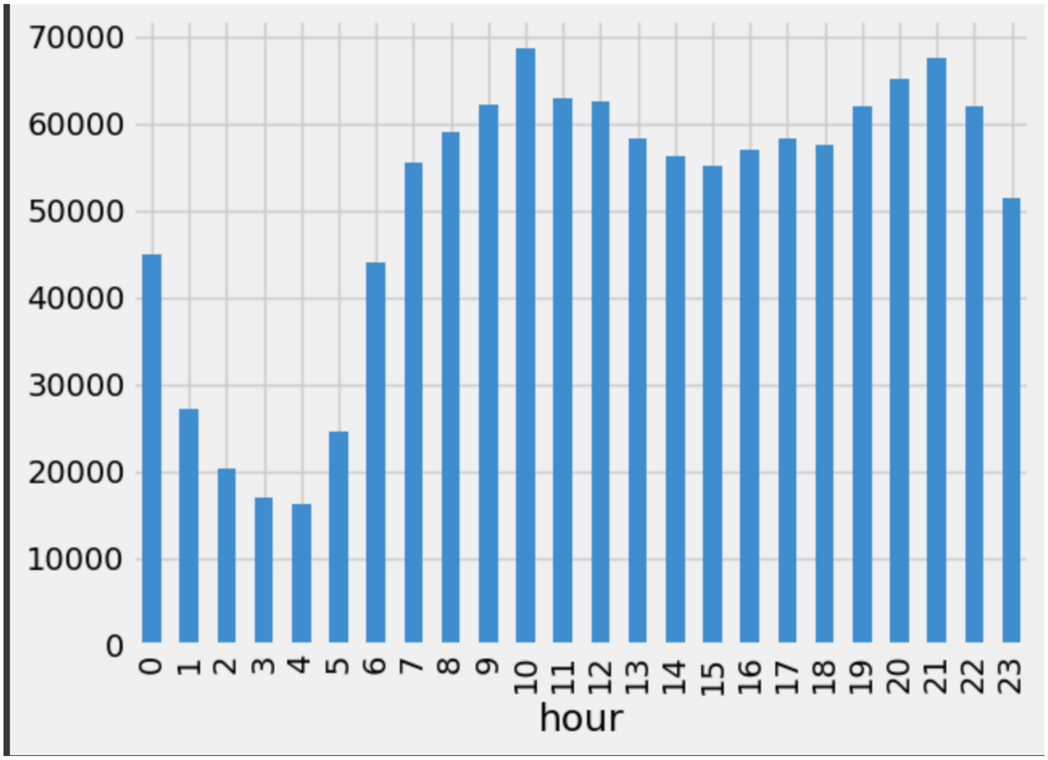
Box plot to see the stats



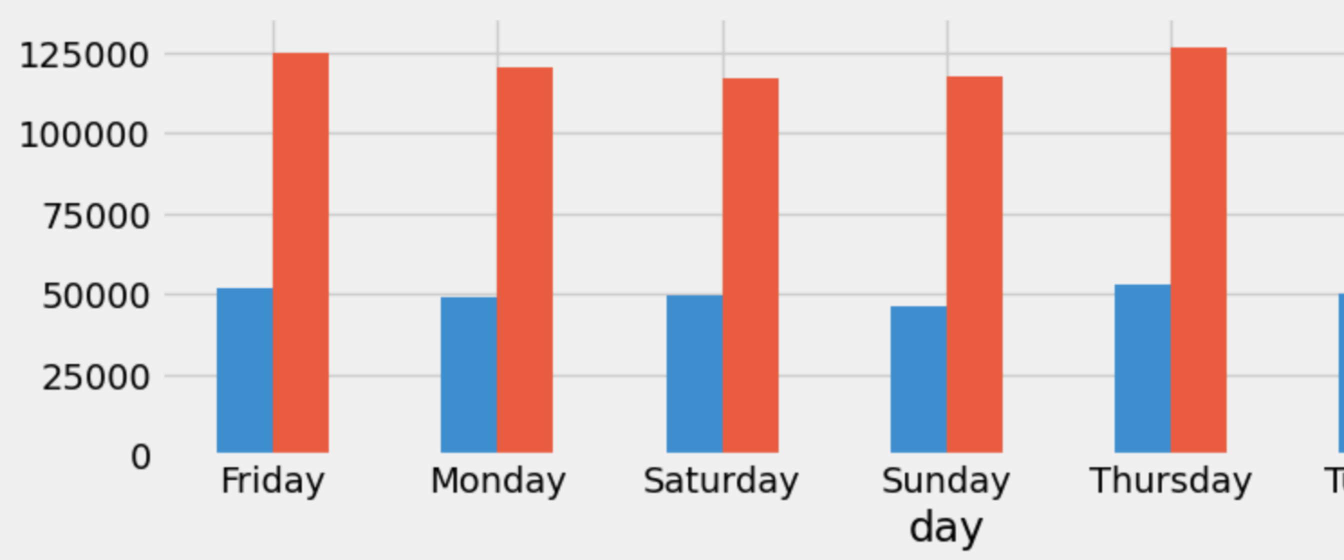
Day wise event bifercation



Hour wise event bifercation

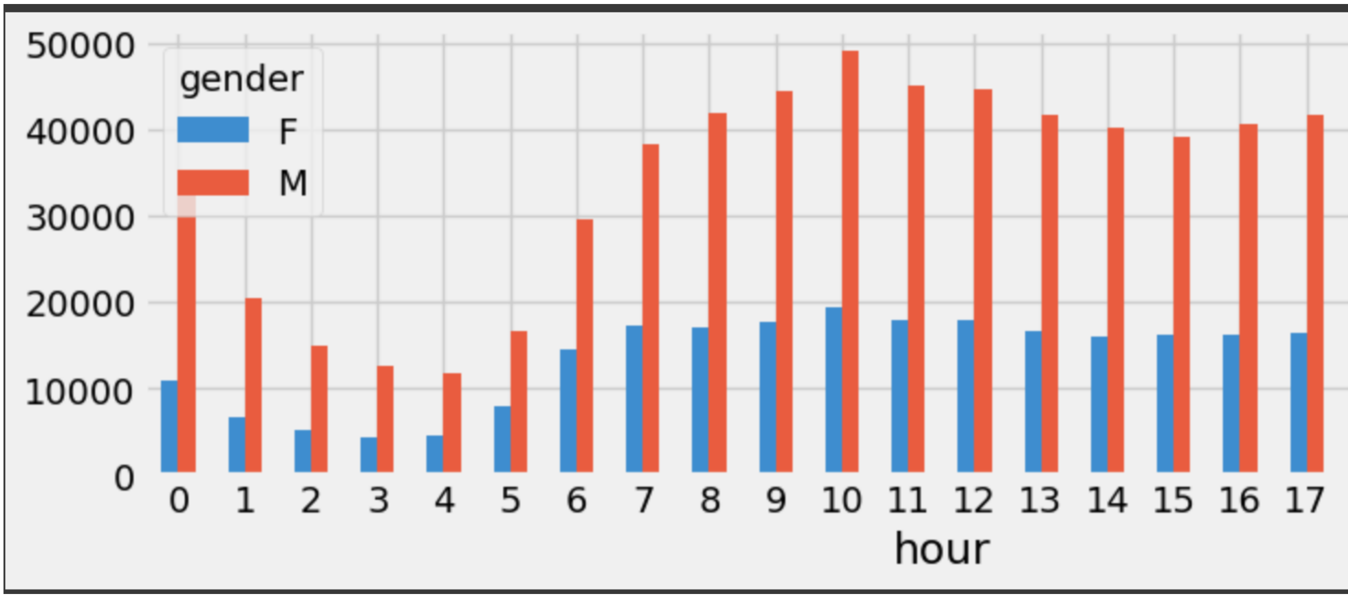


Gender based event bifurcation on day basis

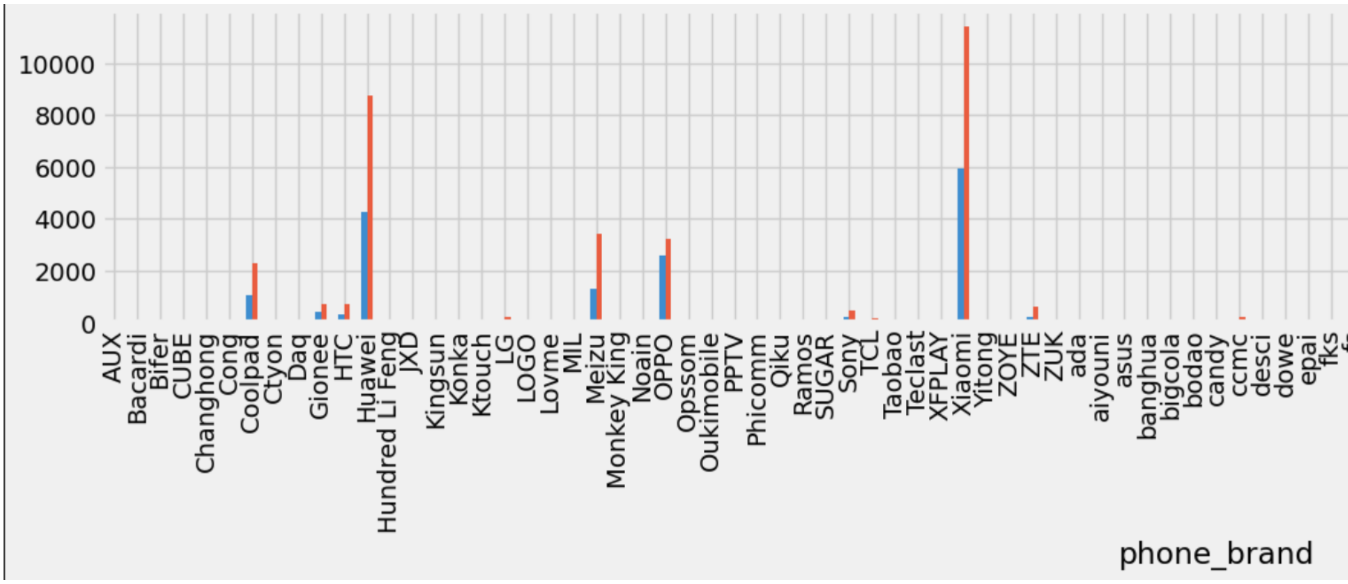


\

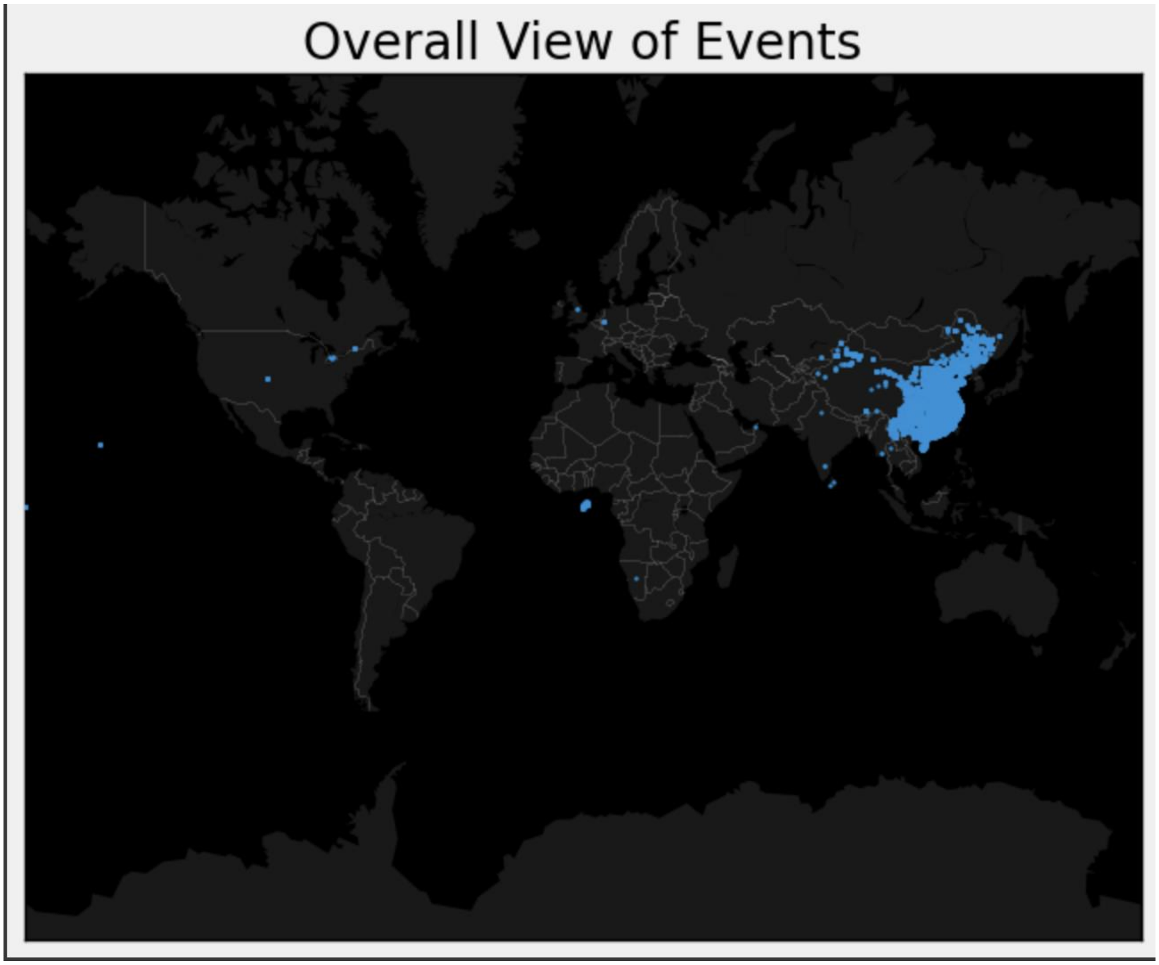
Gender based event bifurcation on hour basis

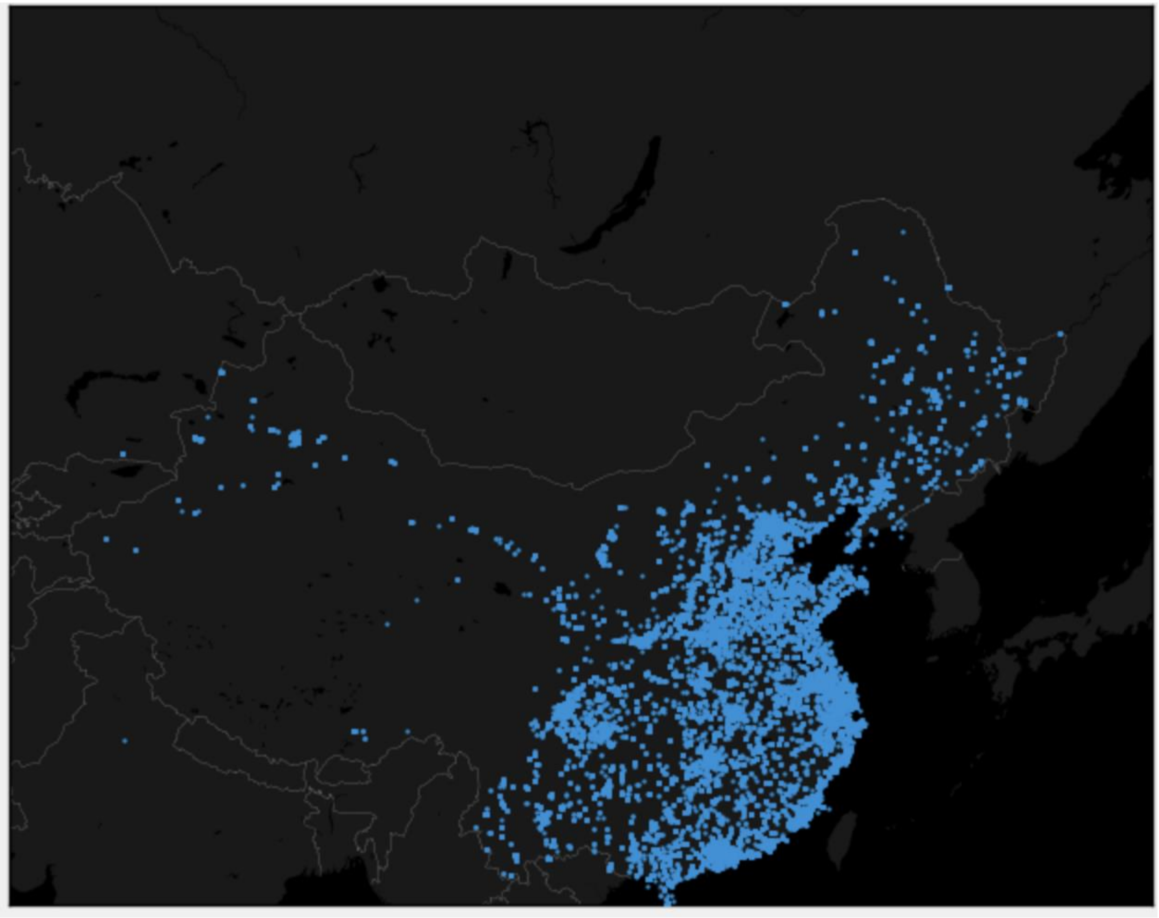


Most used phone brands based on gender

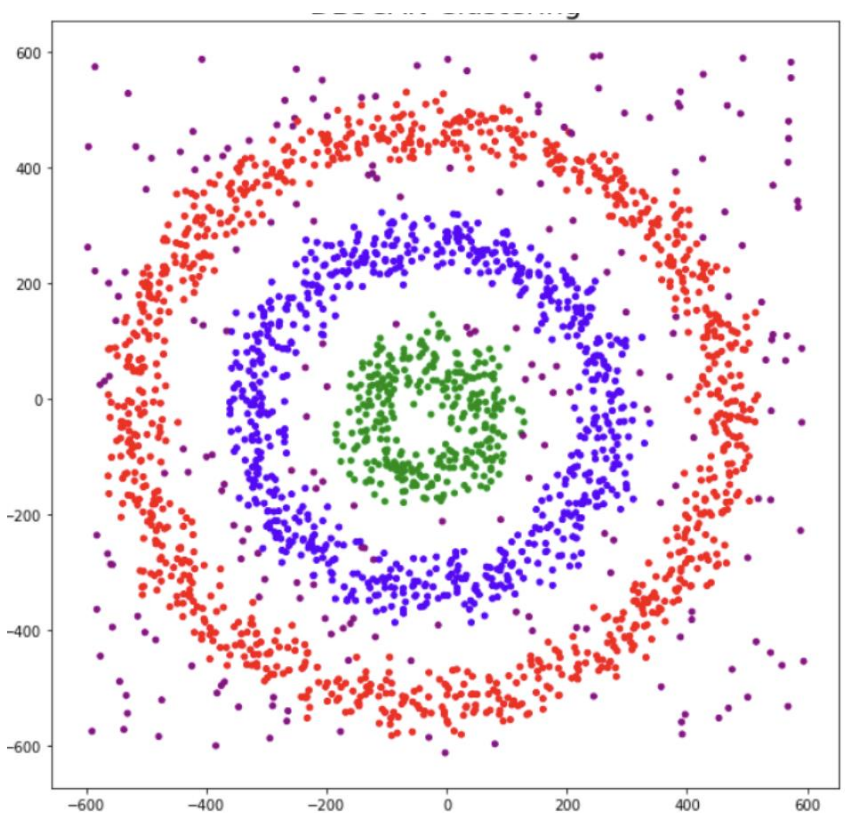


5. Geospatial visualisations along with the insights gathered from this visualisation


Overall View of Events



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



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

   Converting the latitude & longitude to float & categorizing it with DBscan

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

    1. Cleaning
    2. Scaling
    3. Clustering
    4. Encoding
    5. Splitting based on the test/ train flag

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.)

```
model = xgb.XGBClassifier(min_child_weight=1,
                    gamma=0.5,subsample=0.2,
                    max_depth=3,
                    n_estimators=100,
                    learning_rate=0.001)

for clf, label in zip([clf1, clf2, stacking_demo],
                 ['lr',
                  'Random Forest',
                  'StackingClassifier']):

   scores = model_selection.cross_val_score(clf, x_train, y_train, cv=3,
scoring='roc_auc')
   print("Accuracy: %0.2f (+/- %0.2f) [%s]" % (scores.mean(), scores.std(),
label))
```

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

The choice between regression and classification for age prediction depends on the specific context and requirements of the problem. Here are the reasons for using regression or classification for age prediction:
Regression:

1. Continuous Output: Age is typically considered a continuous variable since it represents a person's exact age in years, months, or even more precise units. Regression models are well-suited for predicting continuous variables, as they can provide output in the form of a specific numerical value.

2. Granularity: Regression allows for fine-grained predictions by considering the precise numerical relationship between the input features and the target variable. This can be valuable in scenarios where the distinction between different ages is important, such as predicting the age of an individual in months or days.
3. Confidence Intervals: Regression models can provide estimates with confidence intervals, indicating the range within which the predicted age is likely to fall. This can be useful in understanding the uncertainty associated with the predictions and assessing the model's reliability.
4. Training Data Availability: If the training data contains labeled examples with precise age values (e.g., 25 years, 32 years, etc.), regression models can directly learn from these labeled data points to make accurate predictions. This eliminates the need for additional preprocessing steps like binning or converting age into categorical labels.

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

    Gender Prediction: Accuracy: 0.708375778155065

    Age Prediction: Accuracy: 0.9866440294284098