Ride Time Analysis of Chicago's Divvy Bikes

Joseph Margolis
Machine Learning Course
Fordham University
New York, NY, United States of America
jlm14@fordham.edu

***Abstract*—The expansion of Chicago's Divvy bike system has been an integral part of Chicago's initiative to not only improve the city's "bike friendliness", but also ease the load on public transportation. This paper aims to guide the growth of Divvy through predictive analyses of ride times. Both regression and multinomial classification models are produced, tested, and hypertuned in an attempt to not only create viable predictors of ride times based on the starting factors of a rental, but also using the feature importance of these models to show how Chicago can best allocate resources in the continuing expansion of Divvy. The top model for regression was a Quadratic Ridge Regression model with an overall RMSE of 18.12 minutes, and for classification the best model was a Support Vector Machine with adjusted thresholds which achieved a weighted F1-Score of 0.6902. While a poor distribution of the data ultimately hindered both the regression and classification models' performances, resulting in models that are not accurate enough for real-time implementation, important information was revealed from the models' feature importance. These include making the lakefront trail more prepared for recreational Divvy use, marketing Divvy memberships to those looking for a new form of daily commute, and ensuring placement of electric bikes in areas where short commuter trips are more common.**

I. Introduction

Year after year, Chicago has been consistently ranked one of the worst biking cities in America across the country. One of the most highly regarded rankings of city "bike friendliness", developed by the Colorado nonprofit organization 'People for Bikes', ranked Chicago the 2,026th best city in the country for biking in 2024, giving Chicago a rating of 9 for "bikeability", while the national average is a 28 [1]. To help combat this issue, along with easing pressures on public transportation, Chicago's rental bike service; Divvy has seen a massive explosion in funding and usage in recent years, including a \$3,000,000 investment from the state of Illinois in 2021. This investment has led to the addition of over 250 Divvy bike docking stations across the city, with usage reaching a record breaking 6,300,000 rides in 2022, 60% more rides than in 2019, and overall biking across the city of Chicago increased 119% in that same time frame which represents the highest growth among the top 10 most populated cities in America [2] [3]. The success has continued, as Divvy's 2022 ridership record has since been surpassed with over 11,000,000 rides reported in 2024, along with the number of Divvy members quadrupling since 2019 [4] [5].

Identify applicable funding agency here. If none, delete this.

With this massive increase in Divvy usage, comes a massive increase in ride data availability, which is extremely important to Chicago's ability to handle the recent expansions, along with continuing to grow the Divvy bike system efficiently. For my project I aimed to use this data to produce a predictor for ride times, with the goals of both producing a model that can accurately predict ride times, along with using the models to assess which factors at the time of rental affect the length of the user's ride. This could play a major role in decision making for future stations including location and what type of bikes (electric or standard) to have at the station. This is also useful for marketing purposes of Divvy's \$143.90 (or just \$5 for low income residents) annual memberships, as they can target the main uses of members [4] [6].

This paper is organized as follows: Section II describes the dataset and cleaning processes prior to modeling. Section III describes the regression modeling process and results, which implements linear regression modeling with various transformations and regularization. Section III describes the classification modeling process and results, which tests a logistic regression model and a support vector machine. Section IV summarizes the findings and acknowledges practical uses.

II. Data Preprocessing

For my analysis, I used data uploaded to Kaggle by Robert Maas, which contains monthly ride data for each month of 2024, containing in total logs for over 4,200,000 rides [7]. The data originally included full scope information of the ride, including starting and ending location and time stamps, along with information about the Divvy Station locations. Since my goal was to create a predictor that could predict ride duration at the time of a user's rental I first dropped all of the ending ride data, focusing only on the location, time of rental, bike type and user type parameters. I also dropped the indentifying information about the start locations, like the station name, as that information would need too much one-hot encoding, likely leading to overfitting and feature explosion when building the model, opting to use the latitude and longitude of the stations for location instead. This left me with features describing the start longitude, start latitude, user membership status, bike type, time started, and month of rental, with trip duration (minutes) representing my target feature.

A. Input Feature Transformation

For both optimization and contextual reasons, many of these features also underwent transformations. For optimization reasons, I normalized the longitude and latitude parameters using a standard scaler. In addition, I replaced the starting time stamp with two features that were derived from it. The first is a flag variable for whether or not it is the weekend, as weekend vs. weekday commuter behavior varies greatly. I also grouped the time stamp data into the following groupings to align more with commuter behavior: 6-9AM, 10am-3pm, 4-7pm, 8-11pm, 12-5am. Lastly, to reduce the feature space from one-hot encoding, I banded the month variable into seasons. For the bike type parameter, I also noticed that only 1% of the data represented electric scooters. Since this value was so minuscule, and to retain the relevance of my analysis to biking in Chicago, I removed all logs that represented electric scooter rides, leaving only those on standard and electric bicycles. Finally, all of the categorical features underwent one-hot encoding as they were not ordinal categorical parameters. This overall resulted in 7 input features before one-hot encoding, and 17 input parameters after encoding and prior to any kernel transformations (regression analysis) were applied.

B. Target Feature Transformation

For my regression analysis, the target feature used was Trip Duration in Minutes. However, when analyzing this parameter, a major area of concern was uncovered. As seen in the first plot of "Fig. 1", this parameter is not at all on a normal scale, with the vast majority of the data showing rides of less than 25 minutes, and the remaining data skewed very right with extremely high outliers. In order to handle some of the outliers which are likely due to stolen bikes or incorrect inputs while still retaining representation in the dataset of important rides representative of recreational use, I first removed all rides over four hours and under 3 minutes from the dataset, which was applied to both regression and classification models. Then, in order to transform the data closer to a normal distribution for regression I applied a log transformation to the trip duration variable, as seen in the second plot of "Fig. 1". This still did not fully fix the stray from log distribution of the target feature, which ultimately effected my regression results.

For my classification analysis, I used the same parameter, Trip Duration in Minutes, and grouped it into more contextually relevant groups to describe the practical use of the ride. For this I used three groups: Commuter (rides 20 minutes or less), Extended Commuter (rides greater than 20 minutes and less than or equal to 45 minutes) and Recreational (rides more than 45 minutes). This not only allowed for classification analysis, but also added context to the expected use of the ride from the biker based on the ride time. One area of concern with these groupings lies in the class imbalance, which is displayed in "Fig. 2". Overall, around 75% of the data falls in the commuter group, 20% is in the extended commuter group, and just 5% of the data represents rides. In my analysis this issue was addressed by applying class weights to my models, but this was still a notable issue in my model performance. In the model analysis, label encoding was applied to Trip Category, applying 0,1,2 values to the commuter, extended commuter, and recreational groups respectively.

III. Regression Analysis

A. Model Building

Once the data was fully prepared, I began building regression models for predicting Trip Duration based on my input features. Overall I produced six models. I first applied two separate kernel transformations for my models, the first was a basic linear kernel, which did not change the data, and the second was a quadratic kernel. Other more complex kernels were attempted including an RBF and a polynomial degree three kernel, but it was found I would have to sacrifice too much of my data in order to produce models with these kernels efficient enough to run without crashing my python kernel. On top of each of these two kernels, I also used three methods of regularization, linear regression, ridge regression, and lasso regression to test different methods of limiting the feature space.

To evaluate the best performing model I used a stratified train-test split of my data in order to preserve the trip duration distributions within the train and test data. For efficiency reasons, as these models take over 10 minutes to run altogether, I used a 50/50 split of the train and test data, allowing the models to all only have to train on half of the data, and allowing for extensive testing after. Each model was trained with cross validation, using a three fold validation once again for optimization reasons, and each was tested on their Cross Validation Mean Squared Error. Once the results for all of the models were extracted, One Standard Error rule was applied to find the simplest model that still performs at least within one standard deviation of the highest performing model. After the best model was chosen, it was tested on the test set for RMSE performance and feature importance was extracted.

B. Results

After running all the models on the training set, the model performance of each was calculated and the results are displayed in "Tab. I". From this table we can see that the ridge and linear regression models overall displayed better Cross Validation Mean Squared Errors than both of the lasso regression models. This shows that using lasso to set feature coefficients all the way to zero was very harmful to the model's performance, displaying that most of the parameters proved important in improving the predicting power of my models. The results also show that the two highest performing models in terms of CV MSE were the quadratic kernel models, which both outperformed both linear kernels with a greater CV MSE of about 0.02. After applying the One Standard Error rule to the cross validation results, the top performing model and simplest model remained the overall highest performing model which was the quadratic kernel with ridge regression, which showed an average squared error in validation of about 0.551.

The quadratic ridge regression model was then evaluated on the test set for its RMSE value which is shown in "Tab. II",

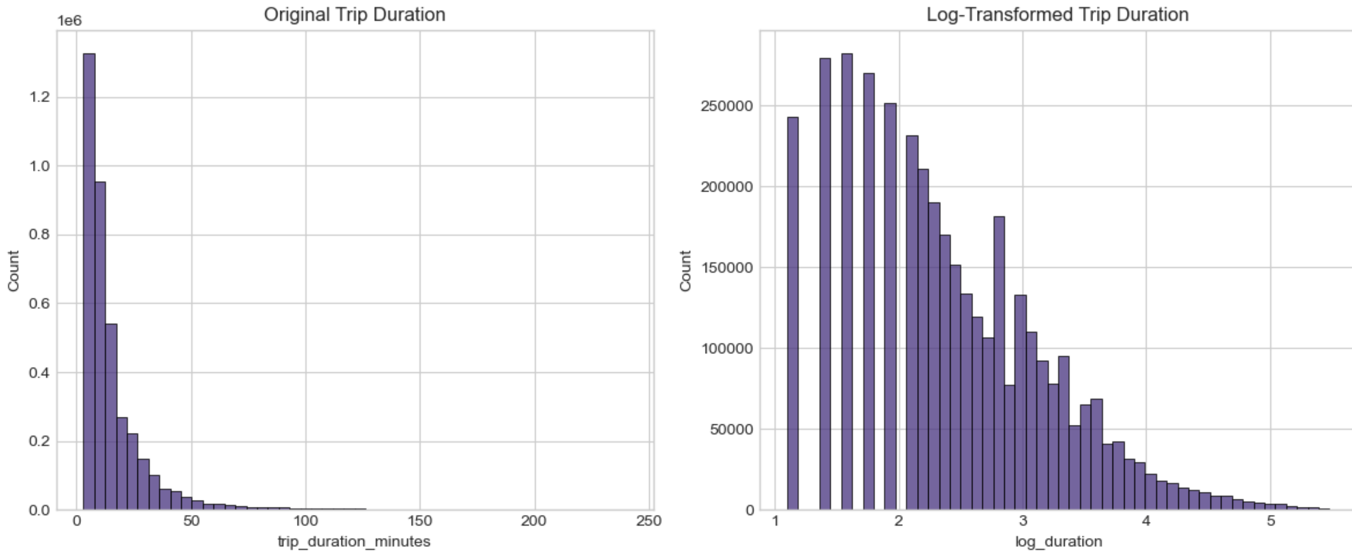

Fig. 1. Distribution of Trip Duration (Minutes) in original vs. log scale

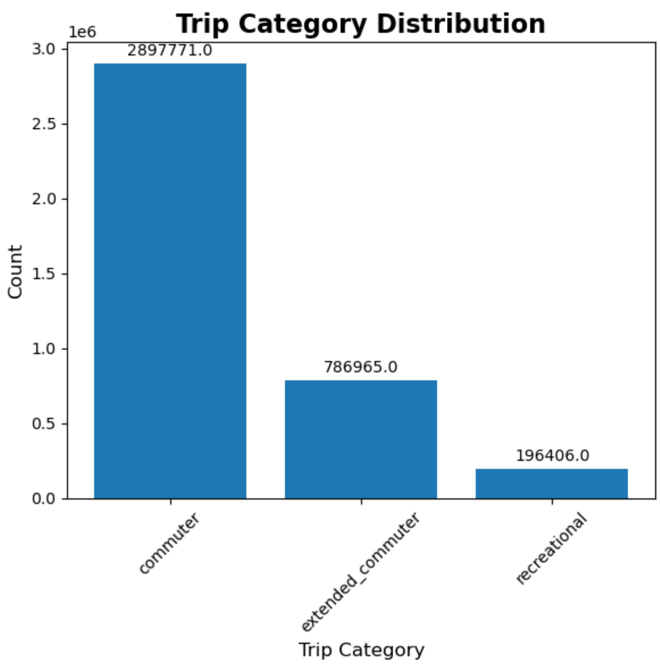

Fig. 2. Distribution of rides across Trip Category classes

TABLE I
MODEL PERFORMANCE COMPARISON

Model	CV_MSE	CV_Std	Test_MSE	Complexity
Poly2+Ridge	0.551330	0.000910	0.550856	3.2
Poly2	0.551596	0.001145	0.550929	4.0
Ridge	0.575812	0.000784	0.575091	0.8
Linear	0.575816	0.000782	0.575094	1.0
Poly2+Lasso	0.601167	0.000785	0.600753	2.4
Lasso	0.648763	0.000591	0.648474	0.6

Results are sorted by CV_MSE performance.

TABLE II
PERFORMANCE METRICS FOR POLY2+RIDGE MODEL

Metric	Value (minutes)
Test RMSE	18.12
Lower Quartile (25th percentile) loss	3.56
Median (50th percentile) loss	4.64
Upper Quartile (75th percentile) loss	5.73
75% of predictions within	9.34

resulting in an average error of 18.12 minutes per ride. In terms of the overall, 237 minute range of data, this does not seem to bad, but when acknowledging that around 75% of the data represents rides less than 20 minutes, this is not performing well. After this I extracted quartile loss values as shown in "Tab. II", to see how the model was performing outside if the outliers, and these values showed much better performance. The median loss value of 4.64 minutes, means that of all the test errors calculated in the test set predictions, the median value was under five minutes, showing that the overall RMSE error of 18 minutes is skewed heavily by extremely large errors made on predictions for truly high ride times. This shows that my model is good at predicting ride length for the commuter length rides, but performs poorly in its predictions for a rider intending to go for a longer recreational ride.

From my quadratic ridge regression model, I also extracted the coefficients of each feature from the quadratic ridge regression model. In total, there are 171 features in this model, 20 of which had coefficients driven to be nearly zero by ridge regression. "Fig. 3" shows the ten most important features, with green bars indicating a positive correlation of the feature on trip length, and red indicating a negative correlation. Of the top ten most important features, seven of them include some sort of interacting with our location parameters, starting latitude and longitude. This shows a major importance of the

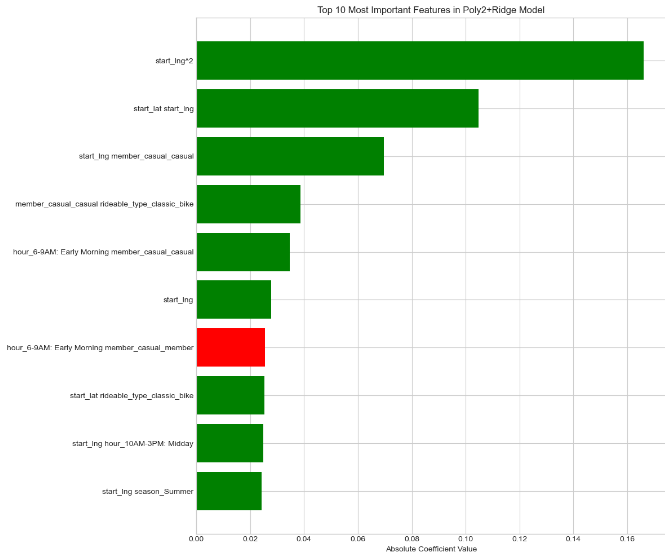

Fig. 3. Feature Importance of Quadratic Ridge Regression Model

starting location of a ride in predicting ride times. The most important parameter, starting longitude squared, shows that as one moves away from the center of the city, their ride length increases heavily, which is not surprising as ones further away from the central, downtown area of Chicago are less likely to use Divvy bikes for shorter work commutes, leading to a higher likelihood of use for longer recreational rides. It is also important to note the positive correlation of longitude with ride time, as an increase in longitude represents movement east in the city towards Chicago's lakefront trail which is a hot-spot for recreational biking and an area of great importance in Chicago's Divvy bike expansion.

Membership status also showed much importance in the quadratic ridge regression model, as it played a factor in four of the ten most important features. "Fig. 3" shows that in the three features where casual riders were represented, there was a positive correlation with ride time, while the one which represented Divvy member rentals showed a negative correlation. This points towards the reason for purchasing a Divvy membership as a means for shorter, more regular commutes as opposed to use for longer recreational rides.

IV. Classification Analysis

A. Model Building

After regression, I completed a classification analysis on the target feature Trip Category, which I derived from the Trip Duration Feature. Since the categories I produced from this feature were ordinal, increasing in value across the groups commuter, extended commuter, and recreational, I ran a multinomial multiclass classification analysis. For this, I started with testing two models, logistic regression and a support vector machine. Since none of the three classes are more important contextually to predict properly as may be the case in a health related problem, I selected the better of the two models based on weighted F1 scores. This method balances the precision and recalls of each of the three groups, but does so in a way that weights the categories to optimize for the real world distribution of the data, i giving more importance to the commuter class which in the real world shows up more often than the extended commuter or recreational classes, providing more importance to my overall performance of the models as opposed to the performance on a class by class basis.

Since Support Vector Machines do not have a true multinomial approach, I used the Crammer and Singer method to provide similar results to my Support Vector Machine, which uses decision scores and partitions the output space into three regions. This methos still allows simultaneous predictions across the three classes like multinomial classification in order to minimize hinge loss as opposed to using one v. rest analysis. For optimization reasons, as this code also takes a long time to run, I made my base assumption for the better if the two model pipelines based on the outputs of the models with standard parameters, with C = 1 for the logistic model and C = 0.5 for the support vector machine. It is also important to note that applying kernel transformations to my models was attempted, but for my classification analysis, any kernel transformation would have required sacrificing too much of my data to be able to run without crashing my python kernel. For this reason, I ultimately stuck with linear kernels for both of my models.

To analyze each I once again used a stratified, train-test split of 50/50 for optimization reasons, and extracted the balanced accuracy of each of my two models to select which one was better. To handle the issue of class imbalance, I calculated weights for each class based on their distribution and applied these weights to my model pipelines. The resulting class weights applied were about 0.44 for the commuter class, 1.64 for the extended commuter class, and 6.70 for the recreational class. After choosing the best model based on their original parameters, I then hyper-tuned the best model to try to improve performance. For this used a grid search cross validation with three folds for optimization, and still using balanced accuracy to determine the best overall model. The two parameters I hyper tuned were the C value, which balances overfitting by decreasing its prioritization of training balanced accuracy as it decreases, and class thresholds, which balances how certain of each class my predictor should be to make a prediction towards that class. Once final model was selected, feature importance was extracted to better understand how each of my features affects the likelihood of my model making a prediction for that class.

B. Results

The results in "Tab. III" show that, based on Weighted F1-scores, the best performing model of the two was my Crammer and Singer Support Vector Machine, outperforming the logistic regression model by about 3.38%. One are of note is how the models handles each class in particular. Both models achieved identical F1 scores for the Recreational class predictions, while the SVM model heavily outperformed logistic regression for the Commuter class, with a F1-score 10% higher. It is also important to note the SVM models

TABLE III
Model Performance Comparison

Metric	Multinomial LR	C&S SVM	Δ
Per-Class F1 Scores			
Commuter	0.69	0.79	+0.10
Extended Commuter	0.22	0.03	-0.19
Recreational	0.22	0.22	0.00
Overall Performance			
Accuracy	0.52	0.61	+0.09
Weighted F1	**0.5741**	**0.6079**	**+0.0338**
Average Metrics			
Weighted Precision	0.68	0.69	+0.01
Weighted Recall	0.52	0.61	+0.09

Note: Δ represents the difference (SVM - LR). Bold indicates the primary optimization metric.

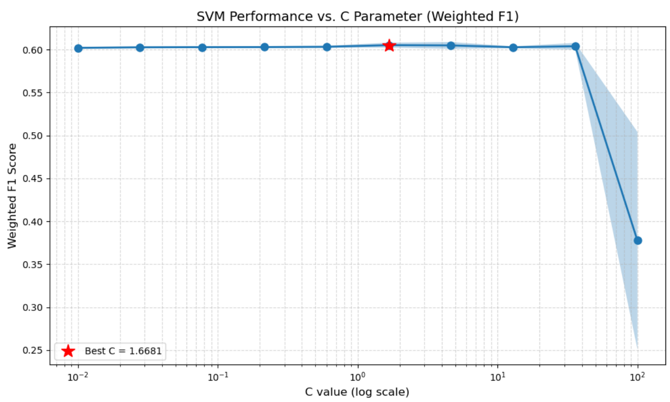

Fig. 4. CV F1-Scores for each C Value

extremely poor performance on the Extended Commuter class, with an F1-score of just 0.03. Although the F1-score for logistic regression was much better at 0.22, it still overall is not very high performing. The data overall shows the great affect the class imbalance had on model performance across all three classes, as even implementing class balancing in my pipeline was not enough to overcome this data issue. Ultimately, since the data shows a real world context with much more use of Divvy bikes for commuter purposes, I did not want the other classes poor performance to take away too much from my interpretation of the best model, so the weighted F1 scores led me to choose the Support Vector Machine as the better option of the two models, with a weighted F1-score of 0.6079, compared to logistic regressions weighted F1 of 0.5741.

After choosing to continue with the SVM model, I tried hyper-tuning parameters for the SVM model, starting with the C parameter which is used to balance overfitting, where a higher value of C results in a smaller margin and is more prone to overfitting to the training data. For this process, I used a Grid Search Cross Validation method, applying three fold cross validation to 10 C values across the log space ranging from 0.01 to 100. The results shown in "Fig. 4" show that across the values tested for C, there was not much improvement in the resulting CV Weighted F1-Scores, as all of them besides C = 100 hovered around CV F1-Scores of 0.6. The overall best performing C value was C = 1.6681, which was then ran

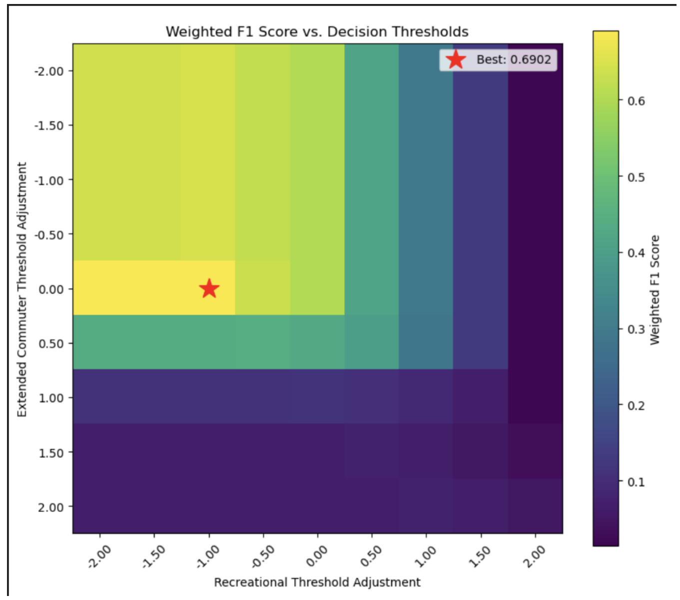

Fig. 5. F1-Score Distribution Across Threshold Tuning

for predictions on the test to attain a very slightly better test set Weighted F1 score of 0.61. This new C value parameter was attributed to the model for threshold tuning.

For my hypertuning of thresholds for the minority classes, ran another grid search cv for adjusted thresholds of both of the minority classes, extended commuter and recreational. For each of these classes, I tested ten threshold adjustments across the linear space ranging from -2 to 2, where negative values result in a lower likelihood that the model will predict that class and positive values resulting in a higher likelihood. From my results shown in the heat map in "Fig. 5", we can see that this hypertuning process was much more successful in improving my SVM model than the C value tuning. I found that the best threshold values to use were a 0 for extended commuter, representing no change, and -1 for the recreational class, which makes it harder for the model to predict recreational by subtracting 1 from its decision score for each prediction. When ran on the test set, the results displayed in "Tab. IV" show a slight decrease in model performance in predicting the Recreational class, which is the class with the least distribution, but improved performance in the Commuter and Extended Commuter classes, particularly with a major improvement in Extended Commuter Predictions, with an F1-Score increase of 0.26. Overall, applying the threshold tuning improved my Weighted F1-Score up to 0.6902, an increase of 0.0823. Still, it is important to note the affect class imbalance has on my model's performance, as the performance of this model is still heavily carried by the high performance on the Commuter Class (F1 = 0.83), and brought down by the still poor performance in the minority classes.

To finalize my classification analysis, I extracted feature importance from my SVM model, particularly focusing on how each feature correlates to each specific class' likelihood of being predicted by the model. This involved extracting the

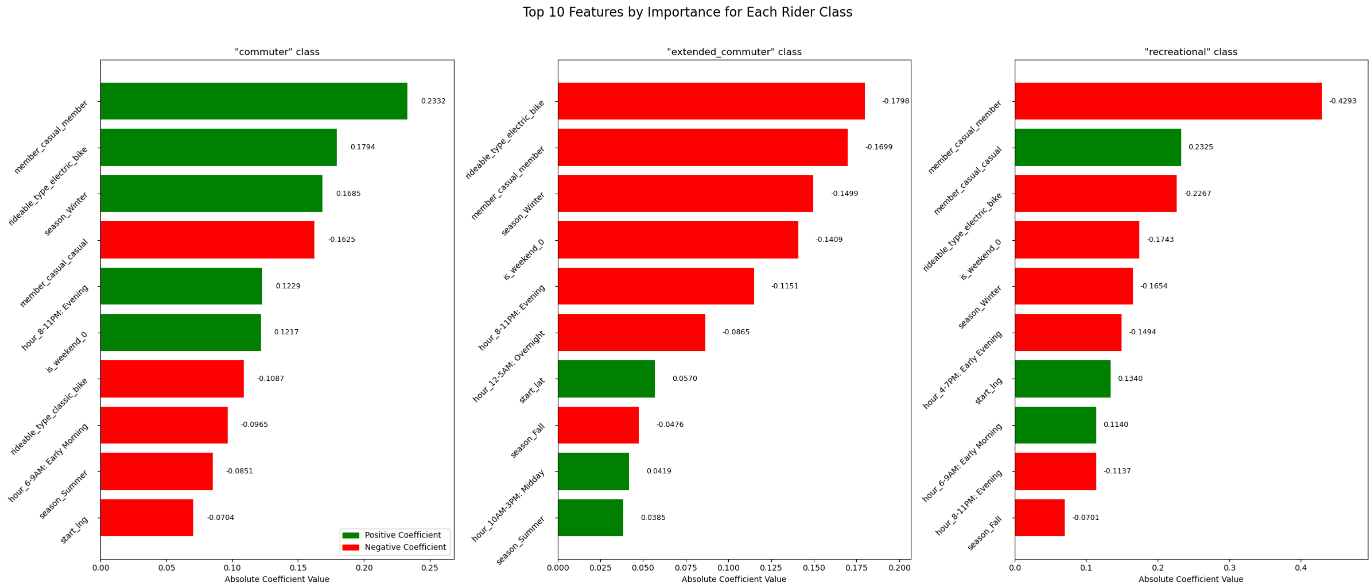

Fig. 6. Feature Importance for each class of Support Vector Machine

TABLE IV
SVM PERFORMANCE COMPARISON: BASE VS. THRESHOLD-OPTIMIZED

Metric	**Base SVM**	**Optimized SVM**	**Δ**
Per-Class F1 Scores			
Commuter	0.79	0.83	+0.04
Extended Commuter	0.03	0.29	+0.26
Recreational	0.22	0.16	-0.06
Overall Performance			
Accuracy	0.61	0.71	+0.10
Weighted F1	**0.6079**	**0.6902**	**+0.0823**
Average Metrics			
Weighted Precision	0.69	0.68	-0.01
Weighted Recall	0.61	0.71	+0.10

Note: Δ represents the difference (Optimized - Base). Bold indicates the primary optimization metric. Thresholds: Extended = 0.0, Recreational = -1.0

coefficients of each feature from the decision score calculation formula for each of the classes. The side-by-side results of the top ten most important features are displayed in “Fig. 5”, once again where the green bars show a positive correlation with that class’ decision score while the red show a negative. Compared to our quadratic ridge regression model feature importance, there is a noticeable decrease in the importance of location in selecting classes, although we can still see that the negative correlation of starting longitude on the Commuter class and positive correlation on the recreational class points towards the previous conclusion that the lakefront trail on the east end of the city is a major area for recreational biking.

The SVM feature importance highlights the importance of two other parameters, membership status, which was also noticed in the quadratic ridge regression model, and bike type. Those who are paying for the annual Divvy membership had a highly positive impact on the SVM decision score of the Commuter class, as being a member increased the decision score by 0.2332, with a heavily negative correlation for the Extended Commuter and Recreational classes (-0.1699 and -0.4293 respectively), once again showing intentions behind purchasing a Divvy membership relating to need for Divvy bikes for more regular, short commutes. For the Bike Type feature, which has two classes, standard bike and electric bike, the results show that using electric bikes had a positive impact on Commuter decision scores (0.1794 increase with electric bike rental) and negative on both of the minority classes. This also points to a similar intention for a user deciding to rent an electric bike as opposed to a standard bike, being for more typical shorter commutes as opposed to a longer recreational ride.

V. CONCLUSION

In conclusion, over all the models tested in this project, none are truly accurate or efficient enough in my opinion for implementation of real time predictions of ride duration or ride type. This was due to the dispersion of the Trip Duration feature being too heavily focused around much shorter ride times, with a major skew of sporadic longer rides. While I produced a quadratic ridge regression model that achieved a median loss time of less than five minutes off of the grand truth, its overall RMSE showing an average error of about 18 minutes shows a major blind spot in the model, with a poor ability to predict ride times for longer recreational rides. Future work would implement regression models that are more robust when dealing with outliers, including Huber and RANSAC regression.

If any model is implementable, it would be the Support Vector Machine classification model with optimized thresholds, as that model was able to achieve a weighted F1 score of nearly 0.7. This is an improvement from a proportional random guessing model, which would randomly predict about

75% of the rides Commuter, 20% Extended Commuter, and 5% Recreational, and would likely have a Weighted F1-Score around 0.6-0.65. Still, the issue with class imbalance results in overall, model that did not perform very well on the minority classes and would likely need improvement in order to be implemented. This improvement would include fine tuning of class imbalance handling, particularly trying variations of SMOTE and random over and under-sampling instead of using class weights to handle imbalance. I would also fine tune how I categorize different rides.

Another area I constrained myself in my model building, was my intention of retaining as much of the data as possible for my analysis. While it is important to retain the data in effort to ensure that all rides are represented, the extremely large dataset used was not efficient for model building and prevented me from applying more complex transformations to both my regression and classification analyses. Another area for future testing would be to try stratified subsets of the data with more complex kernels in an effort to improve model performance.

Although the models may not be ready for implementation, important information can be derived from my models' feature importance rankings. The first standout factor was the importance of location in my regression model on Trip Duration. One finding was the increase in Trip Duration as one moves away from the center of city westward or eastward (start_lng$\hat{2}$). This is important, because the main use for Divvy rides according to the data distribution is for shorter commuter rides. This means that the further distance between pickup and drop-off stations likely means that it is less likely that one would actually use a Divvy bike, acknowledging that it would mean a longer ride time to drop off. With Divvy bike stations being more sporadic the further away from the center of the city one goes, it shows an area of improvement for Divvy expansion may involve the addition of more stations on the outskirts of the city to promote shorter commuter trips between the residential neighborhoods as opposed to just the central office focused downtown area. Another location factor that stood out was the positive affect longitude had on ride time, as an increase in longitude means a movement east towards Chicago's lakefront trail, the city's hot-spot for recreational biking. Acknowledging the use of Divvy's for recreational purposes at the lakefront trail is important, as Divvy can combine that information with the negative impact that electric bikes showed for recreational ride decision trees from the SVM to properly station their electric bikes away from the lakefront trail, creating more access to standard bikes and setting the electric bikes in areas more useful for shorter commutes (ie. Downtown). Finally, the SVM model showed major importance of Membership Status, with members having a highly positive impact on Commuter Class decision score, and negative impacts on the longer ride classes, a pattern that was also supported by the quadratic regression model. This is a very important distinction for marketing reasons, as it shows that Divvy memberships should be marketed more as an alternative to the CTA (Chicago's train and bus services) for those who have regular commutes. This could be particularly useful for the areas of the cities that do not have close proximity to train stations, and could be combined with the feature importance of Bike Type to place more electric bikes in those areas to promote the use of Divvy's for commuter travel.

References

[1] "Chicago Illinois City Rating Page: Peopleforbikes 2024 City Ratings." PeopleForBikes, 2024, cityratings.peopleforbikes.org/cities/chicago-il.

[2] "Measuring Chicago's Boost in Biking." Replica, 3 Apr. 2024, www.replicahq.com/post/measuring-chicagos-boost-in-biking.

[3] "Divvy for the Entire City: Divvy Service Hits All 50 Wards." City of Chicago:: Divvy for the Entire City: Divvy Service Hits All 50 Wards, 2 May 2025, www.chicago.gov/city/en/depts/cdot/provdrs/bike/news/2023/april/divvy-for-the-entire-city–divvy-service-hits-all-50-wards.html.

[4] "Chicago Sees Record 11+ Million Shared Bike and Scooter Trips in 2024." City of Chicago:: Chicago Sees Record 11+ Million Shared Bike and Scooter Trips In 2024, 8 Jan. 2025, www.chicago.gov/city/en/depts/cdot/provdrs/bike/news/2025/january/chicago-sees-record-10–million-shared-bike-and-scooter-trips-in.html: :text=Divvy

[5] Spielman, Fran. "Record Number of Divvy Rides, More Bike Lanes Made 2023 an 'incredible' Year for Cycling in Chicago, City Official Says." Times, Chicago Sun-Times, 11 Jan. 2024, chicago.suntimes.com/2024/1/10/24033443/bicycles-cycling-chicago-protected-bike-lanes-streets-divvy.

[6] Mulima, Monique. "Divvy Fares Are Going up in February." Block Club Chicago, 8 Jan. 2024, blockclubchicago.org/2024/01/08/divvy-fares-are-going-up-in-february/.

[7] Maas, Robert. "Divvy Ride Data for 2024 (Chicago, IL)." Kaggle, 5 Feb. 2025, www.kaggle.com/datasets/bobbymaas/divvy2024-tripdata?select=combined_tripdata.csv.