Chicago Taxi Trip Duration through Leveraging Stochastic Gradient Descent

1) Abstract

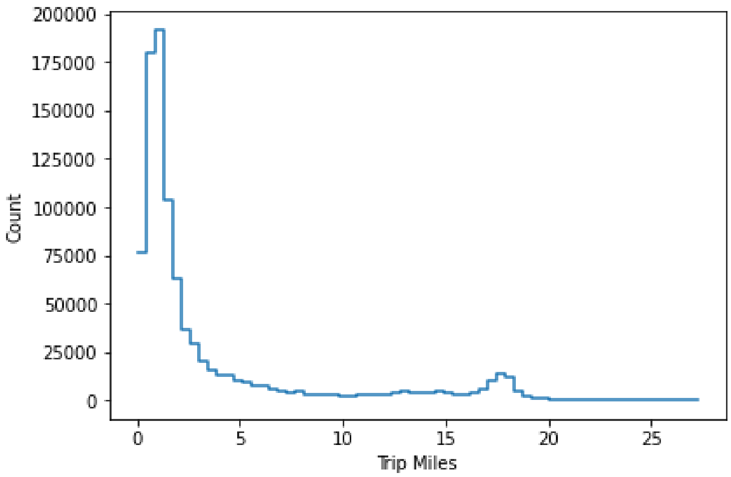
For customer experience and resource management in intelligent transportation systems, such as taxi and ridesharing services, accurate forecasts for optimal route suggestion, trip duration, and total distance are critical.

Predictive analytics are essential for data-driven decision making, exploiting the patterns found in the historical data to identify risks and opportunities. However, these analyses can often be misleading, largely because of the quality and quantity of the data used for prediction. Finding a high-quality dataset and developing a well-defined predictive task, coupled with the selection of appropriate modeling techniques, can yield favorable results when applied to the decision-making process. In this task, predictive modeling is used to forecast taxi trip duration in the city of Chicago, factoring in variables such as the day of week, time of day, departure point, and destination. To accomplish this, a dataset of taxi trip records from 2018 has been used to train a variety of predictive models, to be tested on a dataset composed of the first month of records from 2019. To evaluate model efficiency, the metrics of Mean Squared Error (MSE) and Mean Absolute Error (MAE) have been implemented. Modeling algorithms, such as linear regression, XGBoost and Stochastic Gradient Descent Regressor, were used and the Stochastic Gradient Descent Regressor was chosen as the final model due to its exceptionally low MSE and MAE, in comparison to the alternative models.

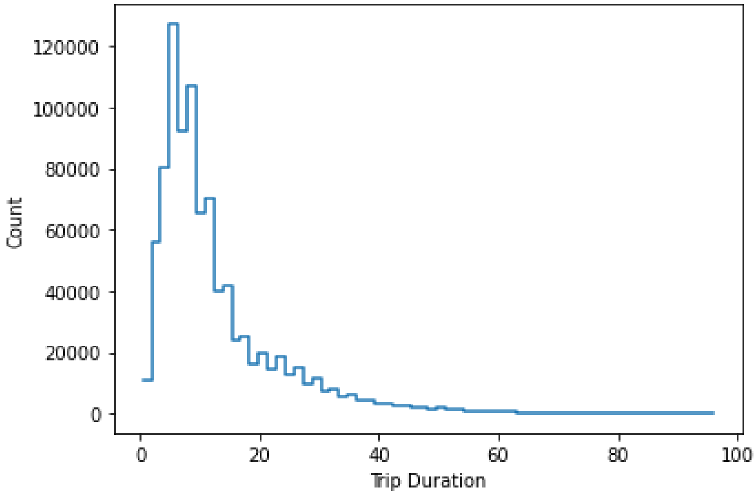
2) Exploratory Data Analysis

For this predictive task, the primary dataset is the Taxi Trips dataset, collected and reported by the City of Chicago, from two major payment processors-- which constitute most taxis in Chicago. The dataset consists of data for 198 million distinct taxi rides. The data collection began in 2013 and has been updated monthly, with the last update being November 10th of 2021. There are 23 fields, 21 of those being numerical and 2 being categorical. A table of the available fields and a short description of each can be referred to in the appendix following the references section of the analysis.

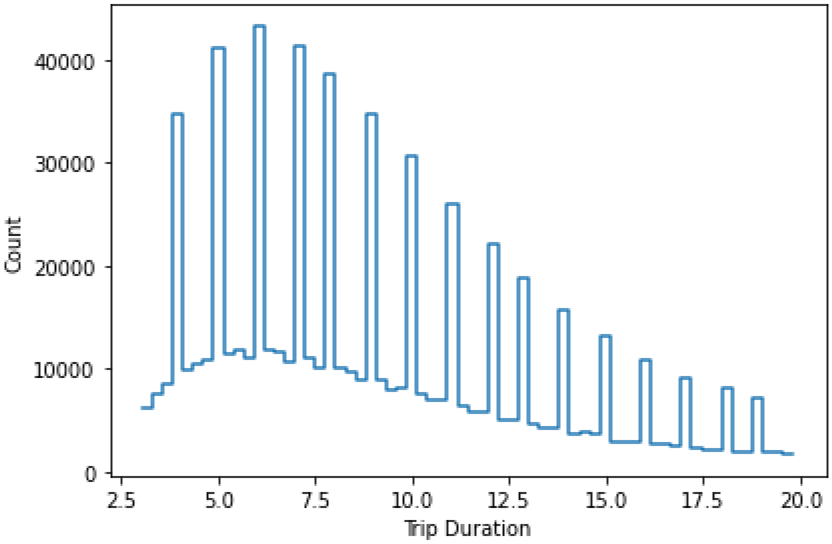
Visualized below, are a series of distributions for two fields, from the year of 2018 and month of January 2019 that were used for the training and testing sets respectively, and fundamental in the model building process. ‘Trip Miles’, defined by the distance of a taxi ride in miles, exhibited in *Fig. 1,* illustrates that the majority of riding miles are within the 0 to 3 mile range, with a maximum count of about 190,000 trips at about 1 mile.



‘Trip Duration’, defined by the time period of the cab ride in minutes, is exhibited by *Fig. 2* below. This distribution illuminates the fact that the majority of trips are within the 3 to 20 minute range, with a maximum count of about 125,000 trips at about 5 miles. Additionally, there are noticeable distinct peaks that will be discussed in the analysis next as they are more distinguished in the following figure.



The removal of outliers was appropriate for this predictive task because of the main interest in trips only within the city limits. *Fig 3* below depicts the ‘Trip Duration’ data reduced to the majority of records following the outlier removal, between the range of 3 to 20 minutes. The distinct peaks mentioned prior are now much more prominent. These peaks at integers can be attributed to the rounding of taxi cab trips to the nearest whole minute, which will be limiting to the true predictive accuracy of the model in a real world application, due to the duplicity of the data points.



Furthermore, the ‘Trip Miles’ and ‘Trip Duration’ features were right skewed and accordingly transformed using Standard Scaler. In addition, the Hour of Day and Location features were also transformed and have been visualized in the Model section of the analysis.

The Vaex library was utilized to handle the extensive amount of data. Through its application the processing time was tremendously reduced, allowing a much larger dataset for training the model.

3) Identify a Predictive Task

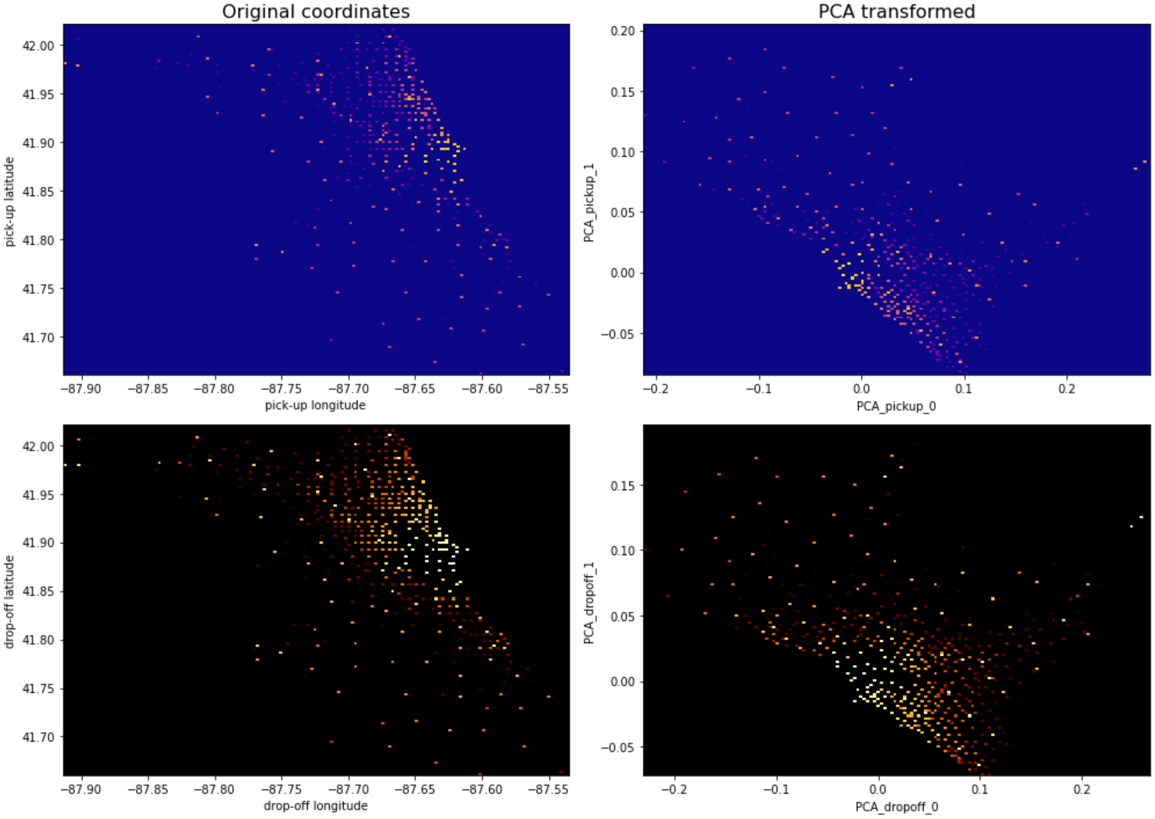
The predictive task of this model is to determine the trip duration of a taxi ride in Chicago given a set of features commonly collected during a taxi trip: pickup location, drop off location, time of the requested ride, and whether it was a weekday or weekend for example.

Using a variety of methods within the vaex library, data cleaning, feature engineering and pre-preprocessing of the data will be done prior to training a few models. To train the models in this task, the data spans twelve months in 2018, while the test set is one month of data from 2019. After checking the distribution, as discussed in the EDA above, the training set was cleaned to not only drop the variables that were not intuitively useful, but to also remove statistical anomalies.

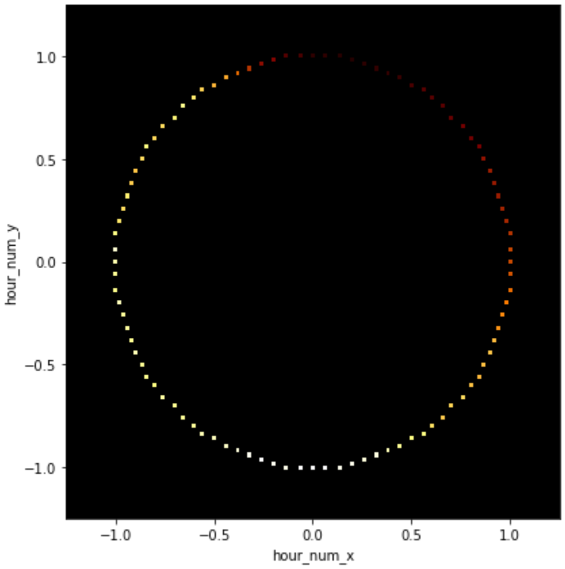
The pickup and drop off information was missing for approximately 2% of the data due to privacy concerns. Rather than imputing this data, the choice was made to not consider it at all. Additionally, the latitude and longitude information was only present for rides that occurred within Chicago. So if either a pickup or a drop off occurred outside of the Chicago district lines, the data was not considered. As seen in the distribution of trip duration above, since the number of trips extending beyond 20 minutes drastically tapers, the data considered was also only for trips that were between 3 and 20 minutes. After removing all of these anomalies and outliers from the data, the final working data, approximately 14 million trips with minimal abnormal samples, would be used to train the model.

After this initial level of data cleaning, the next step was the creation of new features, the first set of which are simply extracted from the pickup timestamp of each ride. The next feature is computationally more demanding: the angle direction of the taxi trip, which will serve as an indication of the direction the taxi is traveling to its destination from its point of origin.

The next step before passing this data through models is to suitably transform the data, in this case, in 2 ways: through a principal component analysis (PCA) transformation and cycle transformation. The first set of features to transform will be the coordinates: pickup and drop off latitudes and longitudes. Since the streets in most cities, including Chicago, form a grid pattern, one way to improve the performance of a model is to apply PCA transformations to these coordinates. The main goal of PCA is to scale large numbers of samples. For this particular task, by applying PCA transformations to the pickup and drop off coordinates, the exponential number of longitudes and latitudes would be optimized by aligning with natural geographical axes, instead of being at odd angles. As seen in *Figure 4,* post PCA transformation, the most densely populated coordinates are centered around *(0,0)* for both latitude and longitude.

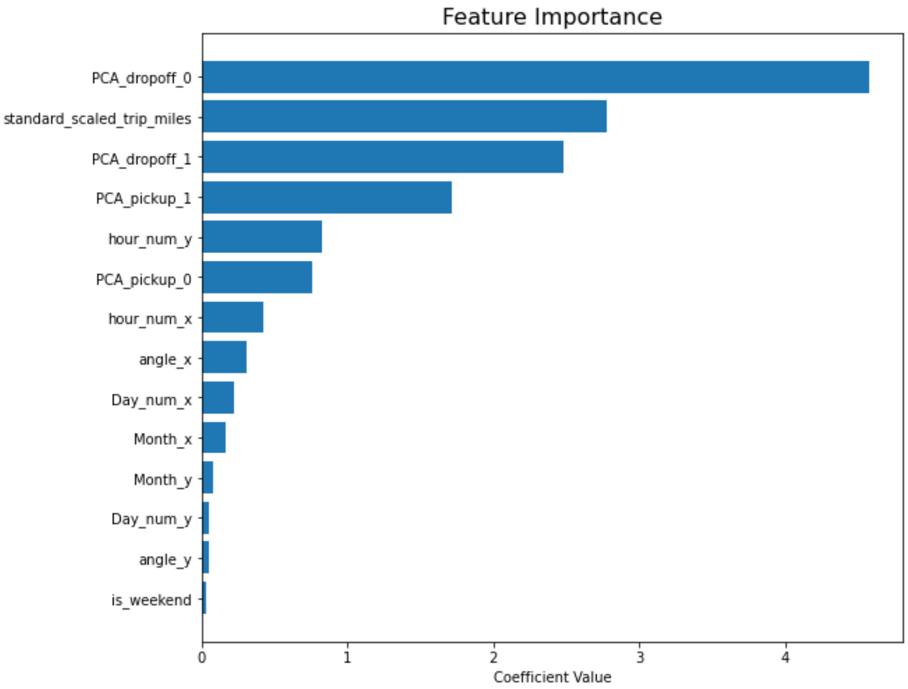


The next set of features is with respect to features defined earlier, which are cyclical in nature. For example, when looking at the days in a week, Monday is as close to Sunday as it is to Tuesday. Thus, using CycleTransformer, each feature is treated as the angle φ in a unit circle in polar coordinates. Taking these coordinates, in a polar coordinate system, the transformation converts them into a Cartesian coordinate system. This results in each feature getting two components *(_x and _y).* This transformation is applied to three time related features and the direction of travel feature created. If this transformation is plotted, it should result in a unit circle, and as seen in *Figure 5,* the *hour* feature plotted results in a perfect unit circle.



The last pre-processing task is to deal with the trip mileage, which as seen in *Figure 1* is a highly skewed continuous feature. This feature will simply be scaled using standard scalar.

After all these pre-processing steps, the final model will be built using 14 variables as seen in *Figure 6* -- 13 features described above and an additional feature which flags whether the trip occurs on a weekend.



For each trip within the training and testing sets, the ‘ground truth’ is known since the actual trip duration is provided within the data. Thus, to assess the validity of the resulting predictions from the models, a comparison between the actual trip durations and predicted trip durations can be done. To quantify the significance of these predictions, the two main metrics used were: MAE & MSE.

4) Modeling

Making adequate choices for modeling algorithms is imperative to the success of a predictive task. In this case, a form of linear regression was the most appropriate for predicting the trip duration with the available predictive features. The models applied to this predictive task were a Scikit-Learn Linear Regression, an XGBoost Classifier with a regression objective, and a Scikit-Learn Stochastic Gradient Descent Regressor.

The Linear Regression was a first attempt at a baseline model because of its easier interpretation, allowing the opportunity to address inconsistencies in the training and testing sets, as well as any barriers to the overall performance of the predictor. However, the MSE and MAE were higher than commonly accepted limits. To improve the efficiency of the estimator, XGBoost Classifier and SGD Regressor were then implemented.

As for the XGBoost Classifier, the XGBoost library offers several appealing features that make it a popular algorithm choice for many predictive modeling projects. XGBoost requires less feature engineering, has attributes that make interpreting feature importances easy, does not require outlier removal, and is less prone to overfitting. The model performed better than Linear Regression yielding considerably lower MSE and MAE. However, XGBoost failed to outperform the SGD Regressor in this predictive task because of the overall difficulty of meticulously tuning the excess of parameters required to build a high-performing model. The resulting MAE and MSE of the Linear Regression and XGBoost Classifier models can be referred to in the appendix below.

Of the three models, the SGD Regressor was the most favorable with a relatively lower Mean Squared Error and Mean Absolute Error. In a stochastic gradient descent algorithm, the loss is estimated each sample at a time and the model is updated along the way with a decreasing learning rate. The Scikit-Learn SGDRegressor is a simple yet effective optimization algorithm that identifies the values of parameters and coefficients of functions that minimize the cost function. It’s application is in discriminative learning of linear classifiers under convex loss functions. The SGDRegressor is a strong predictor when employed to large-scale datasets, such as the Taxi Trip dataset in this case with millions of records, because the update to the coefficients is performed for each training instance, rather than at the end.

Due to the magnitude of the training and testing sets, the SGDRegressor performed the best. To efficiently employ the model we solicited the support of the IncrementalPredictor function in the Vaex library, which wraps a scikit-learn estimator and makes it a vaex pipeline object. The SGDRegressor was set at a ‘constant’ learning rate, with an initial learning rate value of 0.0001. The model was then fed into IncrementalPredictor, along with the features after conducting PCA, the target, a batch size of 11 million, an epoch value of 1, and shuffle set to false. Following the fitting of the pipeline object, the MAE and MSE of the training and testing sets were output. The training set reported a MAE of 2.53 minutes and MSE of 9.95 minutes squared. The testing set reported a MAE of 2.47 minute and MSE of 10.6 minutes squared.

5) Related Literature

Trip duration prediction is a particular field of study that is extensively modeled because it is applicable to many different industries, whether it be public or private transportation, ground shipping, or just the average commuter. Progress in the optimization of calculating trip duration will lead to more accurate estimates of the time it takes to travel along a certain route, which will be significant for moderating the flow of traffic, preventing gridlock in cities, and informing businesses and individuals about the optimal times they should depart to reach their destinations.

Taxi trip data is a robust source for modeling trip prediction because of the vast amount of data points collected, the reliability of the data, and the range of times captured all in a densely populated area, where elements such as traffic, events, and seasonality are all factored into play. Our dataset comes from the City of Chicago Data Portal, maintained by the Department of Business Affairs & Consumer Protection. The recording of data began in 2013 and has been continual through present day, updated monthly. It is a duty of the City of Chicago to collect and report taxi trip data as a regulatory agency and it serves a purpose for quality and safety assurance, as well as aggregate analysis.

Several studies have advocated using taxi trips to estimate journey time between two places. There are two types of works in this category: road segment-based and path-based. The trip travel time was divided into link travel times and intersection delays using a road segment-based technique. Explicitly modeling the time delay at an intersection, on the other hand, is difficult. As a result, some works simply represent a trip as a series of connected road segments and estimate the trip travel time by adding the travel times of each individual road segment. Rahmani et al. [1] use the correlation between different road segments in terms of their historical traffic patterns to calculate travel time and delay at intersections. Yuan et al. [2] propose a road segment-based method variant. They construct a landmark graph based on the trajectories of a large number of taxis, where a node (entitled a landmark) is a road segment frequently traveled by taxis and an edge denotes the aggregation of taxis' commutes between two landmarks. The total of the journey times between landmarks is then used to approximate the path's travel time.

Path-based trip travel time estimation methods use frequent trajectory patterns to estimate the total travel time of a path. It uses the average journey time of a pattern to represent the trip of the path corresponding to the pattern after mining frequent patterns from previous trajectories in advance [3].

To estimate the trip time of a path, a PTTE model-based method is proposed [4]. They first use a context-aware tensor decomposition approach to estimate the trip time of a road segment, and then use a dynamic programming solution to find the most efficient concatenation of trajectories given a query path. They infer the trip time for individual segments, but instead of concatenating them one by one, they combine the time with trajectory patterns to construct a subpath. Some studies used this strategy to treat common trajectory patterns as subpaths and concatenate the subpaths into a target path [5]. The total of the travel times of these subpaths is then used to approximate the travel time of a path. There is no need to model intersection delay in these methods. The query pathways, on the other hand, may not fit into any patterns in the present time period or in the past. These methods must choose more trajectory patterns by employing a tiny support to be able to respond to varied query pathways.

Machine learning techniques are used in a few studies to anticipate trip duration. Blandin et al. estimate arterial travel time using machine learning approaches and convex optimization [6]. The sampled journey time from probe vehicles is considered to be known and used as a training set for a machine learning algorithm to generate a nonlinear estimate of trip time. Through kernel regression, they apply convex optimization to improve the performance of the nonlinear estimate. [7] proposes a dynamic Bayesian network-based technique for estimating journey time on road linkages. The parameters are evaluated using Markov chain Monte Carlo methods, with the trip times on the road links assumed to be independent and log-normally distributed.

The majority of these studies concentrate on estimating travel time on non-trip road linkages or routes. In actuality, passengers may not be aware of the taxi driver's actual path. Furthermore, finding past travels with the exact same path covered for long treks is nearly impossible for a new trip. We propose using a trip-based strategy in our research. We believe that changes in travel duration over time indicate traffic dynamics. As a result, we use the historical trip duration as a training sample and develop the prediction model using Stochastic Gradient Descent Regressor modelling.

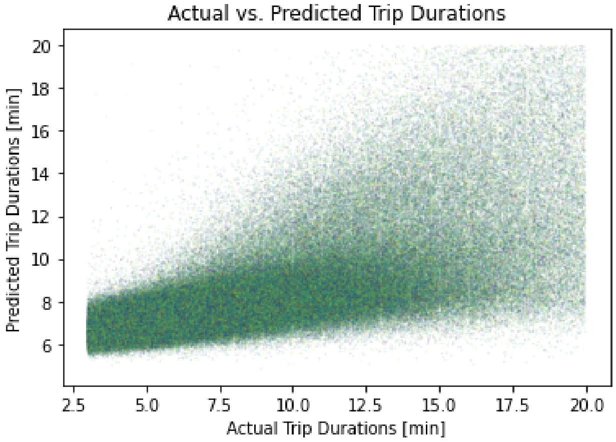
6) Results & Conclusion

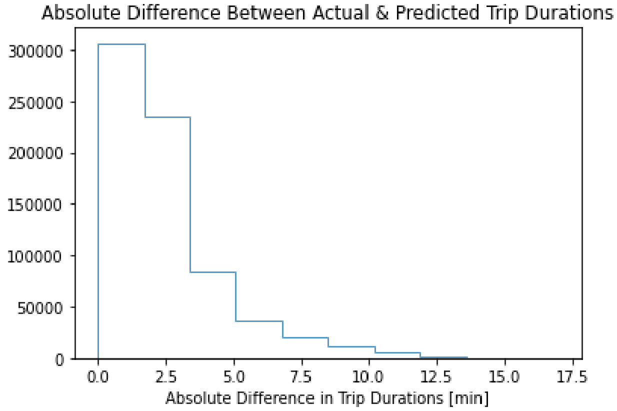
The results of the 3 models used for this predictive task: Linear Regression, XGBoost and SGD Regressor are listed in *Fig. 7*.

Model	**MAE**		**MSE**	
	Train	**Test**	**Train**	**Test**
Linear Regression	5.75 minutes	6.23 minutes	14.86 minutes	15.04 minutes
XGBoost	3.65 minutes	4.53 minutes	12.79 minutes	13.58 minutes
SGDRegressor	2.53 minutes	2.47 minutes	9.95 minutes	10.60 minutes

Exhibited by the table in *Fig. 7,* the SGDRegressor performed remarkably better than the other two models, with the lowest MSE and MAE on both training and testing sets. Since these 2 metrics were chosen to understand the significance of the predictions, these values across the models can be compared. A lower score on both these metrics indicate a more optimal model for this particular dataset and prediction task. And the lowest scores were observed for the SGDRegressor model.

As depicted by the scatterplot in *Fig. 8,* the actual vs. predicted values exhibit a linear relationship, indicating that the model is a highly-accurate predictor for this task. This is because the predicted duration of a trip was significantly similar to the actual duration of the trip. Thus, the SGDRegressor model was trained and able to produce the most accurate predictions, compared to the remaining models. The absolute difference, as plotted in *Fig. 9,* shows a positive trend, with the majority of the predicted trips having a trip duration absolute error less than 5 minutes. Furthermore, nearly 90% of the predictions have an absolute error of less than 5 minutes.





The results indicate that the predictive modeling approach is reliable in predicting the taxi trip duration for a given day, given the time, the departure point, and destination point of the trip. The model's reliability can be further improved through including additional historical data, as the Vaex library facilitates running a predictive model with a dataset as large as a billion records within a few minutes.

References

[1] M. Rahmani, E. Jenelius, and H. N. Koutsopoulos, “Route travel time estimation using low-frequency floating car data,” in Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems: Intelligent Transportation Systems for All Modes (ITSC '13), pp. 2292–2297, The Hague, Netherlands, October 2013.

[2] J. Yuan, Y. Zheng, and X. Xie, “Discovering regions of different functions in a city using human mobility and POIs,” in Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12), pp. 186–194, August 2012.

[3] W. Luo, H. Tan, L. Chen, and L. M. Ni, “Finding time period-based most frequent path in big trajectory data,” in Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '13), pp. 713–724, 2013.

[4] Y. Wang, Y. Zheng, and Y. Xue, “Travel time estimation of a path using sparse trajectories,” Proceedings of the Conference on Knowledge Discovery and Data Mining (SIGKDD '14), pp. 25–34, 2014.

[5] A. Hofleitner and A. Bayen, “Optimal decomposition of travel times measured by probe vehicles using a statistical traffic flow model,” in Proceedings of the 14th IEEE International Intelligent Transportation Systems Conference (ITSC '11), pp. 815–821, October 2011.

[6] S. Blandin, L. El Ghaoui, and A. Bayen, “Kernel regression for travel time estimation via convex optimization,” in Proceedings of the 48th IEEE Conference on Decision and Control held jointly with 28th Chinese Control Conference (CDC/CCC '09), pp. 4360–4365, Shanghai, China, December 2009.

[7] A. Hofleitner, R. Herring, P. Abbeel, and A. Bayen, “Learning the dynamics of arterial traffic from probe data using a dynamic bayesian network,” IEEE Transactions on Intelligent Transportation Systems, vol. 13, no. 4, pp. 1679–1693, 2012.

Appendix

Column Name	**Description**	**Data Type**
Trip ID	A unique identifier for the trip.	Plain Text
Taxi ID	A unique identifier for the taxi.	Plain Text
Trip Start Timestamp	When the trip started, rounded to the nearest 15 minutes	Date & Time
Trip End Timestamp	When the trip ended, rounded to the nearest 15 minutes	Date & Time
Trip Seconds	Time of the trip in seconds.	Number

Trip Miles	Distance of the trip in miles.	Number
Pickup Census Tract	The Census Tract where the trip began.	Plain Text
Dropoff Census Tract	The Census Tract where the trip ended.	Plain Text
Pickup Community Area	The Community Area where the trip began.	Number
Dropoff Community Area	The Community Area where the trip ended.	Number
Fare	The fare for the trip.	Number
Tips	The tip for the trip.	Number
Tolls	The tolls for the trip.	Number
Extras	Extra charges for the trip.	Number
Trip Total	Total cost of the trip, the total of the previous columns.	Number
Payment Type	Type of payment for the trip.	Plain Text
Company	The taxi company.	Plain Text
Pickup Centroid Latitude	The latitude of the center of the pickup	Number
Pickup Centroid Longitude	The longitude of the center of the pickup	Number
Pickup Centroid Location	The location of the center of the pickup	Point
Dropoff Centroid Latitude	The latitude of the center of the dropoff	Number
Dropoff Centroid Longitude	The longitude of the center of the dropoff	Number
Dropoff Centroid Location	The location of the center of the dropoff	Point