CS 289 ML - ALGORITHMIC MACHINE LEARNING

PROJECT REPORT

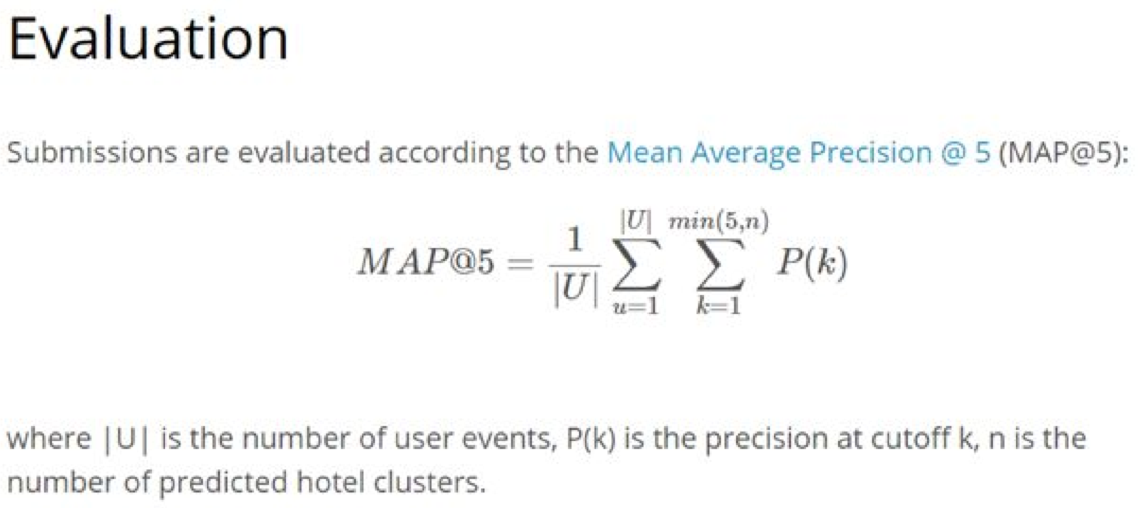
Atishay Aggarwal (UID- 704758311)
Aanchal Dalmia (UID - 404741586)

Expedia Hotel Recommendations

1. PROBLEM STATEMENT

This project is based on the Expedia Hotel Recommendation challenge on Kaggle.com (https://www.kaggle.com/c/expedia-hotel-recommendations). This challenge is about predicting which hotel group, a user would book. Expedia uses its own algorithms to form hotel-clusters, wherein similar hotels are grouped together. So, our aim in this challenge is to predict the hotel cluster for a user, based on their search parameters and other relevant attributes. Kaggle has provided a dataset of logs of customer behavior with Expedia. The application of this project would be predicting the hotel clusters that the customer is likely to book and providing recommendations to the users.

The evaluation metrics to be used are:



Evaluation

Submissions are evaluated according to the Mean Average Precision @ 5 (MAP@5):

$$MAP@5 = \frac{1}{|U|} \sum_{u=1}^{|U|} \sum_{k=1}^{min(5,n)} P(k)$$

where |U| is the number of user events, P(k) is the precision at cutoff k, n is the number of predicted hotel clusters.

Figure 1 : Evaluation metrics

Our aim is to predict 5 hotel clusters for each user and the output is an ordered list. We get a higher score if the correct cluster is present closer to the start of the list.

2. DATASET

The training dataset that we have used consists of 2 million examples and the testing dataset consists of about 20000 examples. There are 17 attributes present in the dataset(Table 1). There is also a file ‘destinations.csv’. This file consists of 149 attributes which aren’t labelled, but have some effect towards the user’s selection. It consists of features extracted from the hotel reviews text. Since these attributes do not reflect any qualitative information, we use Principal Component Analysis (PCA), to determine the 3 principal components that affect the selection, and then we append these 3 components to our original training dataset by performing a ‘join’ operation over the attribute ‘destination_id’.

Table 1

Feature Name	Description	Datatype
site_name	ID of the Expedia point of sale (i.e. Expedia.com, Expedia.co.uk, Expedia.co.jp, ...)	int
posa_continent	ID of continent associated with site_name	int
user_location_country	The ID of the country the customer is located	int
user_location_region	The ID of the region the customer is located	int
user_location_city	The ID of the city the customer is located	int
orig_destination_distance	Physical distance between a hotel and a customer at the time of search. A null means the distance could not be calculated	double
is_mobile	1 when a user connected from a mobile device, 0 otherwise	int
is_package	1 if the click/booking was generated as a part of a package (i.e. combined with a flight), 0 otherwise	int
channel	ID of a marketing channel	int
srch_adults_cnt	The number of adults specified in the hotel room	int
srch_children_cnt	The number of (extra occupancy) children specified in the hotel room	int
srch_rm_cnt	The number of hotel rooms specified in the search	int
srch_destination_id	ID of the destination where the hotel search was performed	int
srch_destination_type_id	Type of destination	int
hotel_continent	Hotel continent	int
hotel_country	Hotel country	int
hotel_market	Hotel market	int

3. DATA PREPROCESSING

Before building models on the data, preprocessing techniques are used to clean the data. The following 2 steps are followed.

3.1 PCA

The first step to the approach is to use Principal Component Analysis. This reduces the number of columns in ‘destination.csv’, while preserving the same amount of variance per row. There are 149 latent features that describe the destination. We have selected the top 3 features which preserves most of the variance and also saves a lot of computation time for the machine learning algorithm. These 3 features are appended to the training data be performing a join operation on ‘destination_id’.

3.2 Feature Generation

In order to build a powerful prediction model, it is important to select an optimal subset of features and to add some useful features. We have generated new features based on date-time, check-in and check-out dates. Non-numeric columns like date-time are removed. Any missing values in the table are replaced by -1.

3. MODELS AND APPROACH

We have tried various approaches to the problem using the algorithms studied in the course and some algorithms that we came across.

3.1 Logistic Regression

We first tried to find whether some correlation exists between different columns of the dataset. Pearson coefficient was used to determine the correlation.

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[n\Sigma x^2 - (\Sigma x)^2][n\Sigma y^2 - (\Sigma y)^2]}}$$

Figure 2: Pearson coefficient formula

Pearson coefficient gives values ranging from -1 to 1. A value of 1 indicates strong positive correlation and a value of -1 indicates strong negative correlation. A value of 0 indicates no correlation. It computes the correlation between the attributes pairwise. Since the attribute ‘hotel_cluster’ was to be predicted, we tried to determine the correlation between this and other attributes. We observed that most of the attributes gave a Pearson coefficient around zero. This was an indicator that linear and logistic regression models would not be suitable for our dataset.

We started with implementing a Logistic Regression model. In logistic regression, the dependent variable, i.e 'hotel_cluster' in this case, is predicted using the various independent variables. However, logistic regression works well if the dependent variable is binary. This was evident from the accuracies in prediction obtained using this model which were quite low(0.15). We even used 'l2 norm' penalizing factor in our regression model. This basically resolves the issues of multicollinearity between different attributes. Also, it shrinks the coefficients of those attributes that have very little contribution towards the outcome.

3.2 Random Forest

We have used a 3-fold cross validation approach to get reliable error estimates. Cross validation divides the training data into 3 parts. It predicts the hotel_cluster for each part by using a model trained by the other two parts. The predictions are made using Random Forest Algorithm. It generates multiple decision trees, one for each class/cluster. It outputs a class that is the mode of the classes of the individual trees. We could achieve an accuracy of 0.16 from this model.

3.3 K-means clustering

K-means is an unsupervised learning algorithm that forms k clusters of data such that the intra-cluster similarity is maximum and inter-cluster similarity is minimum. To begin with k-means, we first selected the relevant features that showed high variance with respect to the target variable i.e. hotel_cluster. Hence, we selected the features that describe the user location, his search query and the hotel location. The next step was to select an appropriate value of k such that it will maximise the accuracy while avoiding overfitting the data. Euclidean distance is used to find the dissimilarity between the data.

Trying different values of k gave us the accuracies as shown in Fig.3. It is seen that maximum precision is achieved for k = 250. After that the precision gradually decreases. Hence, it is considered as the global maxima for the dataset. There is a local optima around 150 which also gives a high precision. The graph shows that with the increase in the number of clusters, the precision initially shoots up, decreases a little and then reaches the maxima.

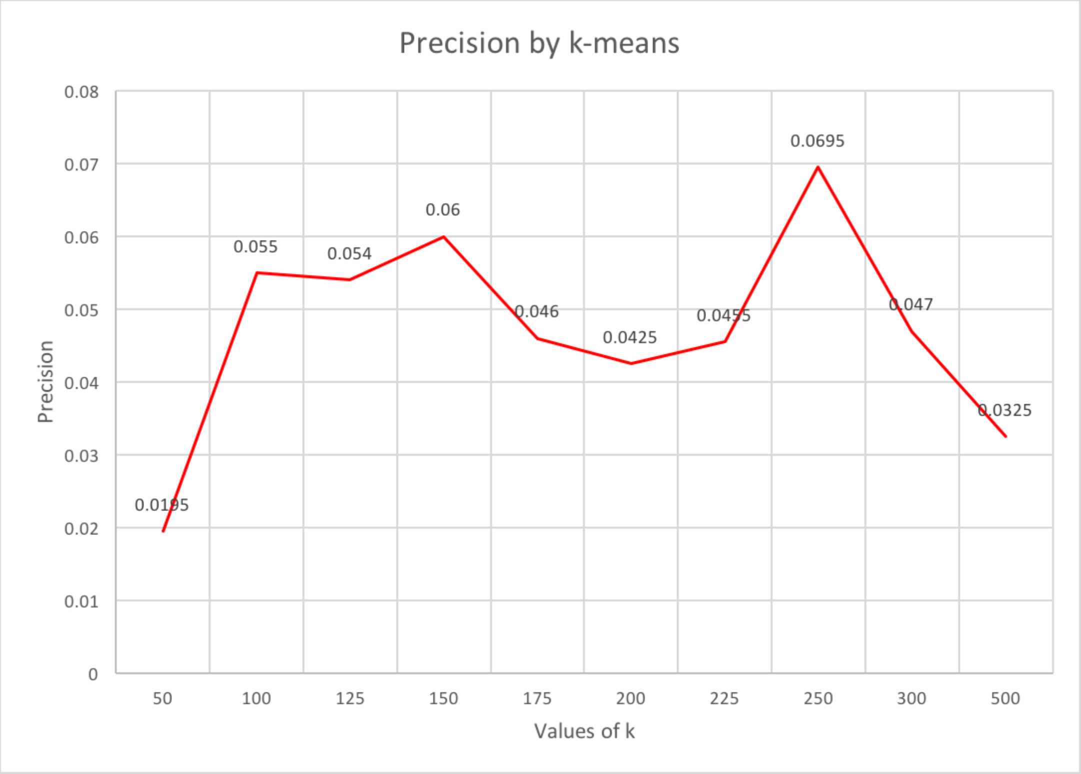


Figure 3: Comparison of precision with different values of k

3.4 K-nearest neighbours

This is a classification algorithm that finds the k-nearest neighbors for the data. We used the clusters from the k-means model. We assigned each tuple from the test data to one of the cluster. We then found the k-nearest tuples for each test data. The hotel_cluster was computed by majority voting i.e. it was assigned the class that was most common among its k- nearest neighbors. Since for the competition we need to predict 5 hotel_clusters, we selected the 5 classes that were most frequent among the neighbors. Selecting an optimal value of k was again crucial here. We could achieve an accuracy of 0.2 with this model.

Testing different values of k gave us the precision as shown in Fig.4. This figure shows the trend in precision for different values of k. It can be seen that both the models react similarly when the values of k are altered. However, knn was found to be more precise than k means in predicting the hotel clusters. The maximum precision is seen around k=250. Thus, k-nearest neighbors proved to be a better model than k means clustering.

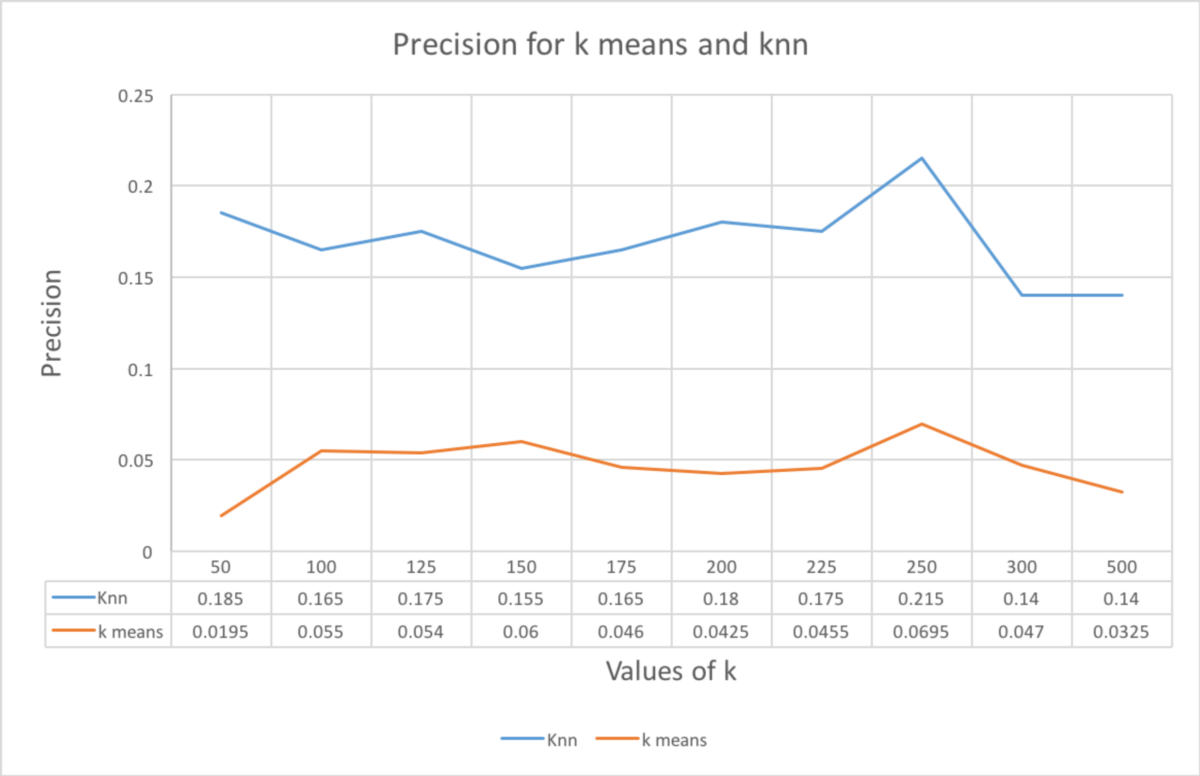


Figure 4: Comparison of accuracies for k means clustering and k-nearest neighbors

3.5 User-based collaborative filtering

Collaborative filtering is based on the intuition that similar users are more likely to select similar hotel clusters while booking. The similarity of the users can be based on their previous bookings of the hotels. We used matrix factorisation to implement a collaborative filter. The matrix is a nxm matrix M, which includes the bookings of n users at m hotel clusters. This matrix has to be decomposed into two matrices P and Q, such that M = PxQ^T, where P is a nxk matrix and Q is a mxk matrix. Here, k is a number of latent features that describes how a user makes a hotel booking. Thus each row of P describes the strength of associations between the user and features and each row of Q describes the strength between the hotel cluster and features. In order to compute the values of P and Q, we initialised them with random values and then employed Gradient Descent such that it minimised the difference between PxQ and M. To avoid over-fitting of the data, regularisation has been used. The major drawback of this approach is that it is extremely time-consuming. This model required the maximum amount of time even for a small subset of training data.

3.6 Assigning scores to cluster based on destination

We went through the kaggle scripts and observed that many approaches had aggregated the hotel_cluster on the destination location. This means that we find the most popular hotel clusters for each destination. We will then predict the top 5 most famous among the hotel clusters for the destination that a user searches. It is intuitive that the user will most likely go to popular hotel_cluster. We first assigned scores to each hotel cluster for each destination_id. These scores were based on the booking data and click data. Hence, for each destination_id, we had a list of hotel clusters with the assigned scores. These scores reflected the popularity of the hotel clusters. For each tuple in the test data, we found the top 5 hotel clusters for the destination_id. This approach gave us a high precision of 0.32.

4. RESULTS

Different levels of accuracies were achieved by running these models on the data. Fig 5 shows the comparison of the precision vs the models. This figure clearly shows that maximum precision is achieved by the model that scored the clusters depending on the destination. This shows that the users generally booked popular hotels for a particular destination. A remarkable thing to observe is the increase in precision when k-nearest neighbors is used in comparison to k-means clustering. The technique of majority voting boosted the precision of this model.

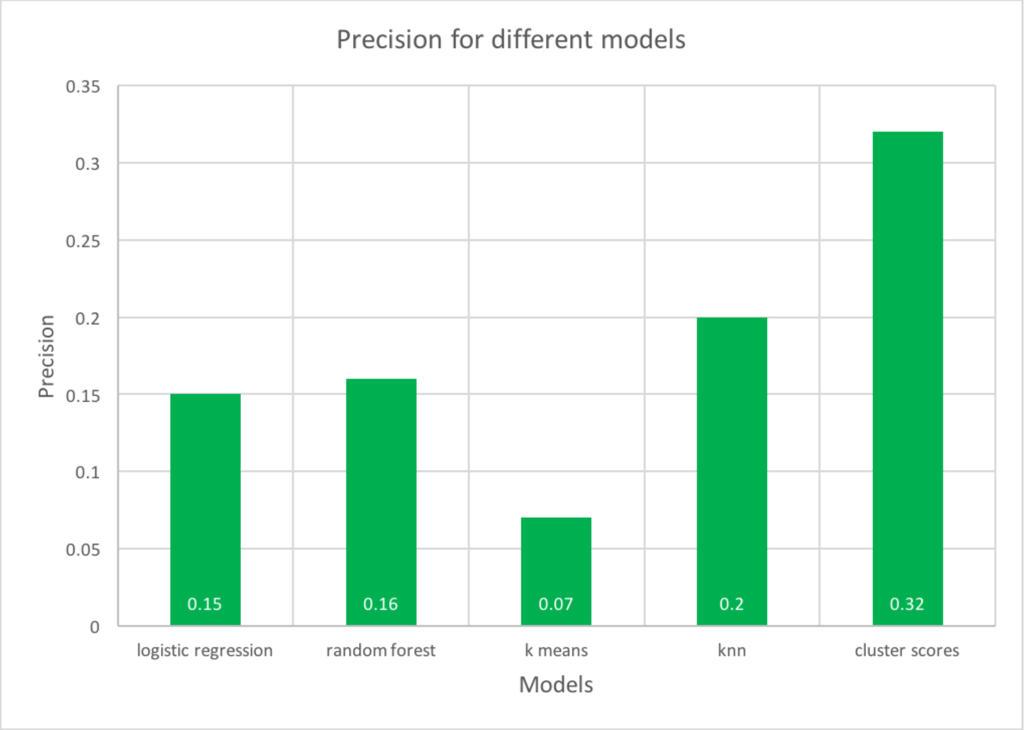


Figure 5: Precision vs model comparison

5. FUTURE WORK

In the future, we plan to implement other machine learning algorithms like neural networks, multiclass SVM and compare their results against ours. We plan to find a better way to replace the missing entries in the training and test data. This could be done by replacing the missing entries with the mean of the column data or building another prediction model to predict the values that are missing. We also plan to combine the different classifiers and take the weighted average of them to get the final prediction. Since, the cluster scores model worked best, maximum weight could be assigned to it.

6. CONCLUSION

This project showed us the practical applications of Machine Learning algorithms on real world data. We implemented different models like Logistic Regression, Random Forests, k-means clustering, k-nearest neighbors and collaborative filtering and evaluated their performance. By tweaking the parameters of these models, we could see a huge leap in precision values. We also learnt about feature engineering and realised that an appropriate selection of features play an important role in deciding the accuracy of the model.

7. REFERENCES

1. http://www.quuxlabs.com/blog/2010/09/matrix-factorization-a-simple-tutorial-and-implementation-in-python/
2. https://www.dataquest.io/blog/kaggle-tutorial/
3. http://acsweb.ucsd.edu/~dklim/mf_presentation.pdf
4. http://cs229.stanford.edu/proj2016spr/report/065.pdf
5. https://github.com/2giridhar/SMLProject
6. https://www.kaggle.com/dvasyukova/expedia-hotel-recommendations/predict-hotel-type-with-pandas