Facebook: Check-Ins Prediction

Shriram Kumar (sckumar@eng.ucsd.edu) (A53221613)
Devyani Kulkarni (dakulkar@eng.ucsd.edu) (A53211917)
Balachander Padmanabha (bpadmana@eng.ucsd.edu) (A53202177)
Nikhil Yogendra Murali (nyogendr@eng.ucsd.edu) (A53220135)

***Abstract*—This report outlines the results of an empirical study of the Facebook check-in data and describes a model designed based on the study to predict check-in locations. The features available are the location co-ordinates, time and accuracy for each check in. The main challenges in the data-set are:**

- **The large number of possible check-in locations (∼38,000)**
- **Evaluation is performed using the mean precision @3 metric**

We use a generative approach to model the distribution of the features. The model is then used to output the top three ranked predictions for every test point. It is shown empirically that this method outperforms discriminative classification methods in terms mean precision @3. It is also observed that there are some independence patterns among the features and this is used to factor the generative model and learn individual models for independent sets of variables. There are several statistically significant observations about the data which are used to engineer features accordingly. The resulting model is competitive with top solutions on the Kaggle leader board with a much smaller amount of feature engineering.

Keywords*—*Check-in, Facebook, Generative Model, KNN, Discriminative Model, Logistic Regression

I. Introduction

User check-ins have gained popularity on social media platforms off late and an important associated task is that of recommending check-in locations to users. This motivated us to look into the 'Facebook V: Predicting Check Ins' dataset on Kaggle. This competition was part of a recruiting event held by Facebook. While in most recommender systems, we build a user profile, here the task is to recommmend check-ins solely based on the location and time of an user. The dataset consists of roughly 30 million anonymized check-ins to roughly 38,000 unique places. For computational feasibility with our resources, we are considering ∼270,000 check-ins in ∼4,000 unique places.

In the first section, we describe our dataset in detail and our exploratory analysis leading to a few interesting insights. We analyze the scope and relevance of each of these insights for our predictive task.

In the second section, we identify our predictive task and describe in detail the features we used in our model. The feature section primarily focuses on engineering the time and accuracy related parameters along with the location features for our model.

In the third section, we start off by developing a baseline model using KNN of all the features. Further, we explore the conditional independence relationships of several features given place_id and design an appropriate generative model for the data. We also discuss how to learn the parameters for this generative model and classify new test points using the model. We also propose a discriminative approach to learn the parameters of our model as an alternative.

We discuss and compare the performance of all our models using mean precision@3 and accuracy as the evaluation metrics.

Finally, we discuss literature relevant to our work and summarize our learnings from this exercise.

II. Exploratory Analysis

A. Data

In this section, we examine the dataset and its features as a whole. The given dataset has data from roughly 30 million anonymized check-ins in a 10 km X 10 km grid in the simulated world created by Facebook for its Kaggle Recruiting Event.

Following are the fields of our data:

- row_id : This specifies the record number of the check-in.
- X : The x-coordinate bounded between [0,10]
- Y: The y-coordinate bounded between [0,10]
- Accuracy: The accuracy with which the check-in was made. This presumably represents the accuracy of the (x,y) location dependant on environmental factors, device characteristics etc.
- Time: This feature is mentioned as being vague by Facebook. After some analysis, we presume that it is the number of minutes passed since some reference time.
- place_id: This is a unique identifier representing the place at which the check-in was made.

The given data has roughly 38,000 place_ids into which every check-in is classified. For the purpose of our task, to make the dataset computationally viable for our resources, we only consider the 1 km X 1 km grid from the top-right corner of the given world. This reduces our dataset to ∼270,000 check-ins with ∼4,000 place_ids.

We further divide our dataset into train, validation and test sets in the ratio of 2:1:1 through random sampling.

In figure 1, we plot the frequencies of places in the dataset. We can see that the distribution has a long tail corresponding to places with very few observations in the training set. This makes the problem challenging as estimating parameters for these points accurately is very difficult. We observe that some places are more popular than others in terms of check-ins.

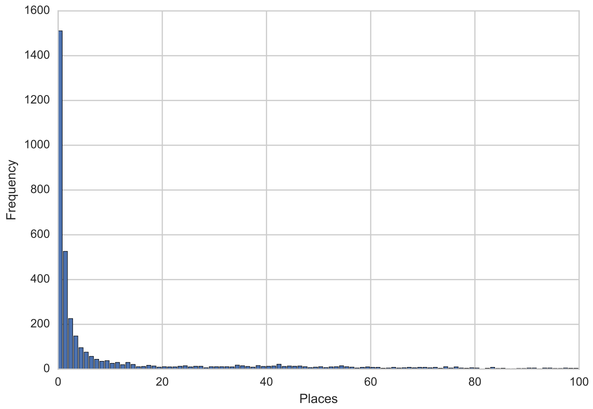

Fig. 1. Frequencies of places

B. *x, y*

This section discusses the distribution of the place_ids with the grid location x and y.

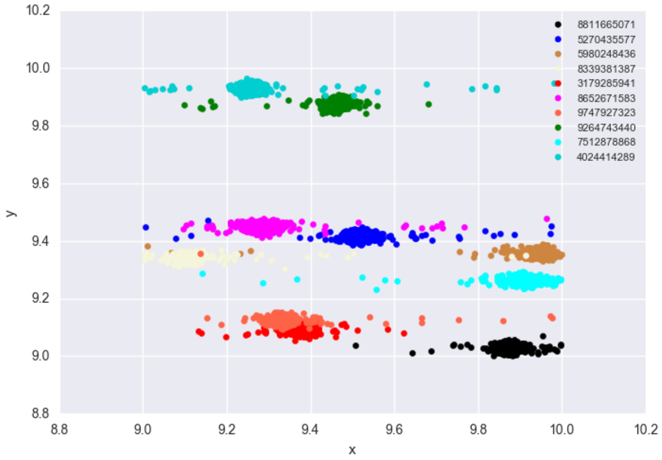

Fig. 2. Location distribution of top 10 place_ids

In figure 2, the distribution for ten most popular places shows that the variation along x is much more than that along y. The larger variance along x could be due to the fact that the check-ins are made on the horizontally placed lanes of the grid. The narrowness of these lanes could explain the low variance along y.

C. *Time*

The definition of time and its scale in the dataset is intentionally left vague as per the specifications. We use kernel density estimation function to get a better idea of how the popularity of a place varies with time. We picked top 5 popular places based on number of check-ins to observe differences in check-in patterns. We see that the popularity of the place can vary significantly with time and we further explore this in the feature design.

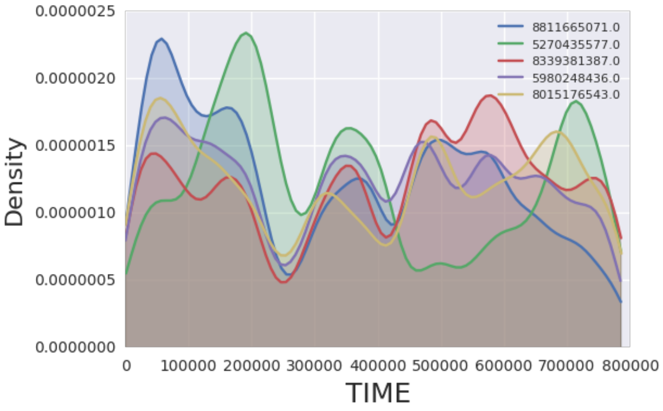

Fig. 3. KDE plot of time for 5 random place_ids

The peaks in the figure 3 give valuable information that certain places have higher/lower check-in rate at specific time periods. From the time values in the dataset it can be inferred that the time spans across 24 hours of the day, 7 days of a week, 12 months of the year and 2 specific years. The reason certain places might be popular at certain time values could be due to the nature of the place example, parks have higher number of visitors during the day while pubs have during the night. We also explore such periodic characteristics in the feature design.

D. *Accuracy*

The dataset specifies that the values are from an artificial world. However, we are not sure what the accuracy values signify. We can use accuracy as a measure of the variance in x,y for a point given a place_id. Another way accuracy can be used is, given a place, we can have a profile of the type of devices used to check-in.

To check the relationship between accuracy and place_id, we plotted histograms representing the bins of the accuracy distribution for the most frequent place_ids. The plots show that the distribution changes with respect to places and hence can be used as a feature.

To analyze the full data, we plotted histogram of all accuracy values.

From the plots 4 and 5, it can be observed that there are generally 3 peaks in accuracy distribution. This is further discussed in the next section

III. Predictive Task

The purpose of our model is to predict the place_id of a given check-in containing (x,y) coordinate,time and accuracy. Since every place_id is unique, the task at hand is to classify

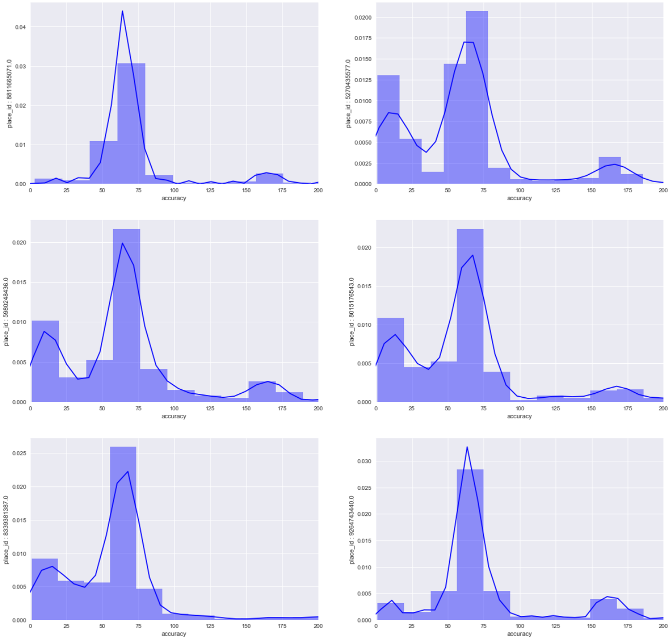

Fig. 4. Accuracy vs place_id for top 6 places

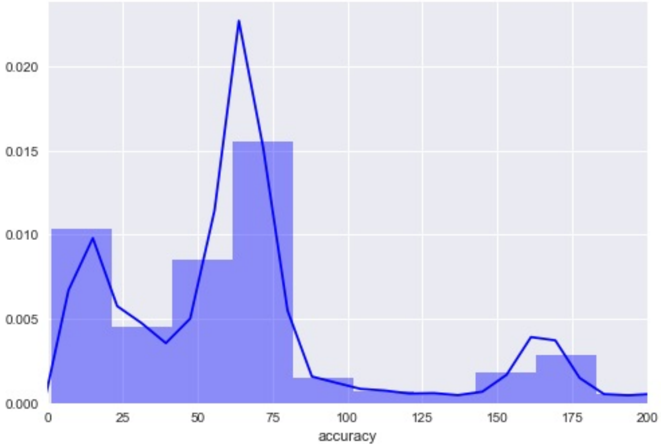

Fig. 5. Accuracy vs place_id for full data

our prediction into one of the ∼4,000 place_ids possible. The large number of place_ids in the dataset make this multi-class classification task computationally intensive. Places which have lower data points are harder to predict due to dominance by places with larger data points in the dataset.

We use KNN as a baseline model to predict the place_ids. To outperform this baseline, we implemented Discriminative, Generative and Ensemble models. Each of these models inherently use strengths of different features as described in the feature design section. The metric that is used to measure performance is mean precision @3. Optimizing mean precision @3 is analogous to optimizing ranking performance.

As per the results, Ensemble of several generative models outperforms the other models as the different hyperparameters used in it allow us to capture location parameters at different scales more accurately. Further details about different models and their performance are provided in the coming sections.

IV. Feature Design

In this section we describe various features that we believe are important to our multi-class classification problem.

A. *x, y*

The exploratory analysis of these features demonstrates that they are well clustered around the place_ids they represent. Since the variation along y is less than that along x, y should be more predictive than x.

B. *Time*

We reason that popularity of places might be affected by several periodic factors like time of day, day of week etc. We extract these features from the data for two randomly selected place_ids and plot their densities.

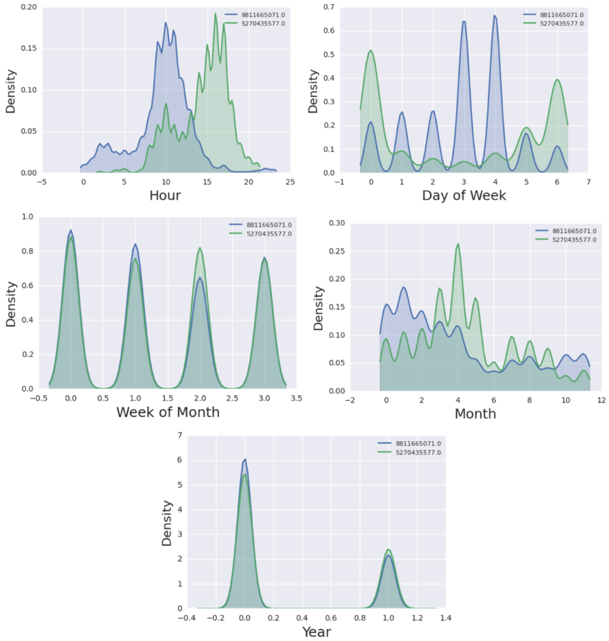

Fig. 6. Time features density plot

We observe in figure 6 that there are strong patterns in the frequency of hour, day of week and month with place_id. Week of month and year seems to be poor predictors in this plot. While training our classifier, we observe that the week of month is indeed a poor predictor while year gives us performance gains.

C. Accuracy

The exploratory analysis of Accuracy demonstrates that, in general, there are 3 peaks in the distribution and hence the classification can be done using 3 bins. On careful observation, the bin ranges were chosen as $accuracy < 40$, $40 \leq accuracy < 80$, $80 \leq accuracy$ and one-hot encoding has been done with these bins as three features.

V. Model

The probability of a place given its features can be expressed in the form

$$p(place \mid x, y, time, accuracy) \propto \\ p(x, y, time, accuracy \mid place) p(place) \quad (1)$$

We reason that the time, location and accuracy features are independent given the place. We validate this assumption by observing data grouped by place_ids. The paired plot above shows the data for two places. We can see that the accuracy distribution is fairly independent of x,y and time for both places. Similarly we can also see that the location is pretty much independent of time for both places. This leads us to formulate the model as

$$\text{p(place} \mid x, y, time, accuracy) \propto p(x, y \mid place) \\ p(time \mid place) p(accuracy \mid place) p(place). \quad (2)$$

The remainder of this section describes how to model each of these distributions. We take a generative approach to learning the parameters of these distributions rather than a discriminative approach. Traditionally, generative approaches are not used as the feature distributions are difficult to approximate accurately. However, here we have a large amount of data and small number of features allowing us to model conditional distribution accurately. The reason for this is that although discriminative models learn parameters with better generalization capability, they are not guaranteed to be consistent in the way they rank instances other than the most relevant one. Since the metric used in evaluation here is mean precision @3, we hypothesize that the generative models might deliver better results. A comparison in the performance between the discriminative and generative methods of estimating parameters is shown later in the section.

A. Modelling x,y

We tried several models for the x,y coordinates given a place. The first model we tried was based on the k nearest neighbors model. The exact procedure is as follows: (1) We build a kd-tree using the training data. (2) Given a test point, we retrieve the k nearest neighbors from the training set and count the number of instances n_p corresponding to each place p in that set. (3) The probability of each place given x,y is estimated as $\frac{n_p+s}{k+ks}$ where s is the Laplace smoothing constant. (4) We calculate the probability of x,y given the place using Bayes rule. In this model, we introduce an additional hyperparameter α that scales the importance of distances in y relative to x. This parameter is useful as the variance in y is much smaller than in x and hence it is a more useful to discriminate among places.

We also tried fitting Gaussian mixtures with up-to 5 components to the data from place with larger than 20 check-ins. The number of components was chosen using the BIC value on a validation set. However, we did not see a significant increase in predictive power and since the distribution of x,y for places does not seem to satisfy Gaussian assumptions, the non parametric KNN model was chosen.

B. Modelling Time and Accuracy

We have three cyclic features, day of week, time of day and month of year. The other time features are year and the actual absolute time in minutes. The way these features are designed, we expect them to be independent of each other. Hence we model $p(time \mid place)$ as a product of these individual models.

Day of week is a Multinomial variable with 7 classes. We estimate it's parameters using the maximum a posteriori approach. An appropriate prior is chosen to reflect the base belief that every class behaves similar to the population average.

A similar approach is taken to modelling other binned variables i.e. cyclic time variables as well as the accuracy variable. The windowed density estimate reflects the popularity of that place at the given time. This is calculated by counting the number of occurrences of the place_id in a window around the current time divided by the total number of occurrences of that place_id. We use Laplace smoothing to make these estimates more reliable. The window sizes were chosen from among one week, one day and a few hours using a validation set.

C. Discriminative model

Here the probability estimates for the time and accuracy features are the parameters. The KNN model and the windowed time estimate have no parameters to be optimized. The likelihood function of our model is similar to the logistic regression model since the remaining variables are multinomial distributed. Using a one hot encoding for all the time and accuracy features, we can train a modified logistic regression model to perform multiclass prediction among the places. This forms our discriminative model for comparison.

D. Ensembling

To obtain the best results, we ensemble generative models using several settings of the parameters k i.e the number of neighbors for the nearest neighbor classifier and the relative importance given to y in the distance metric α. We use α values of $1, 2.5, 5, 10$ and k values set to $10, 20, 50, 100$. The predictions of various classifiers are averaged out. The top three ranked places in terms of the average score are given as predictions.

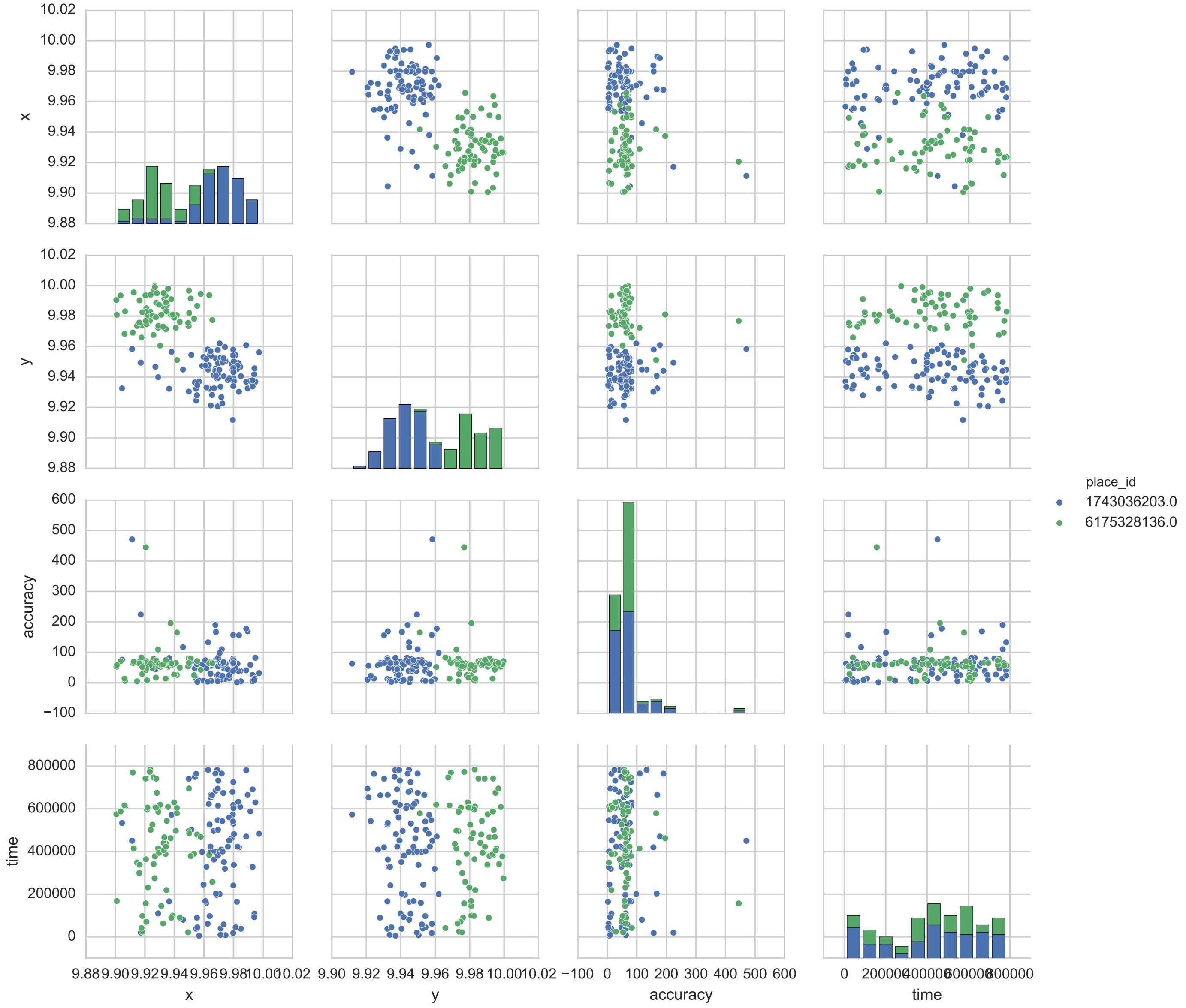

Fig. 7. Paired Independence plots for the features

VI. Results

We show the accuracy and mean precision @3 for the following models: The baseline KNN model, the discriminative model, the model with parameters estimated generatively and an ensemble of several generative models trained with different hyper parameters.

Model	**Accuracy**	**Mean Precision@3**
K-Nearest Neighbors	0.26	0.39
Discriminative Model	0.51	0.56
Generative Model	0.505	0.602
Ensemble	0.501	0.61

Our ensemble model achieves mean precision close to the values at the top of the Kaggle leaderboard. Although we have used only a small subset of the data, we expect that the model will scale well.

We observed that the y coordinate was the most helpful feature, followed by x, time of the day, day of the week accuracy binning. The other features, even though were important, provided small performance improvements.

VII. Relevant Literature

As stated before, this dataset is taken from the Kaggle competition held by Facebook. The dataset consists of roughly 3o million anonymized check-ins to roughly 38,000 unique places. For computational feasability, we consider ∼270,000 check-ins in ∼4,000 unique places. Even though this is a generated dataset, similar real-world social network data can be utilized, in similar fashion, to predict personalized check-in recommendations. Similar datasets exist for Google Plus, Twitter and other social networking platforms.

We describe some of the top solutions on the Kaggle leaderboard as well as some research papers we referred to, to understand the problem better.

The top solution on the leaderboard was a very complex model consisting of several layers. The first level of learners are used to narrow down the candidate solutions and the second level learners further identify the top three. The dataset is divided into grids and the models are learned for each of the resulting cells. The solution makes use of gradient boosted trees and KNN at both layers across cells. The final solution is an ensemble across a wide range of 70 settings of Hyperparameters. Out of all the features from first level learners, 21 are manually selected and applied to second level learners. While the model is more complex compared to ours, we make use of binning of accuracies as a feature. We also use the idea of ensembling across hyperparameters although at a much smaller scale.

We referred to several papers that helped us understand the problem better. In [4], Purushottam Kar et al. tackle the problem of optimizing precision @k using a ranking formulation. They propose a convex surrograte to the precision @k metric. They demonstrate several desirable properties such as consistency and margin conditions and propose a perceptron algorithm to optimize the proposed cost function. We also referred to several papers about one class recommendations and extreme multilabel classification which helped us understand the problem better [5] [6].

VIII. Conclusion

The baseline KNN model performs way worse than the other models due to the fact that our feature engineering of time and accuracy is not suitable for this model. We see that although the discriminative model and the generative model both have similar accuracies, the discriminative model performs almost 4% worse in terms of mean precision @3. This gives empirical evidence to support our claim that the generative model is a better ranker than the discriminative model. The overall best predictor is the Ensemble of several generative models. The different settings of k and α let us capture spatial patterns at several scales leading to better performance.

The proposed ensemble model achieves performance comparable to the top solutions on Kaggle, with limited feature engineering. The main take-away from this project was that careful selection of priors makes a big difference in predicting check-ins for this dataset.

References

[1] Facebook Inc, https://www.kaggle.com/c/facebook-v-predicting-check-ins/data

[2] Tom Van de Wiele, https://www.kaggle.com/c/facebook-v-predicting-check-ins/discussion/22081

[3] Michael Griffiths, https://www.kaggle.com/msjgriffiths/facebook-v-predicting-check-ins/exploratory-data-analysis

[4] Purushottam Kar, Harikrishna Narasimhan and Prateek Jain, *Surrogate Functions for Maximizing Precision at the Top*, Proceedings of the 32nd International Conference on Machine Learning 2015, Lille, France

[5] Kush Bhatia, Himanshu Jain, Purushottam Kar, Manik Varma, and Prateek Jain, *Sparse Local Embeddings for Extreme Multi-label Classification*, Advances in Neural Information Processing Systems 28 (NIPS 2015)

[6] Rong Pan1, Yunhong Zhou, Bin Cao, Nathan N. Liu, Rajan Lukose, Martin Scholz and Qiang Yang *One-Class Collaborative Filtering*, IEEE International Conference on Data Mining (ICDM 2008)