ITCS 6156 – Machine Learning
Final Project Report
Group 12 – Pritam Borate, Puneeth Devabhaktuni, Aditi Helekar, Lily Naoudom

Abstract
Uber Technologies, Inc. is a transportation network company and a pioneer in the shared economy business. The Uber software application, better known as the “Uber App”, was developed using GPS technologies for Uber drivers and their clients to use during a transaction. Pertinent details regarding a transaction include the date, time, and geographical location of the Uber request. Data sets containing this information for New York City were found online at Kaggle.com, and were used to create visual depictions such as charts, graphs, and maps to highlight the busiest locations for Uber drivers.

The problem addressed in this project is determining where future Uber pick-ups will be. Using Uber data sets that document transactions from April 2014 to September 2014, a variety of machine learning algorithms were applied to predict the geographical location of Uber pick-ups in New York City, in September 2014. After a detailed process of refining, training, and testing the data, we were able to make predictions of Uber pick-up locations, by longitude and latitude coordinates, within 88 percent accuracy.

Introduction
Currently, Uber drivers have no way of knowing when and where their next Uber pick-up will be. Drivers typically go to a specific location that gets busy at a certain time of the day, and will wait for an Uber request. Records and statistics of past Uber transactions may show correlations between certain areas, such as particular boroughs in New York City, and certain times (morning, afternoon, or evening). However, there is no way of predicting where future pick-ups may be. Using relevant data sets and a range of machine learning methods, our project aims to make predictions of the longitude and latitude coordinates of Uber pick-ups for September 2014 in New York City.

Approach
The data pre-processing part involved cleaning empty data, converting the locations from address to latitude and longitude. We are trying to predict based on latitude, longitude and hour of the day for the months of September based on August month data who is most likely to get the ride that is Uber or Lyft(as competitor).Using different supervised learning algorithms we tried to predict, but as Uber data dominated than that of Lyft data the predictions. Uber hires vehicles from companies referred as Base Companies for pick-up. We tried predicting which company is more likely to pick up at certain location on particular hour of the day. For both the predictions mentioned above we implemented K means clustering which gave 55% maximum accuracy. To refine the predictions we converted the latitude and longitude from float value to integer to get precise clusters.

ITCS 6156 – Machine Learning
Final Project Report
Group 12 – Pritam Borate, Puneeth Devabhaktuni, Aditi Helekar, Lily Naoudom

Materials and Methods

The primary language used in this project is in Python. The libraries used are from Sci-kit Learn, an open-source library archive that implements a range of machine learning algorithms. The machine learning algorithms used in this project to predict Uber pick-up locations are Decision Tree Classifier, Gaussian Naïve-Bayes, Neural Networks, AdaBoost-Ensemble, and K Means Clustering.

Results

The table below shows comparison of accuracy obtained from various approaches:

ALGORITHM USED	ACCURACY
K Nearest Neighbors with 200 Neighbors	88.40%
K Nearest Neighbors with 20 Neighbors	87.99%
K Nearest Neighbors with 5 Neighbors	86.76%
Gaussian Naïve Bayes	87.36%
AdaBoost Classifier with Decision tree	88.81%
K Means Clustering	55.71%

1. **K Nearest Neighbors**

```
Model is -
Prediction is -
KNN..
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=200, p=2,
           weights='distance')
0.884077417662
[[604236  72328]
 [  8967  15756]]
[ 0.89309511  0.6373013 ]
[ 0.98537678  0.17887471]
--- 575.9977898597717 seconds ---
```

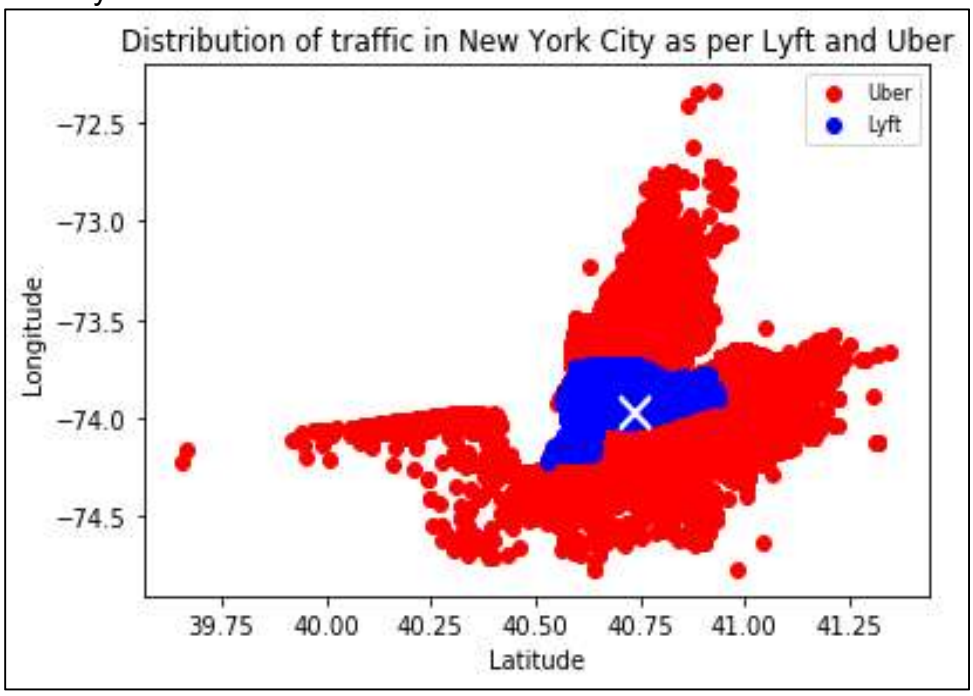
Using K Nearest Neighbors algorithm we could get accuracy of 88.40% as seen from the figure above with 200 nearest neighbors. The distance between the locations was used as metric for the analysis.

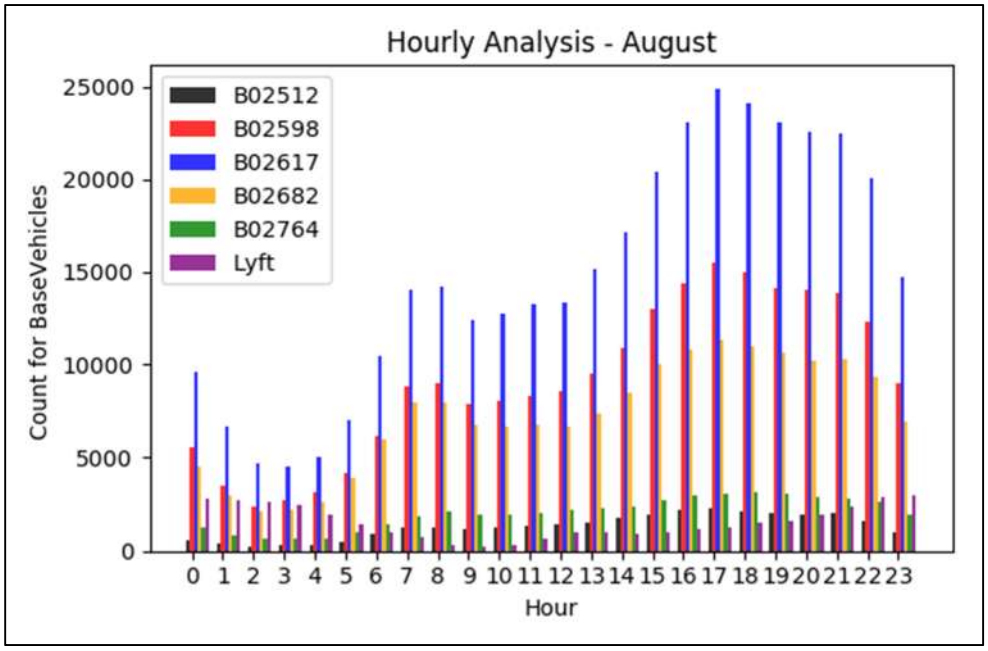
2. K Means Clustering

K Means Clustering could give accuracy of 55.71%.

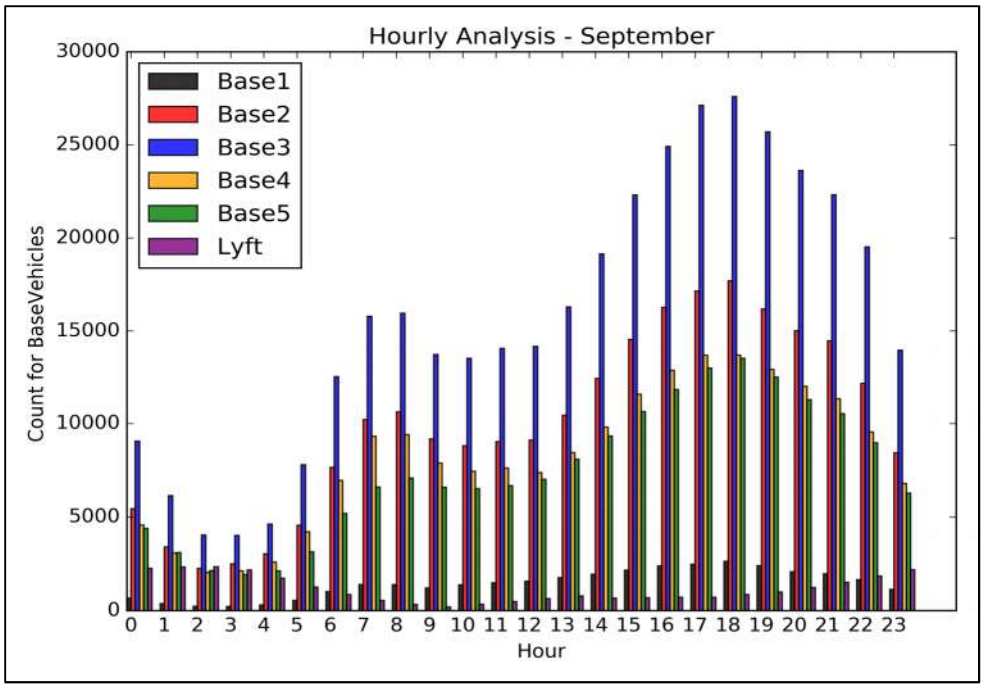
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=1000,
    n_clusters=2, n_init=20, n_jobs=1, precompute_distances='auto',
    random_state=None, tol=0.0001, verbose=0)
Centroid:  [[ 40.00141629 -73.17486685   6.66918561]
 [ 40.00134359 -73.19791645  18.07582263]]
Labels:  [1 1 1 ..., 1 0 0]
Accuracy:
0.557143341151
Confusion matrix:
[[     0 336854  67845]
 [     0 591996  65861]
 [     0      0      0]]
```

From the scatter plot of the locations of Uber and Lyft pickups we could identify that Lyft was majorly active in central part of the city where as Uber was spread out across the city evenly specially in the Manhattan and Brooklyn areas.





The graph plot above shows the distribution of number of vehicles of the base companies dominant with respect to hour of the day. The base company B02617 has most number of vehicles all over the day. The peak hours in the evening show a substantial rise in the number of vehicles due to the increased number of pickups. We have plotted the distribution for the month of August and September 2014. During this period we can see significant change in the use of taxis in new York City.



ITCS 6156 – Machine Learning
Final Project Report
Group 12 – Pritam Borate, Puneeth Devabhaktuni, Aditi Helekar, Lily Naoudom

Analysis

Supervised learning techniques used with the results are as follows:

1. K Nearest Neighbors
   a. Neighbors = 5

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=5, p=2,
           weights='distance')
0.867654797748
[[1058435  112708]
 [  56041   47883]]
[ 0.90376239  0.46075016]
[ 0.94971538  0.29816739]
--- 165.52366375923157 seconds ---
```

   b. Neighbors = 20

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=20, p=2,
           weights='distance')
0.879923956937
[[1083447  122076]
 [  31029   38515]]
[ 0.89873607  0.55382204]
[ 0.97215822  0.23983287]
--- 189.1745686531067 seconds ---
```

   Changing the number of nearest neighbors for KNN did not significantly affect the accuracy. The maximum accuracy without overfitting was obtained with 200 nearest neighbors.

2. Naïve Bayes

```
GaussianNB(priors=None)
0.873681147736
[[1111876  158465]
 [   2600    2126]]
[ 0.8752579   0.44985188]
[ 0.99766707  0.0132386 ]
--- 2.5567269325256348 seconds ---
```

   With Naïve Bayes we couldn't use Multinomial approach as we have negative values in latitude and longitude so we opted for Gaussian Naïve Bayes. It predicted accurately with 87.36% accuracy. The probabilistic model depends on the training data and it tends to predict values with high occurrence which in turn leads to predicting 'Uber' rather than 'Lyft' for

most of the test data. This is not acceptable scenario so we shifted to KNN approach for better results.

3. AdaBoost with Decision Tree classifier

```
AdaBoostClassifier(algorithm='SAMME.R',
          base_estimator=DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=100,
            max_features=None, max_leaf_nodes=None,
            min_impurity_split=1e-07, min_samples_leaf=1,
            min_samples_split=2, min_weight_fraction_leaf=0.0,
            presort=False, random_state=None, splitter='best'),
          learning_rate=0.05, n_estimators=100, random_state=None)
0.888128231693
[[1066477   94645]
 [  47999   65946]]
[ 0.91848832  0.57875291]
[ 0.95693133  0.41064568]
--- 1610.0283188819885 seconds ---
```

The AdaBoost with Decision tree approach predicted with accuracy of 88.81% with overfitting issues.

The code snippet for supervised learning approaches is shown below:

```
start_time=time.time()
model = KNeighborsClassifier(n_neighbors=200, weights='distance', radius=5)
model = DecisionTreeClassifier(max_depth=100)
model = GaussianNB(priors=None)
model = AdaBoostClassifier(DecisionTreeClassifier(max_depth=100),n_estimators=100,learning_rate=0.05)
model.fit(X_train,y_train)

print(model)
#test=np.array(testData[["Lat","Lon"]])
prediction=model.predict(X_test)
print(accuracy_score(prediction,y_test))
print(confusion_matrix(prediction,y_test))
print(precision_score(y_test,prediction, average=None))
print(recall_score(y_test,prediction, average=None))
print("--- %s seconds ---" % (time.time() - start_time))
```

The confusion matrix obtained for KNN with 200 neighbors has predicted the true positives for Uber and Lyft correctly as compared to that of other techniques. The precision suggests how accurately if predicted from the actual values which is obtained as desired for KNN.

The code snippet for K means Clustering is shown below:

```
X = trainData[["Lat","Lon","hour"]]
y = trainData[["taxi_code"]]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=42)

from sklearn.cluster import KMeans

for k in [2]:
    uberCluster = KMeans(n_clusters=k, init='k-means++', n_init=20, max_iter=1000)
    print(uberCluster)
    model=uberCluster.fit(X_train, y_train)
    predictions = model.predict(X_test)
    #pred_cluster_centers = [uberCluster.cluster_centers_[i] for i in predictions]
    centroids = uberCluster.cluster_centers_
    print("Centroid: " , centroids)
    labels = uberCluster.labels_
    print("Labels: ", labels)
    #print("Score: " + model.score(predictions))
    print("Accuracy: ")
    print( accuracy_score(predictions,y_test))
    print("Confusion matrix: ")
    print(confusion_matrix(predictions,y_test))
```

Prior Research

Previous research has been to done to determine whether Uber uses its own developed prediction algorithm for rides or an existing one. Rohit Joshi explains that Uber is “aiming for an algorithmic monopoly, control of a market through contract pricing.” Over the years, statistics show that Uber is gradually displacing the “open cab market”, especially as the number of people who use Uber has increased. With more people using Uber, there will be a lower demand for taxi cabs, and therefore lower taxi cabs in business, putting Uber at the top of the race. Uber is attempting to not only infiltrate regulated cab markets, but to eliminate them. However, Joshi explains the issues with Uber’s model. The first issue is that Uber controls all of the information in the market, both from the buyer (client) and the seller (driver) (3). Since Uber functions as the middle-man of the operation, it has the power to regulate or control what information is shown to the buyer and seller, demonstrating an algorithmic monopoly. Other issues include history of discrimination from cab drivers, and the threat of Uber as a company gaining too much power over its drivers.

Another article discusses factors in the algorithm that Uber uses when assigning rides to its drivers. In regards to the Uber App, a driver will get a request if they satisfy three main criteria: their availability, their proximity to the request, and if they have a 4-star review or higher (4). Samuel Feurer explains potential flaws in this model. The first is the proximity of the driver to the request. Even if there is another Uber driver within a few feet away, a request from the client will be sent

to the Uber driver that is closest to their location. GPS technologies that are integrated into the Uber App may help guide the driver to get to their client's destination, but it could also limit the number of requests a driver could receive if there are other Uber drives in closer proximity to the request. In addition, the rating system that the Uber App uses plays an important part in the number of requests a driver may get. If a driver has a rating of lower than 4 stars, the system will just skip that driver and the request will go to the next closest available driver.

The biggest flaw with both of these models in predicting Uber pick-ups is that they overlook the business platform that Uber thrives on – which is allowing both Uber drivers and Uber clients to have control over their transactions. For instance, drivers make up their own schedules for when they are out looking and waiting for potential clients. Some Uber drivers have a primary job during the day (morning and afternoon), and will start their Uber driving schedule in the evening. Even if the busiest time and location for requests in New York City was Manhattan, some Uber drivers will not be available at that time or at that location. On the other hand, clients have the option to accept or decline any driver (and vice versa) for any reason. A full, complete transaction involves the consent and cooperation of both the driver and the client.

Conclusion

The main idea of the project is to apply Machine learning algorithms to classify and predict likelihood of UBER pick-up based on latitude and longitude coordinates and hour of the day provided in the data sets and even compared it with the lift data we have. We divided the UBER data with respect to its base company dispatchers. We observed that the UBER pickups are spread out in the New York city as compared to Lyft pickups. Lift pickups are clustered in the central part of the city. By using K Nearest Neighbors and distance as weight we are able to get an accuracy of 88% for the UBER Pickups. The base company B02617 has most number of vehicles all over the day. The peak hours in the evening show a substantial rise in the number of vehicles due to the increased number of pickups. With Naïve Bayes we couldn't use Multinomial approach as we have negative values in latitude and longitude so we opted for Gaussian Naïve Bayes.

By using our model UBER have a chance to improve their business by locating where they are getting more rides at what times of the day and they can encourage more drivers to be in that particular area at that particular hours. The future scope of the project is to include the number of active vehicles on the road which belongs to UBER or Lyft and make the model predict which company have more likelihood of getting the ride.

ITCS 6156 – Machine Learning
Final Project Report
Group 12 – Pritam Borate, Puneeth Devabhaktuni, Aditi Helekar, Lily Naoudom

References

1. Mitchell, Tom. Machine Learning. 1997 March 1. McGraw-Hill.
2. Kleiman, Iair. 2017 January. Uber Plots, Heatmaps, and Tables. Retrieved from https://www.kaggle.com/ikleiman/d/fivethirtyeight/uber-pickups-in-new-york-city/uber-plots-heatmaps-and-tables
3. Joshi, Rohit. 2014 December 13. Does Uber use its own-developed prediction algorithm or did it use an existing one? Retrieved from https://www.quora.com/Does-Uber-use-its-own-developed-prediction-algorithm-or-did-it-use-an-existing-one
4. Feurer, Samuel. 2016 March 19. What are the factors in the algorithm Uber uses to assign rides to drivers? Retrieved from https://www.quora.com/What-are-the-factors-in-the-algorithm-uber-uses-to-assign-rides-to-drivers

Distribution of Work

Team Member	Contributions
Lily Naoudom	Topic research, project poster, paper
Pritam Borate	Topic research, Implementation, Analysis
Aditi Helekar	Topic research, paper, Implementation, Analysis
Puneeth Devabhaktuni	Topic research, project poster, paper