Predicting your footsteps on Facebook

Manav Verma
mverma4@ncsu.edu

Sameer Sharma
ssharm20@ncsu.edu

Shivam Gulati
sgulati2@ncsu.edu

Abstract

Facebook check-in activity has become a source of recommendations and targeted digital advertising for the customer and businesses these days. This paper aims to improve upon the check-in prediction accuracy on the dataset provided by Facebook. In this paper, we use different machine learning classifiers and compare their results in terms of accuracy to find the optimal model. We also discuss how this could be enhanced further for real time solutions tailor-made towards the businesses and customers.

1 Background and Introduction

1.1 Introduction

As a person, what if you knew where you could be heading to next? As a business or a place of interest, how about you had an idea how many customers or people could be interested for the same. We travel, eat out at the restaurants, visit other places of interest, and the generation being Facebook friendly, check-in at different places.

A system which could classify and analyze social media check-in data to find patterns on how users check-in activity would be beneficial for the customers and businesses. Although, achieving this would never have a hundred percent accuracy, but a good prediction would certainly do wonders for both. Our idea is to predict where a user could be checking into from their location and timestamp. This project was a coding challenge on kaggle.com[1], and the dataset we are using is from the website itself.

We read through numerous similar projects and researches, blogs and articles to study how such a problem was approached considering a huge dataset which would need a lot of processing prior to mining to achieve the best possible results. There are multiple check-in applications like FourSquare, Swarm, etc. but in the recent few years Facebook having the biggest user base, has a strong reason of being in focus a lot more than others. The related work section talks more on the similar works on other datasets and how it was useful for ideas in our own work.

1.2 Applications and business value

Classification and prediction becomes the basis of recommendation of places of interests. Such systems have already made a major impact in online service industry. This would provide great business values to the businesses from digital advertising point and view as well as to the customers. Few of the applications are highlighted below:

- See the most common places of interests as per the geographical area.
- Allow businesses to promote themselves as per customer check-ins and find ways to improve and work on customer retention
- Realize how customer visit places as per the seasons, days, hours, etc. More check-ins in evening could be a sign of restaurant outings and allow to find favorable eating joints.
- Allow users to find the most visited hangout places around.
- Allow tourists to better plan their trips highlighting how people go about visiting a city and area as per different time of the day.

- Customized offers and personalized customer based advertising on social media and other networks.

1.3 Related work

This has been a booming area, and not only specifically Facebook, but location prediction in general has been widely researched. We did find some interesting related work which was a guide to us working on this project.

The paper “A Random Walk Around the City: New Venue Recommendation in Location-Based Social Networks”[2] talks about how large datasets from apps and devices are extracted and analyzed with algorithms and how efficient it proves for location recommendation. The authors have focused on the datasets of Foursquare and Gowalla for this one.

A study[3] was done on YELP database to predict the next restaurant check-in following Generalized Linear Model against user reviews, ratings and other attributes with an achieved accuracy of 0.35. The idea was to allow restaurants an estimate of customer turnout.

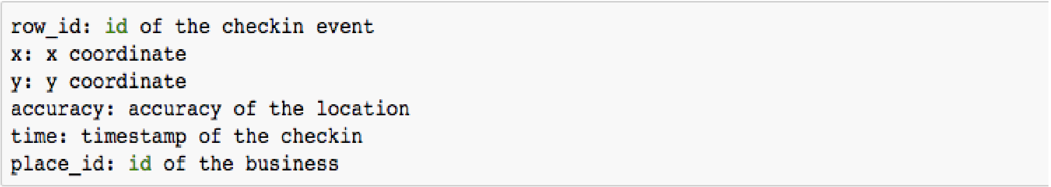
Another interesting work was performed using Gowalla and Dianping and the authors proposed a Short and Long Term Preference Model (SLoP)[4] for predicting checking with collaborative filtering techniques. They also compared it to other models which guided us to see how different methods would compare against each other to obtain a better result.

We also came across the mobile data mining framework NextLocation[5], which predicts the next place, taking privacy and many other factors into our account. It talks about how the data was preprocessed and transformed with sequences and timestamps for the best prediction and analysis. The data was all collecting from Nokia handsets.

With data sharing, also comes the security aspect. Users do not prefer to share all the personal information without any consent. The paper “Predicting Users’ Motivations behind Location Check-Ins and Utility Implications of Privacy Protection Mechanisms”[6] talked about the privacy concerns and suggested how generalization of check-ins by finding the motivation behind it solves the purpose and yet, helps us to analyze the way we want it. In our dataset, we only had the information required for the project and the names were generalized as ids beforehand.

1.4 The Dataset

The dataset we are using was provided with the project on kaggle.com. We were provided with both the testing and training data. The data files contain the following columns:



```
row_id: id of the checkin event
x: x coordinate
y: y coordinate
accuracy: accuracy of the location
time: timestamp of the checkin
place_id: id of the business
```

The place-id is what we are trying to predict. It could be a most favorable place_id or even extend to a numbered ranking list as per different patterns. One of the major challenges was to make the data more meaningful by finding patterns on features. Hence, data preprocessing and feature extraction is a critical step in this scenario. We didn’t have any majority class in the set.

2 Method

For this task of predicting check-ins of the user, Facebook has provided more than 30 million previous check-ins in a 10 by 10km grid. As described above the dataset consists of only two csvs one for training and other to test our model. The train csv consists of features: x, y, accuracy, timestamp and place_id. Our task is to predict top three probable place_ids or businesses a user checks into based on spatial and temporal information given in testing data. This is what we are trying to achieve in nutshell in this project.

To get started with the project and explore further the data in hand we set out by applying some of the most common algorithms like Linear Regression to the train data. These algorithms resulted in very bad results as we have around 100K place_ids in our dataset. Most of the supervised learning algorithm fails to work successfully with that number of classes. Thus, we needed to move on a different path or transform our data in a way these algorithms can be applied to it.

2.1 Underlying challenges in dataset

The sheer number of place_ids or classes to predict presents the biggest challenge in this project. Facebook or any other social media site that presents users with functionality like check-in do not put any restrictions on places a user can check into. This results in many places that can be checked into and hence the large number of place_ids present in our data is a common problem faced during the prediction of check-ins for a user.

The other challenging aspect of the given dataset is that definition of accuracy and timestamp column is left open to the interpretation of the candidate. This is done on purpose just to make the task more challenging. Among these two attributes timestamp is easier to interpret and to model how it can affect check-ins of a user. We know that for some businesses like restaurants lunch and dinner hours are busy hours of the day whereas if someone is checking into a mall it can be at any time of a day. Same way some places witness much more crowd during the weekend as compared to weekdays. The second attribute accuracy can be the accuracy with which x and y coordinate values of the check in is recorded. Though, exact distribution function for accuracy and its correct usage in the model is difficult to access.

2.2 Feature extraction

In the data set given to us we have five columns. The first four columns have been used to train the model to predict the place_id. We extracted some additional features from the given data that are discussed below:

- The first feature is 'x': This is x coordinate of the check-in, this feature is important in the data set in predicting the place_id, combined with the y coordinate it is used to cluster the check-in position. No data cleaning was required for this feature.
- The second feature is 'y': Similar as first feature, it is the y coordinate of the check-in.
- The third feature is 'accuracy'. This represents the accuracy of the x and y coordinates. The description for this was vague, thus we decided to use it directly in the random forest model and skipped it in the KNN model. No data cleaning was required for this feature.
- The fourth feature is 'hour': This is extracted from the timestamp and denotes the hour of the 0-24 at time of check-in.
- The fifth feature is 'weekday': This denotes the day of the week 1-7 for check-in.
- The sixth feature is 'month' of the year in range 1-12.
- The seventh feature we used is 'year'.
- The eighth feature is 'day': This is the day of the year (1-365).
- The ninth feature is 'place_id': This is what we are trying to predict in our model.

2.3 Exploratory Analysis

Due to very large number of check-ins (> 30 million) provided to us in the training data, it very difficult to visualize data for whole grid and look for patterns in it. Hence, we can take data in a smaller grid and try finding patterns or clusters in it. We read numerous approaches taken to solve this problem and the first recommended step was to partition the grid or map in smaller unit grids. This partitioning makes sense in our case because check-in for a given place is generally done nearby that location and does not vary by large distances. Thus, given the position of a person in terms of x and y coordinates, we can build an imaginary smaller grid around that x, y location and apply the classification or clustering techniques to only data points in this grid.

To reinforce our understanding of the problem and to visualize the data we start by plotting all the check-ins within a smaller grid of 500 X 500 meters taken at random from the given larger grid. We have plotted only place_ids that have more than 100 check-ins to visualize the clusters.

Some clusters are evident in the plot shown below (Figure 1) but most of these clusters overlap with each other and difficult to separate. To make these clusters separable and more evident we tried using one more feature “hour of the day” as the third dimension for our plot (Figure 2).

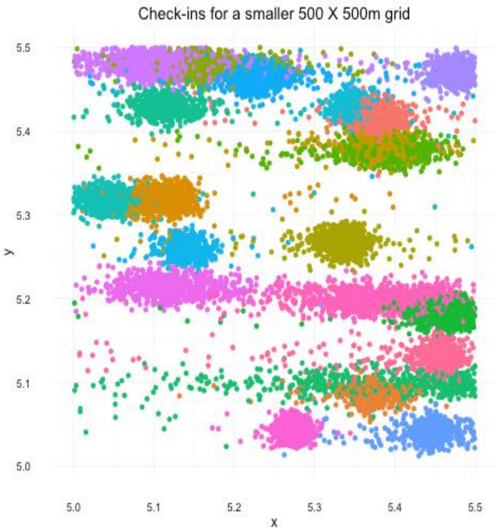


Figure 1: Place_id clustered on location

Addition of third dimension helps and we can see that our assumption that hour of day affects the check-ins for a place is valid. We tried plotting the same data using “weekday” feature that resulted in similar plot. These plots confirm our understanding that check-ins from user depends on the different time components like hour of the day and weekday.

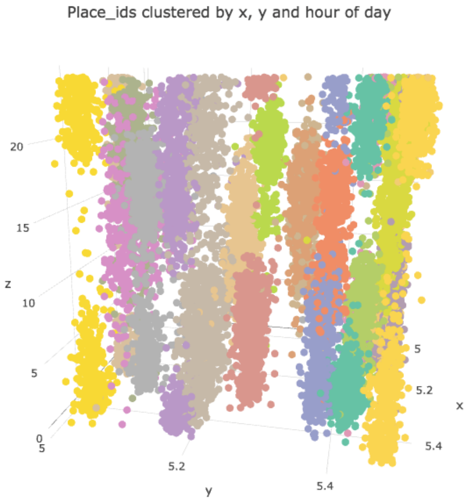


Figure 2: Place_id clustered on location and hour of day

2.4 Algorithms and Data classification techniques

For classification task, we deal with one row of the testing data at a time. As explained in the last section it is safe to assume that a check-in for a place will be done within some specified distance from the location of that place. Thus, the first step we do while processing a row of test data is draw an imaginary grid of 250 X 250 meters around that x and y coordinates. Once we have this grid in place we fetch all the data related to the points inside this grid from our training data. Now for doing the classification we will consider only these rows because the check-in location that a user can select lies in this grid. For classification, we can use multiple classifiers and techniques like clustering with KNN. As our initial models, we chose KNN and Random Forest for the same.

2.4.1 K Nearest Neighbors

Once we have a smaller grid of 250 X 250 meters in place first thing that comes to mind is KNN for the classification task. It is very easy to implement and give good results. The only tricky part about applying KNN is finding out the optimal weights for the variables used. We have used hit and trial method to optimize our model. We plan to use data exploratory techniques in future to narrow down on optimal weights for KNN in final version of the report.

2.4.2 Random Forest

Random Forest was our second choice for the classifier as it is efficient and generally results in more accurate results. The performance factor of random forest is important for us as we are doing classification task on the fly. We chose random forest also because it gives us estimate of the importance of different variables in classification task this helped us in fine-tuning our model for other classifiers as well. We tried different flavors of Random Forest available in Python and achieved best results using sklearn random forest classifier.

2.4.3 Boosted Trees

To improve the accuracy further, we tried boosted trees i.e. tree ensemble model for classification and regression trees (CART). We used XGBoost library[8], short for “Extreme Gradient Boosting”, where the term “Gradient Boosting” is proposed in the paper Greedy Function Approximation: A Gradient Boosting Machine, by Friedman. We tried boosted trees as our third classifier as it was mentioned in the blogs and discussions on this competition to provide good results.

3 Plan and Experiments

The most desirable features for any check-in prediction engine are its robustness and good performance, as the predictions have to be made in real time and should account for the data that is getting accumulated with every user click. The aim of this project was to come up with a prediction tool that can efficiently ingest large amount of spatial data and produce real-time prediction using offline processing of the data.

We extracted a few new features from the data given to us that captured the essence of the data better. Post our feature extraction, we ran different models on the preprocessed data to check the accuracy we are able to achieve using all the extracted features.

Our initial naive approach in the project was of creating an imaginary gr id(Figure 3) of 500 X 500 meters around every data point for which prediction needs to be made. Once we had this grid in place, we would fetch all the data related to the points inside this grid from our training data and train our model on this data. This resulted in a good accuracy but our model was painfully slow. This was due to the large amount of time required to go through the entire training dataset and retrieve all the points that would correspond to this grid.

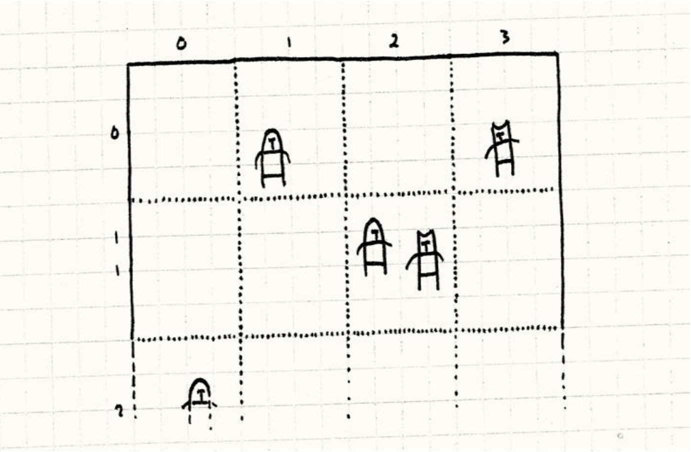


Figure 3: GRID Analogy

To make our models more efficient, we tweaked our approach a little to segregate data for every block on grid beforehand. Before we start predicting the check-ins we divided the entire training dataset into 100 blocks based on their x and y coordinates. For every row in the test data set we would select the block that the data set would lie in based on its x and y coordinates and use the check-in data from that block as well as all the 8 neighboring blocks to train our models to make a prediction. This removed the process of iterating through the entire training dataset for every test data and improved the performance by more than thousand-fold for obtaining the results.

Check-in prediction is a classic classification problem, the output of which can be one of the nearly 100 thousand different place_ids (output class) provided to us in the form of 30 million simulated check-ins. The three models we applied to our data set predict the output class for each check-in in the test data. Thus, to evaluate our models the choice was simple accuracy. The accuracy of each model is defined by the number of prediction the model gets correct with respect to the total number of the check-ins in the test data.

4 Results

From the similar studies on location and check-in prediction, we saw and concluded accuracy to be our best method when predicting the next check-in. The results are summarized in Table 1.

Method	Accuracy
K Nearest Neighbor	0.42
Random Forest	0.48
Boosted Trees	0.54

Table 1: Accuracy measure

After processing the data, we first ran the K Nearest Neighbor model which resulted in the accuracy of 0.42 which wasn't bad but to see if we could improve further, we went on to Random Forest with sklearn library in Python. The result was event better, 0.48 to be precise. We further looked to see if another model can enhance it and Boosted trees worked for us. We ran it with the XGBoost library. Out of all the three, this gave us the best accuracy of 0.54. We can see the comparison in Figure 4.

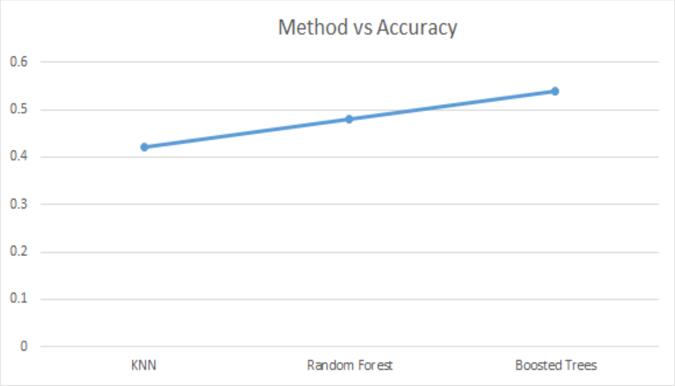


Figure 4: Accuracy comparison on methods

5 Conclusion and Future Work

We obtained good results using simplistic model and grid framework. Although, we tried several classifiers for the prediction task, we included results from our three better ones. As per Kaggle's site, winner of this competition could achieve accuracy of 0.58 and our results were very close to the same with our algorithmic implementation.

We started the project with very straightforward approach of creating grid around every data point for which prediction needs to be made. This resulted in a good accuracy but our model was painfully slow. Further to make it more efficient, we tweaked our approach a little to segregate data for every block on grid beforehand. This improved the performance by more than thousand-fold for obtaining the results.

We believe that further optimizations can be done in the grid framework on similar grounds to improve the performance as well as accuracy of the classifiers. We can make use of ensemble methods in future to improve our accuracy even further. Also, we can take vote from different classifiers to narrow down to the correct place_id for predicted value. An enhancement further could be a ranked list of next place predictions, end even depending upon a particular time the of day.

Apart from improving accuracy, we can also think of additional use case for the check-in prediction engine. Our engine can be used along with other recommendation systems to include the perspective future destination of a user and in turn help recommending meaning services to the customers. Imagine a near perfect next check-in recommender tailor-made to users and at the same time, targeted audience estimation for businesses. This would be a boon to the industry.

Throughout the course of this project, we learnt many valuable lessons that are required to implement any Machine Learning model. Our first take from an application based mining project was that the dataset is always going to be huge, and will probably not make much sense at first. We started the project by exploring the given data by visualizing it using different plots. This helped us understand the dataset in a much better way, and gave us a valuable lesson that we should not get overwhelmed by seeing a large dataset. We used grid approach to divide and hence reduce data fed to classifiers that improved the performance of the system exponentially. In addition to this, we learnt about various machine learning techniques like data preprocessing, data exploration, clustering, and boosted trees that are essential in any data mining task.

6 References

[1] Kaggle project challenge: https://www.kaggle.com/c/facebook-v-predicting-check-ins

[2] A Random Walk Around the City: New Venue Recommendation in Location-Based Social

Networks: Anastasios Noulas, Salvatore Scellato, Neal Lathia, Cecilia Mascolo

[3] YELP Restaurant checkin prediction: https://rpubs.com/WaiKeen/FinalProject

[4] Collaborative Filtering Meets Next Check-in Location Prediction: Defu Lian, Vincent W. Zheng, Xing Xie

[5] Where will you go? Mobile Data Mining for Next Place Prediction: Joao Bartolo Gomes, Clifton Phua, Shonali Krishnaswamy

[6] Predicting Users' Motivations behind Location Check-Ins and Utility Implications of Privacy Protection Mechanisms: Igor Bilogrevic, Kevin Huguenin, Stefan Mihaila, Reza Shokri, Jean Pierre Hubaux

[7] A data mining approach for location prediction in mobile environments: Gokhan Yavas, Dimitrios Katsaros, Ozgur Ulusoy, Yannis Manolopoulos

[8] XGBoost Library: http://xgboost.readthedocs.io/en/latest/model.html

A. Appendix

The dataset we used was from the project's source on kaggle.com. Our code for the preprocessing and data mining techniques is available in our Github repository. Below are the links to the same.

Data Set	https://www.kaggle.com/c/facebook-v-predicting-check-ins/data
Code	https://github.com/shivamgulati1991/Facebook--Footsteps-Prediction