Lisa Taylor
Capstone 1 **Final Report: Ridesharing in Chicago**

1. Problem Statement

Since November 2018, rideshare companies (Lyft, Uber, etc.) operating in the City of Chicago are required by ordinance to submit periodic data reports with basic rideshare information. For this project, I have taken the perspective of a city transportation agency worker applying this dataset to gain insight on travel patterns within the city. Understanding how rideshare customers are utilizing these services provides opportunity to target improvements in public transit offerings, such as making enhancements to bus/train routes and schedules. Insight into rideshare patterns also provides information that can be applied to infrastructure planning and managing transportation construction projects.

The primary questions that I am trying to answer with the rideshare dataset are:

- Can rideshare customer segmentation reveal patterns of rideshare usage that may be better served by a targeted transit option, such as a shuttle bus?
- Can rideshare usage be predicted by census tract for a given day of the week and time?

In addition to city transportation agencies, these kinds of questions are also of interest to rideshare drivers, who could apply a better understanding of rideshare utilization patterns to optimize where they operate at different times of day. Environmental agencies and citizen environmental groups concerned with minimizing air emissions and traffic impacts associated with excess driving may also benefit from these insights.

2. The Data

The main ‘Trips’ dataset includes basic information about individual rides, including: starting and ending location, starting and ending time, whether trips are pooled, and information about fares and tips. Times are rounded to the nearest 15 minutes. Fares are rounded to the nearest $2.50 and tips are rounded to the nearest $1.00. The Trips data are anonymized, so trip starting and ending locations are generalized to the nearest census tract, no customer data is available, and drivers cannot be linked to particular rides they provided. The census tract data were not available for certain rides (census tracts outside the city boundary or with low frequency of usage) and those ride records were dropped from the dataset. Other data tables provided by the City pertaining to driver and vehicle characteristics were not applied to this evaluation.

Data Preparation

The Chicago Trips dataset was downloaded as a csv file from the City of Chicago Data Portal. Since the data file is very large (12 GB), the exploratory data analysis was performed on a sub-sample of the entire dataset. This was accomplished by setting the “chunksize” option of the pandas read_csv function to create an iterator of the csv file that reads in 20,000-line increments, then sampling 5% of each chunk into sample dataframe.

Some additional wrangling steps I performed on the data included:

- Creating datetime fields with the ride start and end datetimes rounded to the nearest hour;
- Removing rides with zero distance or zero fare;
- Creating additional fields for day of week, month, year, and hour; and
- Aggregating the dataset by hour and census tract and calculating average fare, average distance, average duration, and frequency counts

To enrich the rideshare dataset, I merged in additional data resources that may enhance prediction of trip demand and fares. A weather dataset from NOAA containing hourly and daily aggregated measurements of temperature and precipitation was merged with the Trips data by matching on the ride pickup datetime field. Additional information on population characteristics was obtained from the US Census by using the Census API to download median income and population density per census tract. Census tract and Community Area[1] boundary geojson files were also downloaded from the Chicago Data Portal and loaded into geodataframes using the geopandas python library. The geodataframe data structure allows the data and analysis results to be easily mapped by census tract or Community Area, and enables spatial operations to be performed on the data.

[1] Community Areas are groupings of neighborhoods used for planning purposes by the City. They are larger than Census Tracts, approximately 3 square miles.

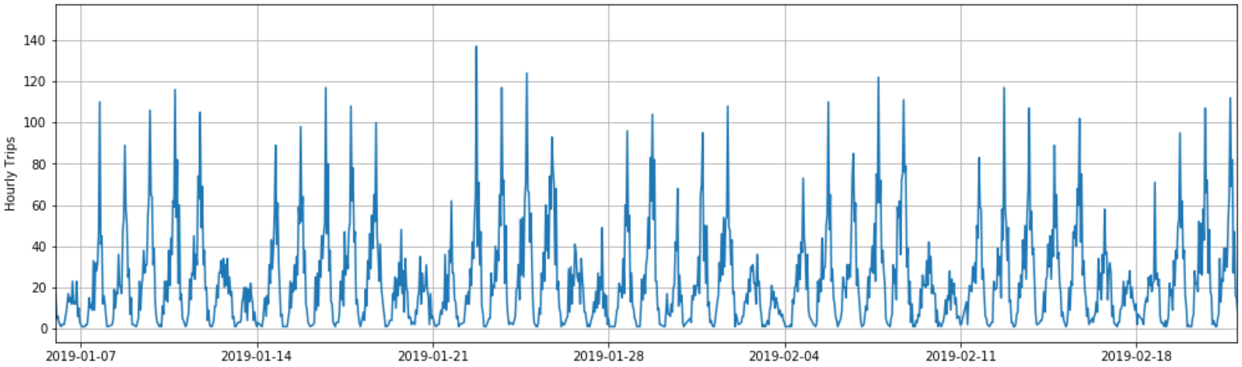
After joining the census geodataframe to the aggregated rideshare and census data I was able to and derive additional fields including population density and distance from downtown Chicago. Since the original geographic data (census tract polygon boundaries and centroid coordinates) were provided in decimal degrees, deriving these fields required transforming the spatial data from the web mercator to the UTM coordinate system for accurate calculations of distance and area. These transformations and calculations were all done with the geopandas library.

One census tract appeared to have an unusually high population density. To check if this was an error, I located the tract in google maps and confirmed that it corresponded to a census tract covering a very small area but housing a series of 40-story residential towers. Given this construction, the population density seems reasonable for this tract.

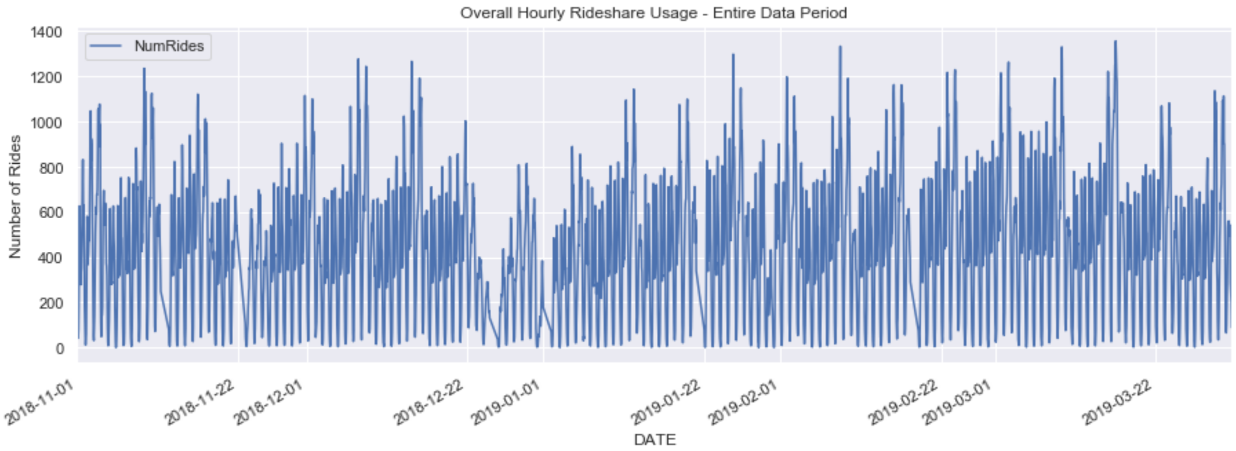
3. EDA and Statistical Analysis

Exploratory Data Analysis (EDA) was performed to gain insight into the dataset.[2] By plotting and mapping the enhanced rideshare dataset I was able to make some preliminary findings:

1) Rideshare usage is cyclical, with a usage pattern demonstrating both daily and hourly components. The following plot shows usage for one census tract over several weeks in January and February 2019, with vertical gridlines indicating midnight on Sundays.

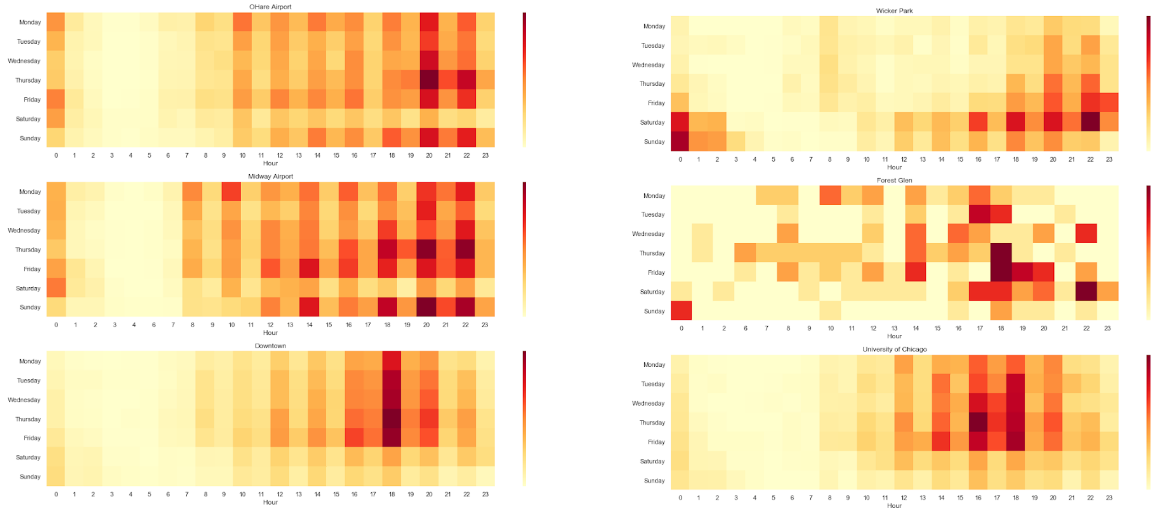


The next plot shows systemwide hourly usage over the entire data period. We see that overall usage is stationary (constant mean), and, aside from holidays, there is little variation in the magnitude of usage between weeks.



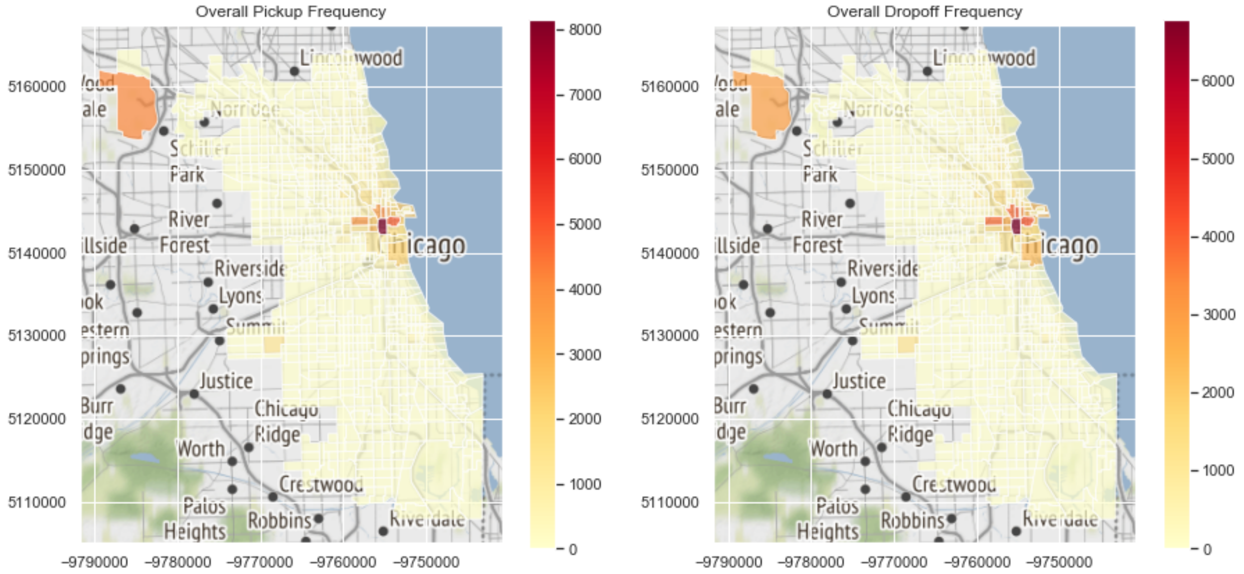
2) Rideshare usage patterns vary across the city. The data demonstrates distinct patterns for downtown, airports, residential neighborhoods, etc. The following heatmaps illustrate usage patterns for a few areas within the city, and show distinct variations between airports, downtown and a neighborhood with popular bars and restaurants.

[2] Some of the visualizations and statistical tests presented here don't necessarily pertain to the main business question, but were done to gain practice with inferential statistics.



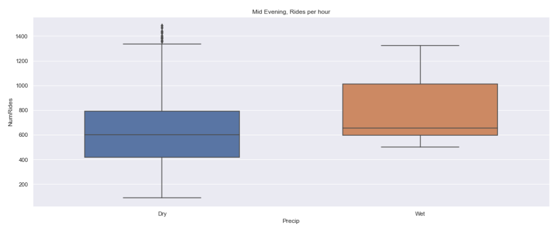
.

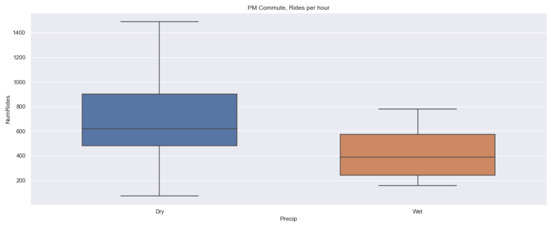
3) The highest rideshare usage occurs within the downtown core and to/from airports, and drops off quickly away from these areas. The following chloropleth map, which displays the sum of total pickups and dropoffs by census tract, illustrates this pattern clearly:



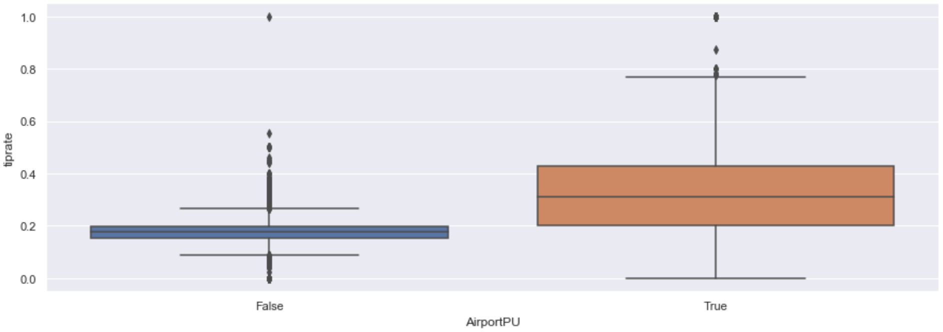
4) The relationship between rideshare usage patterns and weather is not obvious. When viewed at the daily level, rideshare utilization patterns do not appear sensitive to either temperature or precipitation. Zooming in to the hourly level, there appears to be no statistical relationship between average ride duration and precipitation, and only a small statistically significant effect (decrease) in average ride distance. If we isolate a more specific time window (weekday evening commute, post-commute hours) we can

isolate small effects such as shown below, with utilization declining during the commute period, and then increasing in the post-commute period.

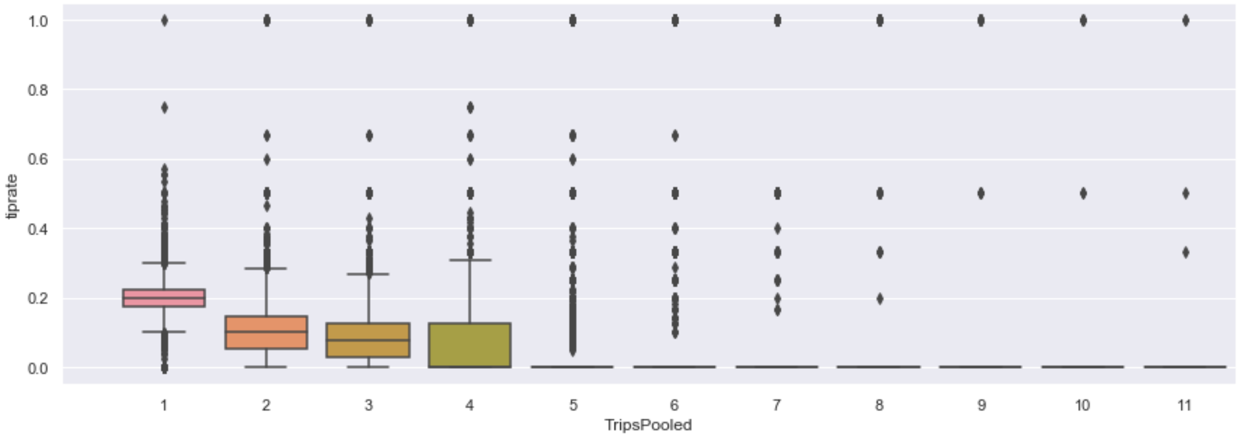




5) Riders tip the driver on approximately 20% of rides. Factors influencing the likelihood of a tip include whether the ride originates at the airport, and whether the ride is pooled. A two-sample z-test of the proportion of tipped rides between airport/non-airport rides demonstrated a statistically significant increase in the odds of tipping for airport pickups. The following box plot, which shows the percentage of tipped rides after the data was sampled into hourly bins, illustrates this effect.



The following boxplot illustrates the distribution of tipping rates for pooled and unpooled rides, with the number of riders in the pool increasing towards the right (TripsPooled=1 indicates the ride was not pooled, though the rider may have authorized pooling). As we move to the right we see the average tipping rate (percent of rides with tips) declining. This suggests price-sensitive customers that are willing to take on a pooled ride are less likely to tip.



6) Given the nature of the Trips dataset, there exist strong relationships between many of the parameters. For instance, the Pearson correlation coefficients between both trip distance and duration (0.80) and trip distance and fare (0.84) indicate strong correlations between these variables. We also see evidence of an inverse correlation (-0.53) between distance from downtown and population density, indicating a decrease in population density as you move away from downtown.

4. Application of Machine Learning

The EDA and statistical evaluation uncovered many interesting patterns within the Chicago Rideshare dataset. Following on this work, I applied machine learning tools to target two business goals associated with the project. These goals included:

1) Gaining insight on rideshare usage patterns; and

2) Developing the capability to predict rideshare utilization at different locations and times.

Rideshare Customer Segmentation With Clustering

A clustering model was developed to gain a better understanding of how ridesharing is being used in Chicago. Unlike a more traditional geospatial clustering analysis, which would identify locations with relatively high or low utilization rates across space and/or time, I instead performed a customer segmentation analysis, seeking to find groupings of rideshare routes with common usage patterns.

For this analysis, the rides data were grouping on the following fields: Community Area of Pickup, Community Area of Dropoff, the period within the day (morning, mid-day, afternoon, evening, late night, early morning), whether the ride occurred on a weekend or weekday, and whether the ride involved the airport at either end. The “Community Area” within the City was used to spatially aggregate ride counts, rather than the census tract, to reduce the number of potential routes in the analysis. The timing of the ride was also simplified into larger buckets (weekend/weekday, period of day) to improve interpretability of outcomes. A boolean airport flag was added since this attribute was found during EDA to have a strong influence on rideshare utilization. Holidays were excluded from the analysis. Within each grouping, the following aggregated values were calculated: number of rides, average ride duration, average ride distance, and average fare. Since the source dataset is a 5% sample of the larger Trips dataset, the field indicating number of rides is not an accurate counting of rides, but rather a relative measure of demand representing roughly 5% of the actual ride count.

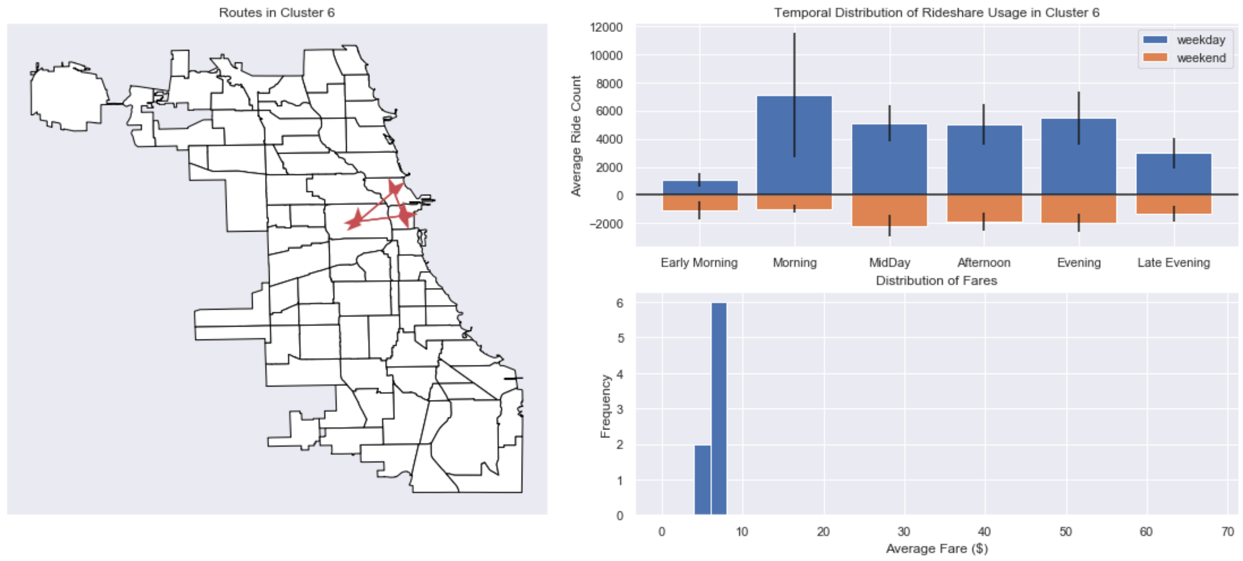
Before applying the clustering algorithm, a quick query was performed on the grouped dataset to identify the most commonly-used rideshare routes within the City. By simply reverse sorting the grouped data by number of rides, it was observed that the top 10 results were dominated by weekday commute-period rides with relatively short distances within the downtown area of the City.

To prepare the data for fitting a clustering model, the grouped dataset was pivoted to create columns containing ride counts for each grouping of weekend/weekday and day period. After this reworking, the final dataset contained a single record for each route, each with a MultiIndex containing the unique combinations of dropoff and pickup Community Areas, and columns containing the ride count columns, the airport flag, and the average fare for the route.

The KMeans clustering model initially selected for the analysis. This model requires the user to specify the number of clusters (k) before fitting the model. To identify an appropriate value of k, the inertia (within-cluster sum of squares) was calculated for values of k between 5 and 20. A plot of this value showed a small elbow near k=7. The silhouette score and associated plots were also generated for a variety of values of k. The results suggested a plateau in the silhouette score above k=5, however the plots demonstrated that several very large clusters begin to break up into smaller groupings around k=11.

KMeans was fit to the dataset using a variety of k values. The data was scaled prior to input to the algorithm. The result at k=7 showed interesting patterns and was used for the final interpretation of results.

A dashboard was built using IPywidgets to facilitate the evaluation of results. The dashboard has a single widget, for selection of the cluster, and outputs visualizations of the routes, distribution of fares, and temporal utilization patterns associated with each cluster. An example, for cluster 6, is shown below:



With k=7, the results suggest:

- Strong commuting patterns were identified near the downtown area. KMeans identified one cluster (8 routes) near downtown associated with heavy morning commute period usage, and one cluster (11 routes) near downtown associated with heavy evening commute usage. A third downtown cluster shows very heavy usage, particularly during both commute periods, within a single Community Area at the city center (1 route). These three routes all represent groups of rides with relatively low fares.
- A cluster was identified with airport pickups.   This cluster demonstrates a weekday evening peak, and a wide distribution of fares.
- A cluster was identified demonstrating commute patterns with higher fares relative to those in the immediate downtown corridor. This cluster also includes an isolated set of rides in the

southeast area of the city (Community Area 41) which appears to represent rideshare use associated with the University of Chicago.

- The remaining two clusters contained routes with low frequency of usage and higher fares. Routes in the first of these exhibited a commute-type pattern with fares ranging between $5 and $20. Routes in the final cluster had average fares generally exceeding $20 dollars without a distinct temporal pattern.

These results suggest that ridesharing most heavily utilized within the urban core, for short, relatively cheap rides associated with commuting. The City can utilize this information to make improvements to transit services, such as adding a shuttle service within downtown to help rail commuters efficiently reach their final destinations.

In addition to KMeans, other clustering models available in Scikit-Learn (i.e DBSCAN, Agglomerative Clustering, Spectral Clustering) were applied to the rideshare dataset. These algorithms produced similar results to KMeans. Other variations on this analysis included using different parameters to represent typical ride characteristics for a route (i.e. distance, fare, duration), however given that these values are correlated they tended to produce similar outcomes. Finally, given the evidence from silhouette scoring that additional structures may become evident at higher k values, clustering was run for k values up to 19. The outcomes produced additional groupings and present an opportunity to develop additional insights on rideshare patterns.

Rideshare Usage Prediction Model

For this evaluation, the Trips data was applied to develop a model for predicting rideshare usage for a given location as a function of day and time.

As with the clustering model, the original Trips data was reworked by grouping and aggregation. An hourly aggregated rideshare dataset was generated by grouping the raw rides data by pickup census tract and ride starting time rounded to the nearest hour. The fare, tip, and distance of each ride were averaged over each time/space grouping. A ride counts field was generated for each grouping, however as with the aggregated data used for the clustering model, these counts are calculated on 5% of all rides, and represent more of a relative measure of rideshare usage. As previously mentioned, the aggregated Trips data was enriched with hourly weather data, census-derived data (median income, population density) and geographic data (distance to downtown). The final aggregated dataset has over 500,000 records.

The EDA of the Trips dataset suggested strong cyclical effects in rideshare usage at the hourly and daily levels, and a strong relationship between demand and distance to downtown at certain times of day. The EDA did not show evidence of a strong trend in overall usage over the measurement period (i.e. mean usage is stationary). Aside from holidays, there also appeared to be little variability in usage measured over the same time period in different weeks (i.e. variance of usage is stationary). Given these patterns, I attempted to predict usage as a function of distance from downtown, a general date and time (i.e. weekend or weekday, time of day), and other variables which EDA suggested may influence demand (weather, etc).

In addition to these parameters, some new engineered features were added to the dataset to see if they aid in predicting utilization:

- **IsAirportPU:** The preliminary data analysis indicated that the airports have different patterns than other parts of the city. This flag identifies rides that originate at an airport. Since we are interested in pickup demand, not dropoff demand, this field was only derived for pickups.
- **IsHoliday**: The time series plots generated for EDA demonstrated a dropoff in utilization for mid-week holidays. This flag was added to enable dropping these days from the dataset.

Linear Regression Model

A variety of simple linear regression models were fit to the entire dataset, with the goal of evaluating whether this method could be effective at predicting ride counts as a function of ride pickup characteristics (location, time of day, day of week, etc).

A major issue with this approach is that the response variable, NumRides, is not normally distributed. Since NumRides measures counts of ride pickups, it is an integer field and always greater than zero. Distribution plots of NumRides suggest it may be a zero-truncated Poisson random variable, which has a minimum of 1. There is no out-of-box GLM model that can fit this distribution in statsmodels, so I abandoned this approach.[3] To prevent predicting negative ride counts, I fit the OLS model to the log of NumRides.

Linear models require that the independent variables be uncorrelated. Several of the features in the original Trips dataset are strongly correlated (i.e. fare, trip duration, trip distance). Due to these strong relationships I only incorporated a single one of these features in each attempted regression model to avoid multicollinearity. Linear models also require that there be no temporal or spatial autocorrelation in the error terms, which could be an issue with this dataset which has both spatial and temporal components.

Although the OLS model is not the best choice for this situation, and the approach was not expected to perform well.

The linear regression models were implemented using the OLS function implemented in statsmodels. Statsmodels was selected due to the detailed summary report it provides that can be used to interpret the model outcome, and the ease with which you can attempt different models and transform categorical parameters using the formula API. I was also interested in inferring the relative importance of different parameters and their influence on rideshare utilization.

Many OLS and Poisson regression models were attempted in statsmodels using varying combinations of the input parameters, including interactions and transformations of the

[3] I also attempted to fit a negative binomial GLM to a version of the dataset that includes zero counts. This is not discussed in this report.

parameters. The best OLS outcome was achieved by regressing on fields DistToDowntown, TripTotal (average total fare), and categoricals Precip, IsAirportPU, DayPeriod, and IsWeekday.

Although the p-values of the regression coefficients generally suggested that the parameters were statistically significant, the overall predictive power of the attempted OLS models were low, with R-squared values below 0.26. Since this method did not appear effective at explaining ride counts, no further evaluation was performed, such as applying regularization with cross-validation.

Non-Parametric Model

Given the poor performance of the linear regression approach, I switched directions to a non-parametric modeling approach to predict rideshare usage. Unlike regression, non-parametric models make no assumption about the functional form of the relationship between the parameters and the dependent variable, and have the potential to perform better in this application.

Non-parametric models that may be applied to this problem include k-Nearest Neighbors regression, RBF-kernel SVM regression, and tree-based regression approaches (i.e. decision trees, bagging, boosting, random forest). Given the large dataset available to train the model, the tree-based approach was selected. An advantage of tree-based models is that they require very little data preparation. I started with out-of-box Random Forest, Gradient Boosting and Bagging regressors as implemented in scikit-learn. These are ensemble models that are trained by fitting a series of decision trees to bootstrap samples taken from the training dataset.

Since the data is a time series, rather than using a random train-test-split, the dataset was divided into before (80%) and after (20%) periods to be used as training and test datasets. In addition, holidays and airport pickups were removed from the dataset. Each baseline model was trained on 80% of the dataset using parameters 'DistToDowntown', 'Precip', 'Hour', 'IsWeekday', and 'TripMiles' and fit to the ‘NumRides’ field and scored on the test dataset.

The results of each out-of-box model, as evaluated on the test dataset, are compared below:

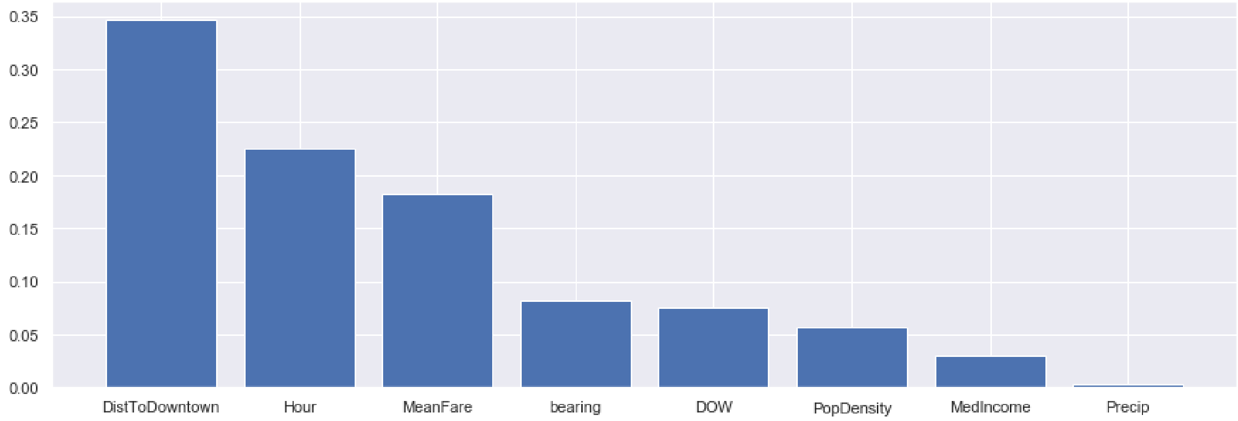
Baseline Model	**R-squared**	**RMSE**
Random Forest	0.81	1.9
Gradient Boosting	0.68	2.5
Bagging	0.81	1.9

The results demonstrate that rideshare pickup rates in Chicago for given locations and times can be fairly accurately predicted using either a random forest model or bagging model given a few pieces of information.

Parameter Selection

As previously mentioned, the EDA suggested that rideshare usage is focused on downtown, with a sharp dropoff in utilization as the pickup spot moves outwards from the city center. The baseline model was built using distance from downtown as the only spatial parameter. For this enhancement, a "Bearing" field was added to the dataset indicating the direction of each pickup spot relative to downtown, in degrees from North (i.e. a pickup location due west of downtown has a bearing of 270). In essence,the xy coordinates of the pickup locations have been converted to radial coordinates expressed as distance and angle relative to downtown Chicago. The code used to perform this coordinate transformation is provided in my python module RideshareDataPrep.py.

A RandomForestRegressor with default parameters was fit to the enhanced dataset using fields ‘DistToDowntown','Bearing’,’Hour’,’DOW’,'Hour',’PopDensity’, ‘Precip’ and ‘MeanFare’. ‘MedIncome’ was correlated with ‘DistToDowntown’ (Pearson r = -0.47) and not included. The model fit scores were better than those produced by the baseline model, with an R-squared of 0.84 and RMSE of 1.77. Bagging performed similarly. By inspecting the feature_importance attribute of the model (weighed decrease in variance for nodes split on that parameter, averaged over all trees), we see that distance to downtown is the most important parameter in the model, followed by hour, mean fare, day of week, and bearing.



Trial and error experiments on the parameter set input to the random forest regressor found the following set of parameters to produce the best fit while avoiding unhelpful parameters: DistToDowntown, Hour, MeanFare, bearing, DOW and PopDensity.

Hyperparameter Tuning

The Random Forest Regressor is trained by fitting a series of decision trees to bootstrap samples taken from the training dataset. Because we are predicting ride demand using time series data, we cannot use the out-of-bag (OOB) sample (portion of the dataset not included in the bootstrap sample) for cross-validation because the missing samples will be highly correlated with other samples adjacent in the time series that are included in the bootstrap.

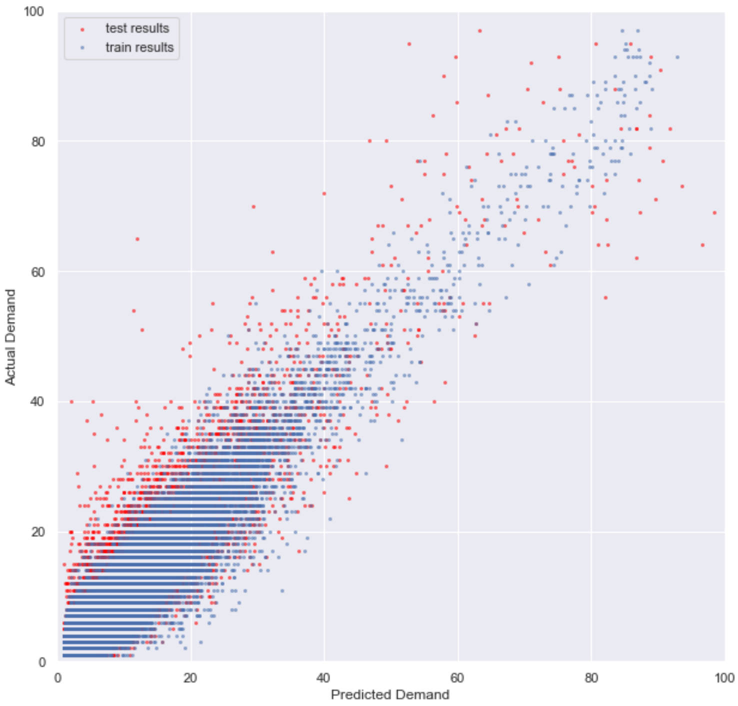
Instead, the training data were partitioned into four contiguous portions using the TimeSeriesSplit function in scikit learn. Over three iterations, the model was trained on the early portion (subsets 1, 1+2, and 1+2+3), and then scored on the equally-sized later subsets (2, 3, and 4) of the training data. The resulting R-squared scores for each iteration were averaged to produce the final cross-validation score.

The following series of cross-validation evaluations were performed to select the best hyperparameters for the Random Forest Regressor (future work: combine into one CV with three loops):

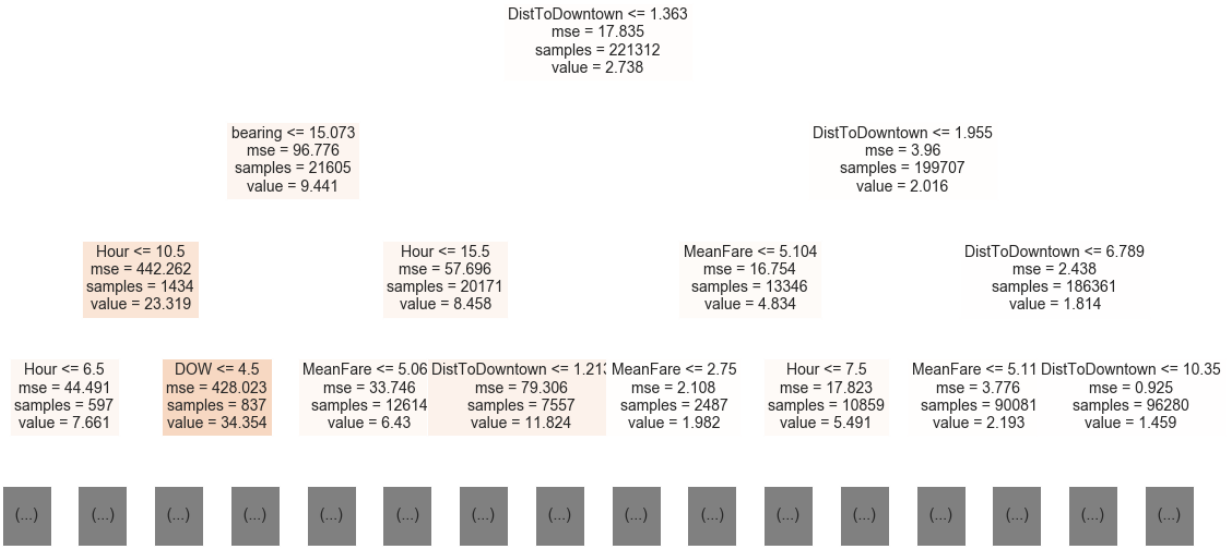
- Optimize number of estimators: Models were fit with 10, 20, 30 and 50 estimators. The resulting R-squared values were 0.825, 0.831, 0.832, and 0.834, illustrating that adding estimators produces modest improvements to the test score. Higher values of the parameter were not attempted due to long run times.
- Optimizing max_features and tree depth: A gridsearch-type optimization was performed using different combinations of parameter selection method (max_features = log, square root, or all) and tree depth. The number of estimators was set to the default (10). The best performance was achieved with allowing all parameters at each branch (equivalent to bagging), and 16 levels of tree depth.

A final version of the model was prepared with n_estimators set to 50, max_depth set to 16, and max_features set to None. The resulting R-squared and RMSE values were 0.85 and 1.69, respectively.

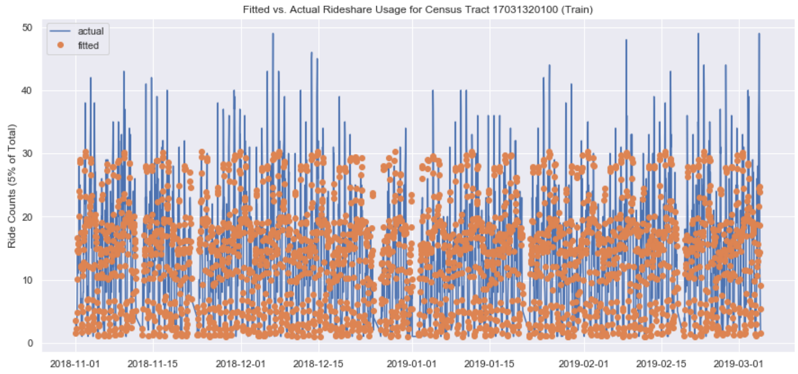
The following scatter plot shows compares the predicted and actual ride counts for the first 50,000 records of the training and test datasets. The alignment of the data along x=y demonstrates that the model fits both the training (blue) and test (red) data fairly well.



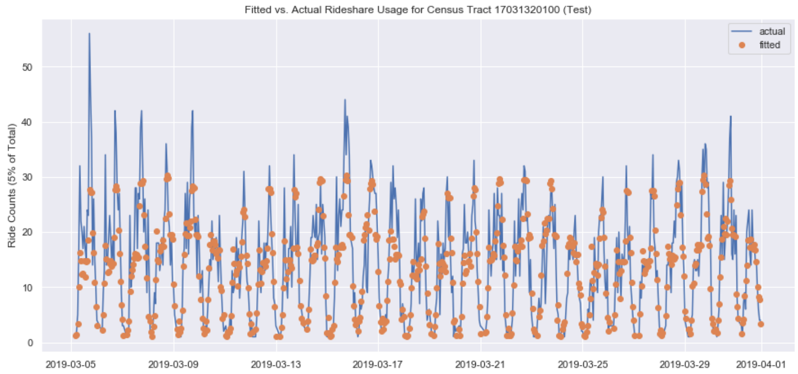
The following plot illustrates the top three levels of one decision tree fit by the model:



The following plot demonstrates the model fit to the training and test datasets for a single census tract near downtown Chicago:



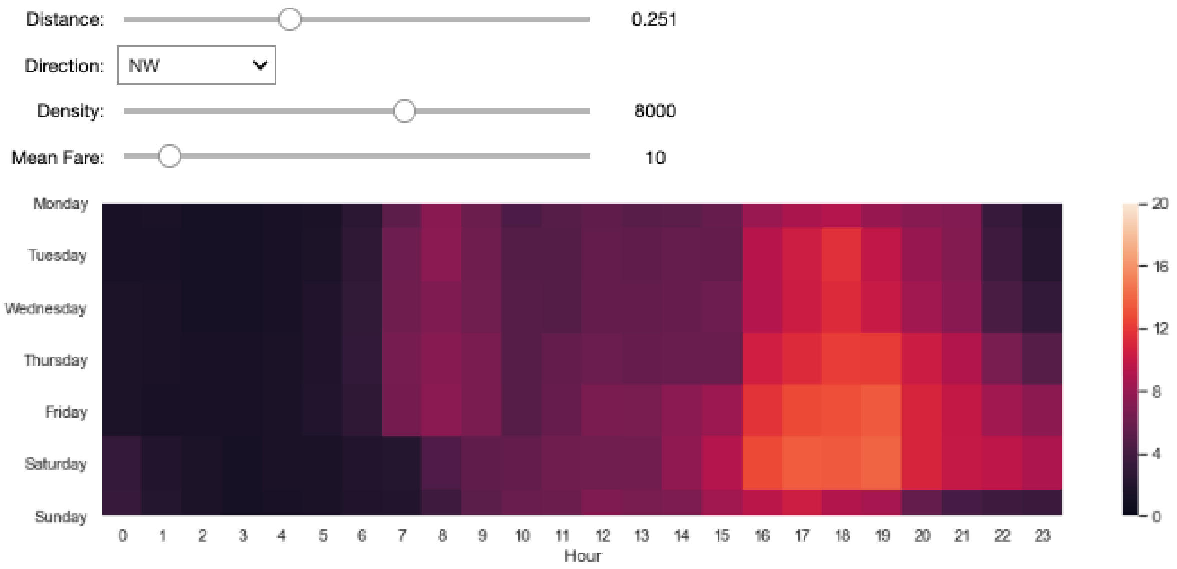
Fitted vs. Actual Rideshare Usage for Census Tract 17031320100 (Train)



Fitted vs. Actual Rideshare Usage for Census Tract 17031320100 (Test)

Interactive Visualization

A small ipywidgets dashboard was built into the Jupyter notebook to provide an interface to the model. Using this dashboard, the user can select different combinations of input parameters to simulate the hourly demand for each day of the week. The dashboard presents this demand in a heatmap, illustrated below:



Distance: 0.251

Direction: NW

Density: 8000

Mean Fare: 10

Discussion

The final regression model uses bagging to produce reasonably accurate predictions of rideshare usage as a function of distance and time.

Why does it work so well, and what is the model actually doing?

The time measurements are input to the random forest regressor as a range of integers (hour, day of week). At some level in each tree, the random forest slices these integers into ranges to allocate ride demand to categories. The same day and hour are effectively resampled each week resulting in duplicate samples with different values of the dependent variable (i.e. multiple measurements of ride count for a given tract on mondays at noon). As a result, the model is essentially predicting future behavior from an average of past behavior, with some randomization due to the bootstrapping performed by the random forest algorithm. This ultimately works because the time series is stationary and very repeatable. A dataset with an evident trend or change in variance over time would not work as well. Removing outliers (airports and holidays) also helps the model fit the more typical usage patterns.

5. Conclusions and Future Work

The Chicago Trips dataset has been successfully applied to segment the Trips dataset into distinct usage patterns and to make predictions of rideshare usage at different places and times throughout the City. Some additional follow-on work which could be accomplished with the Trips data include:

- Ride Count Measurement: As discussed, the ride counts used for these evaluations were derived from a 5% random sample of records from the full Trips dataset, and thus aren't technically total ride counts. In addition, several months of new data have become available since Trips was first downloaded. Future work could include rebuilding the aggregated data using the full Trips dataset (potentially parallelizing the aggregations with Dask), and integrating the newer data. Incorporating data from spring and summer of 2019 would also enable using month or season as parameters in the model.
- Data Split: The dataset was randomly divided into train vs. test, so test timepoints are distributed throughout the timeline. It would be interested to see how well this method can predict future demand if the train-test-split were performed using a cutoff time (i.e. before = train, after = test) rather than a random sample.
- Demographic Effects: One reason for incorporating demographic data at the census tract level was to enable evaluation of whether rideshare services are consumed equally among all socioeconomic groups, with all other factors held constant. I have demonstrated the ability to integrate income and population data, and could easily bring in other variables (race, age distribution, housing characteristics, etc). Although population density and income were included as parameters in the final regression model, I haven't done additional work to isolate the relative impact of these parameters. This data can also be integrated into the clustering model to provide additional features for segmentation. Alternatively, a simpler regression model could be run to predict overall ride counts per census tract aggregated over the entire dataset or a study period (rather than a time series) to infer the general relationship between demographic factors and rideshare usage.
- City Features: Other variations on these evaluations would include incorporating distances to other city features such as transit stations, cultural features, restaurants and schools to see how proximity to these features affects demand or clusters of behavior.
- Clustering Scale: The clustering model was run at the Community Area level to simplify interpretation of the results. Clustering at finer scale (i.e. at the census tract level) may reveal more detailed usage patterns.
- Clustering Weights: The clustering model used equal weighting for each of the parameters, effectively giving more influence to the fields that summarize ride counts per periods of the day, and less influence to the remaining fields. Additional work would involve adjusting the weighting of these parameters. It would also be interesting to add in some demographic data (income, etc).
- Time Series Modeling By Tract: It would be interesting to forecast utilization for individual census tracts using ARIMA or another time series model.