Final Project CS82 : Predicting Facebook Check-ins

Submitted by : Mohammed Raashid Kheruwala & Oleg Smolanko

The project aims ***to predict which place a person would like to check-in to***. Facebook is the number one social media website of the world with approximately 2 billion users worldwide. When a Facebook user wants to share information about their current location, they can "check in" to a local business, restaurant, or practically any public place. When people do this, their update and location appears in their News Feed for all their friends to see. Hence if someone uses the Check-in function when they visit the business, they are telling their friends that they are currently at and enjoying the services or products. This results in free advertising, increased brand awareness and is essentially an endorsement from the user. Checking in to a business on Facebook is a very easy thing to do for anyone who has a smartphone.

The data was provided by Facebook itself, where it created an artificial world consisting of more than 100,000 places located in a 10 km by 10 km square. *For a given set of coordinates, it returns a ranked list of the most likely places.* The data was modified to resemble the location signals coming from mobile devices.

The training data consists of approximately 29 million observations where the **location** (x,y), **accuracy** and **timestamp** is given along with the target variable, the check-in location. The test data consists of 8.6 million observations where the check-in location was predicted based on the location, accuracy and timestamp. The train and test data were split based on time. Also, *worth noting is the fact that there is no concept of a person in the dataset and all the observations are events, not people*. A ranked list of the top three most likely places was calculated for all test records.

Data-set :

- There are approximately 29 million observations for training data along with the target variable, check-in location.
- Similarly, there are approximately 8.6 million observations for test data.
- Initial feature set : *x, y, time, accuracy*

Exploratory Data Analysis

Upon initial exploratory (data) analysis of the location of the train check-ins revealed quite an interesting pattern between the variation in x and y and it appeared that there was more variation in x than y which could be related to the streets of the simulated world. However, the difference in variation between x and y is different for all the places in the artificial world and there is no obvious pattern in the relationship.

The way ranking worked was ranking the actual place as the most likely candidate got a score of 1, the second most likely got a score of 0.5 and a third place got a score of 0.33. A total score is calculated by taking the mean of observation scores.

Essentially, based on initial analysis, the description of the data-set is given below:

- **row_id** the unique number of rows in the data frame.
- **x** is bounded between [0, 10] on the x-axis on the 10-km square.
- Similarly, to x, y is bounded between [0,10] in y-axis.
- **Accuracy** has the smallest value, 1.00 and the biggest value is 1033.00 with no units.
- **Time** has no units but further analysis will reveal the unit of the time column.
- **place_id** is a unique identifier of assumed places. There are approximately 108390 unique values for place_id and it is uniformly distributed which means there is no obvious leakage.

Understanding the features : Let us analyze the major features, Accuracy and Time.

Accuracy

- Accuracy was hardest input to interpret. Initially, it was expected that accuracy would be correlated with the variation in x and y but the pattern was not obvious. Many plots were generated to understand accuracy.
- As we know from the initial analysis, the accuracy isn't exactly defined in a way that the minimum value of it is 1.00 and maximum is 1022. However, we could think of it a few ways,
  - Error on some arbitrary scale. But this seems unlikely, since the max is > 1,000.
  - Error on the same scale as x and y. Now, this could be an estimated radius (with the x and y values as the center); either normal or squared.
- Binning the accuracy in approximately equal sizes made the signal visible. The signal particularly was accurate in the 45-84 range.

- The accuracy distribution seemed to be a mixed distribution with three peaks which changes over time.
- It is likely to be related to three different mobile connection types (GPS, Wi-Fi and cellular).
- The places showed different accuracy patterns and features that were added to indicate the relative accuracy group densities.
- It was also discovered that the location is related to the three accuracy groups for many places. This pattern was captured by the addition of additional features for the different accuracy groups. A natural extension to the nearest neighbor calculation would incorporate the accuracy group and it could however be tried as a further enhancement.

Time

- The second largest share of the features belonged to the time features.
- The activities of shops/places usually peak at certain hours in a given day, or in certain days in a given week.
- So, the density of check-ins over time showed periodicity to some extent. To identify the unit of time, we had explored and tried out various methods, one of them was using Fourier Transform, and it was concluded that data is in “minutes”.
- We have approximately 555 days in train data-set and approximately 140 days in test data-set.
- We have converted all time period counts to period density counts in order to handle the two drops in the time frequency.
- Periods include 27 two-week and 27 1-week periods prior to the end of the summary data.
- From this, we can look at day of week to identify trends (weekends), day (to find longer term seasonality).

In the project, we had around 30 million (simulated) check-ins on Facebook in a 10km by 10km grid. The tricky part is that there were around 100k different classes(place_id’s) so most supervised learning techniques wouldn’t work on the entire dataset. But we ran in total 13 different models and then took an ensemble of it.

Modelling : Model & Parameter Selection

- As the size of the data-set is very large, the model and hyper-parameter selection was done on 1/100th of the total data-set
- It actually is the data in the range of : 5<x<6 and 5<y<6.
- Then the last 171360 minutes are used as the validation set. By doing this, we are able to know the performance of a particular model in 20 mintues.

HyperParameter Selections

Following are the hyper-parameters that were selected,

- n_cell_x (int) : number of cells divided in x for test and validation set
- n_cell_y (int) : number of cells divided in y for test and validation set
- x_border (float) : border on the edge on x-axis of the cell for larger range of training set
- y_border (float) : border on the edge on y-axis of the cell for larger range of training set
- th (int) : cutoff for low frequency labels

Ensemble

- Each of the single models will predict the top 10 place_id probabilities for each row_id. These will be stored in the csv files as (row_id, place_id, probability).
- The output file for a validation set is about 3.3 MB and 351 for the test set.
- We ensemble the probabilities of different models using default-dict and output the top 3 place-ids as prediction.
- We ensemble the validation set first to see which combination of models has better performance.
- After that, we ran the ensemble on the whole data set with this combination.

The idea of finding the combination is :

- Trying to increase the diversity of the model. This includes different number of cells and different models.
- Then listing model candidates in different category and sort them by local map3(Mean Average Precision @3) score.
- Doing greedy search to find the combination with highest local validation score.
- Then running the ensemble on the whole dataset.

Models used in the Ensemble are :

- sklearn.neighbors.KNeighborsClassifier/knn_ps2_n1
- sklearn.ensemble.RandomForestClassifier/rf_opt1,
- sklearn.ensemble.ExtraTreesClassifier/et_opt1
- XGBoost/xgb0_n4
- XGBoost/xgb0_n2
- XGBoost/xgb0_n4_th13_border5
- XGBoost/xgb150opt
- XGBoost/xgb150opt2
- sklearn.naive_bayes.GaussianNB/nb_xyat_weight1_n9
- sklearn.neighbors.KernelDensity/kde_opt1
- scipy.stats.gaussian_kde/kde_opt2c
- scipy.stats.gaussian_kde/kde_opt3c
- KDE-for-SciPy/kde_opt4

  The prediction is a soft-voting of these 13 single models, including k-nearest neighbors, random forest, extra-trees, gradient boosting trees, naive bayes, kernel density estimation.

Conclusion

- The insights and the results are very interesting. The best XGBoost model has only max_depth = 3 which is kind of unusual. This kind of indicates that features are almost independent to each other in the dataset.
- Naive Bayes is actually the first assumption of the data distribution. When using sklearn.naive_bayes.GaussianNB, there was an understanding that the distribution of each feature in cell is unimodal.
- But when we did a simple visualization of the x, y for given place_id that assumption proved to be not very good.
- We discovered that to improve the Naive Bayes, we could use kernel density estimation (KDE) rather than one single gaussian to get the probability density function.
- The performance of the KDE was determined by the kernel and kernel parameters(bandwidth for Gaussian kernel). Sklearn.neighbors.KernelDensity provides one with plenty of kernels to choose form. The hyperparameter selection code indicated that the guassian and exponential are optimal choice while tophat, linear, cosine works badly.

- Sklearn does not provide the rule of thumb estimation of the bandwidth, so the bandwidth was selected by cross-validation. We also tried the scipy.stats.gaussian_kde because it provided two common estimation techniques, “scott” and “silverman”. They can give fast estimation but are designed for unimodal distribution.
- We finally get Ensemble model csv files, which we are sharing it via Google Drive link. If one can see one of the output files, “ensemble_train.csv” , for a given row_id there are three most likely ranked place_ids .
- As this was a past kaggle competition, the accuracy got was in top 50.