Traffic Volume Prediction Using Data Mining Regression Techniques

Brendan George Lauterborn
Department of Computer Science
Towson University
Towson, United States
blaute3@students.towson.edu

***Abstract*—Accurate prediction of traffic volume is essential for infrastructure development and planning. This paper aims to predict the volume of traffic on Interstate 94 westbound between Minneapolis and St Paul, MN. Multiple regression techniques were used to determine the best model.**

I. Introduction

Correctly predicting and analyzing traffic volume on an interstate allows for better planning around road infrastructure. It can also enable accurate updates and delays on current travel time. This paper shows research on predicting traffic volume using the Metro Interstate Traffic Volume dataset published on the UCI machine learning repository. It also investigates what attributes contribute most towards traffic flow.

The study begins by first performing exploratory data analysis (EDA). The goal of this step is to get an understanding of the dataset itself. EDA also allows relationships between attributes to be visualized. The next step is to perform Data Preprocessing. This includes normalization of numerical variables, binning of attributes, and performing transformations of numerical attributes to attempt to achieve normality. From there, multiple regression techniques are applied and evaluated. These techniques include linear regression with ordinary least squares, decision tree regression, random forest regression, and gradient boosting regression. The performance of the model is tested using the coefficient of determination (R^2) and Root Mean Squared Error (RMSE).

The results show that Random Forest Regression and Gradient Boosting Regression perform the best compared to the other models. This highlights the benefits of non-linear models for predicting traffic volume.

II. Dataset Description

The dataset chosen focuses on Metro Interstate Traffic Volume in Minnesota. The dataset can be obtained from the UCI Machine Learning Repository. Hourly Interstate 94 Westbound traffic volume is logged. The size of the dataset is 396 KB and contains 48024 tuples of data. There are 8 attributes that make up the data.

The first attribute is named $holiday$ and represents if the current day is a US National holiday or a regional holiday in MN. This attribute is categorical. The next attribute is $temp$ and represents the average temperature at the current hour in Kelvin. The next two attributes are $rain\_1h$ and $snow\_1h$. They represent the amount of rain (mm) and snow that occurred in the current hour respectively. After that, the next attribute is $clouds\_all$. This attribute represents the percentage of clouds that cover the sky at the current hour. From there we have $weather\_main$ which gives a short description of the current weather, while $weather\_description$ gives a longer description of the current weather. The next attribute is $date\_time$ and represents the hour in which the data were collected in local CST time. The time is in format `YYYY-MM-DD HH:MM:SS`. Finally, the last attribute is $traffic\_volume$. This represents hourly I-94 westbound traffic volume.

III. Exploratory Data Analysis

During the EDA portion, we will begin to understand the relationships of the variables by graphing and analyzing scatter plots, box plots and histograms.

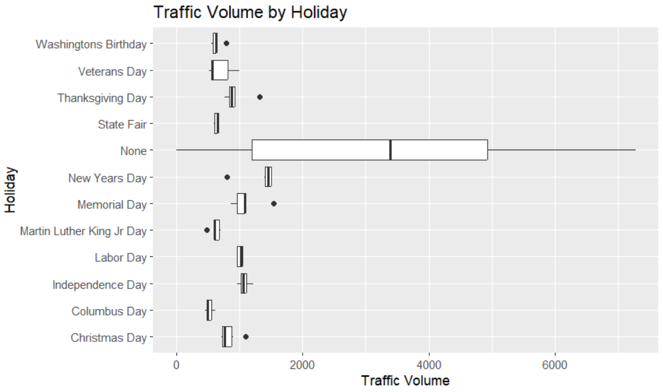


Fig. 1: Traffic Volume by Holiday

We first check to see if the holiday has any effect on the volume of traffic. From Fig. 1 and Fig. 2, it is clear that there is more traffic volume on days that are not considered holidays. Next, we need to check if weather has an effect on the traffic.

From Fig. 3, we can see that weather does not seem to play much of a role on the amount of traffic volume. The IQRs are relatively similar from this box plot. Fig. 4 shows the largest relationship thus far. From Fig. 4 we observe that the time of day has a large impact on the volume of traffic. The early hours of the day tend to have less traffic. There is an early morning

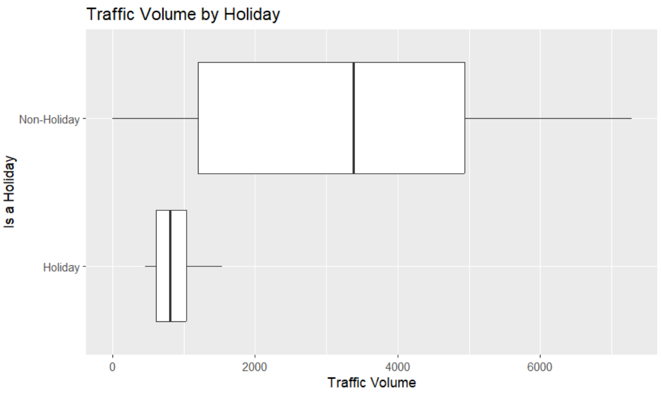


Fig. 2: Traffic Volume by Holiday

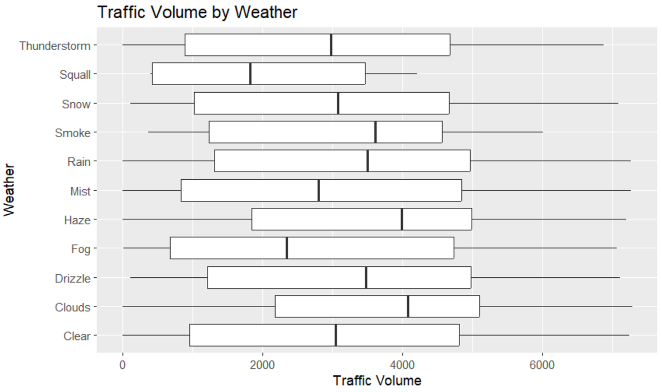


Fig. 3: Traffic Volume by Weather

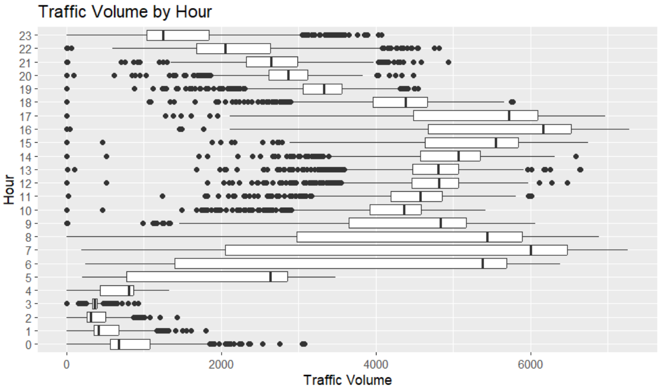


Fig. 4: Traffic Volume by Hour

peak in traffic volume from around 5 hours to 9 hours. The traffic slows down a little until rush hour. There is another peak in traffic volume from about 15 to 18 hours in the day. Fig. 5 shows the relationship between the weekday and the traffic volume. Fig. 5 shows that there tends to be less traffic on Saturday and Sunday. This is somewhat expected as many people are off during the weekends.

From Fig. 6, we observe that the temperature plays little to no effect on the traffic volume. There are a few extreme temperatures that seem unrealistic. There is no relationship that can be drawn from this plot.

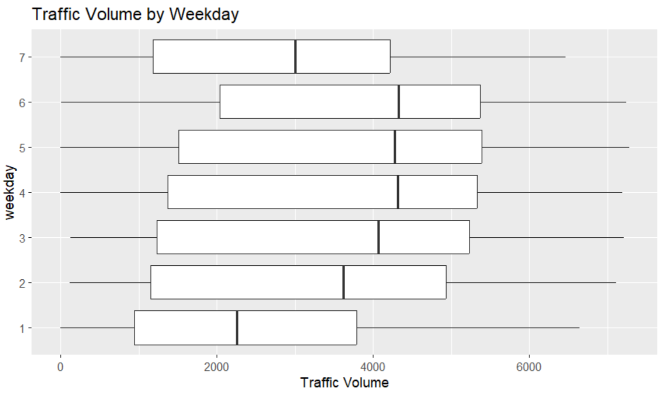


Fig. 5: Traffic Volume by Weekday

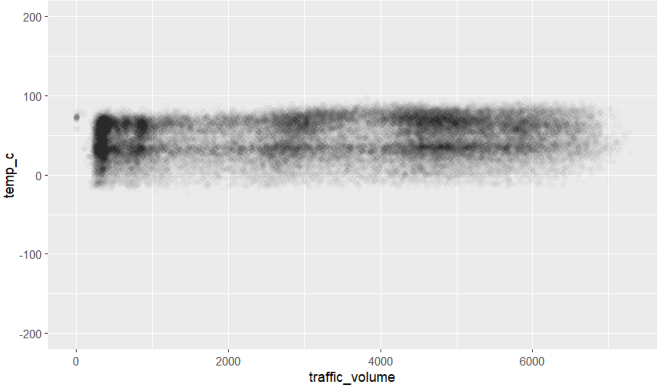


Fig. 6: Traffic Volume vs Temperature(C)

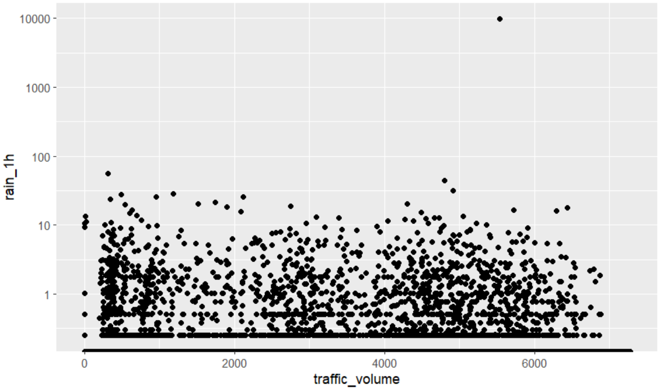


Fig. 7: Traffic Volume vs rain_1h

Next, we plotted traffic volume against rain per hour as shown in Fig. 7. Since there were huge outliers and most of our values were small, we performed a log transformation on the data to compress it. The results show that rain does not seem to have much of an impact on traffic volume as there is no strong relationship.

We then plotted traffic volume against snow per hour in

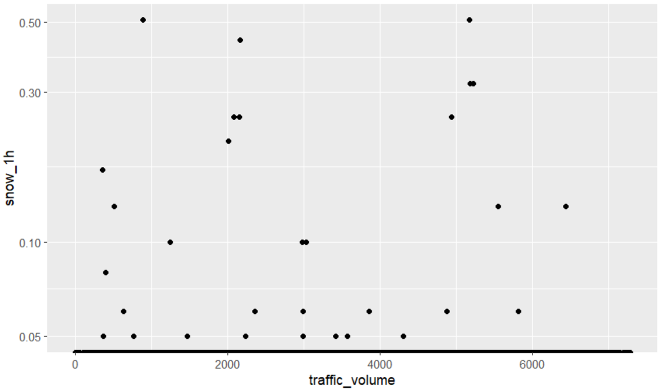


Fig. 8: Traffic Volume vs snow_1h

Fig. 8. Most points are around 0, indicating that there are few snow events. The data shows no clear relationship from the scatter plot. Indicating that snow is not a strong factor in determining traffic volume.

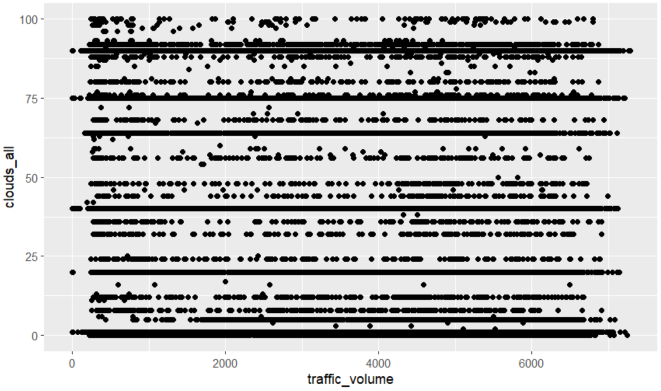


Fig. 9: traffic volume against clouds_all

The final scatter plot in Fig. 9 shows traffic volume against $clouds\_all$. Once again, there is no clear relationship that can be drawn from the scatter plot. The noise is random and there is no pattern. Thus, there is no strong relationship between the amount of clouds in the sky on the traffic volume.

From the scatter plots we observed, $hour$ seems to have the largest impact on the amount of traffic on the interstate. The last part of the EDA will consist of looking at the counts traffic volume.

The histogram in Fig. 10 shows the counts of all recorded traffic volumes. The traffic volume peaks at around 300. This most likely represents the small amount of traffic that occurs very early in the morning and at night. The second largest peak is slightly less than 5000. This most likely represents the rush hour that occurs when everyone leaves work around 5 p.m. The smallest peak most likely represents the traffic rush that occurs when everyone is heading to work in the morning at around 9 a.m.

From our EDA portion of the research, it seems that the largest factors that contribute to the traffic volume on I-94 are the time of day and the day of the week. Next we perform some data preprocessing techniques to clean and transform the data before building and testing our models.

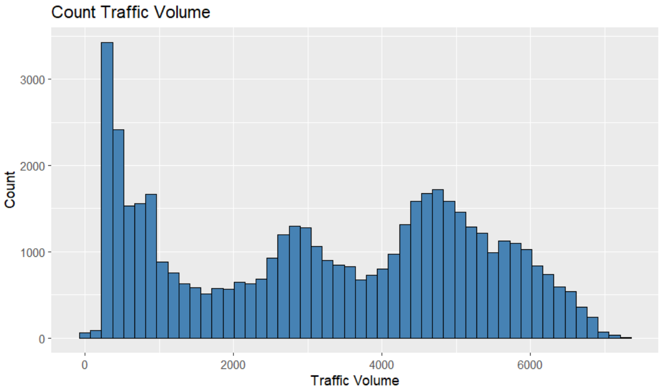
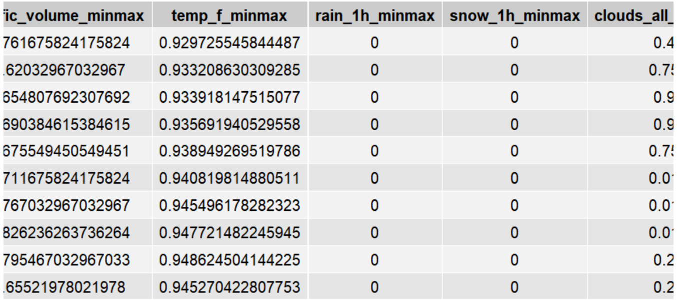


Fig. 10: Traffic Volume Count

IV. Data Preprocessing

There are three steps to this portion of the study. We will first normalize all of our numerical attributes, $traffic\_volume, temp\_f, rain\_1h, snow\_1h, clouds\_all$, using min-max normalization, z-scores, and decimal scaling. Next, we will select a continuous variable, $temp$, and bin it using equal bins, manual ranges, and k-means binning. The final step of Data Preprocessing includes identifying a variable that is not normally distributed and using the natural log, square root, and inverse square root to attempt to achieve normality.

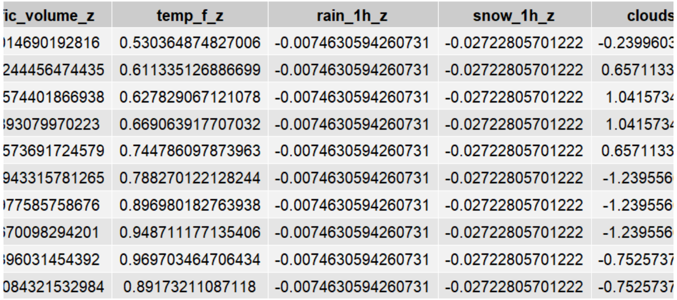


ïc_volume_minmax	temp_f_minmax	rain_1h_minmax	snow_1h_minmax	clouds_all_
761675824175824	0.929725545844487	0	0	0.4
.62032967032967	0.933208630309285	0	0	0.75
654807692307692	0.933918147515077	0	0	0.9
690384615384615	0.935691940529558	0	0	0.9
675549450549451	0.938949269519786	0	0	0.75
711675824175824	0.940819814880511	0	0	0.01
767032967032967	0.945496178282323	0	0	0.01
826236263736264	0.947721482245945	0	0	0.01
795467032967033	0.948624504144225	0	0	0.2
.65521978021978	0.945270422807753	0	0	0.2

Fig. 11: The first 10 rows after applying min-max normalization

To normalize all of our numerical values we first use min-max normalization as shown in Fig. 11. The goal of this method is to scale down our values from zero to one, where the minimum value gets transformed into a 0 and the maximum value gets transformed into a 1. The minimum value is subtracted from each value and then gets divided by the range. This method is sensitive to outliers.

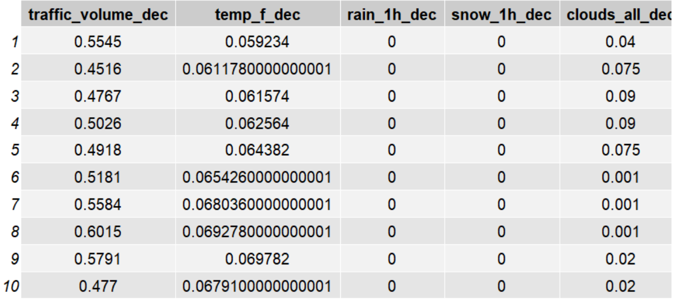
We then use z-score normalization as seen in Fig. 12. The goal of this method is to normalize each attribute so the mean



ic_volume_z	temp_f_z	rain_1h_z	snow_1h_z	clouds
14690192816	0.530364874827006	-0.0074630594260731	-0.02722805701222	-0.2399603
244456474435	0.611335126886699	-0.0074630594260731	-0.02722805701222	0.6571133
574401866938	0.627829067121078	-0.0074630594260731	-0.02722805701222	1.0415734
93079970223	0.669063917707032	-0.0074630594260731	-0.02722805701222	1.0415734
573691724579	0.744786097873963	-0.0074630594260731	-0.02722805701222	0.6571133
943315781265	0.788270122128244	-0.0074630594260731	-0.02722805701222	-1.239556
77585758676	0.896980182763938	-0.0074630594260731	-0.02722805701222	-1.239556
70098294201	0.948711177135406	-0.0074630594260731	-0.02722805701222	-1.239556
96031454392	0.969703464706434	-0.0074630594260731	-0.02722805701222	-0.7525737
084321532984	0.89173211087118	-0.0074630594260731	-0.02722805701222	-0.7525737

Fig. 12: The first 10 rows after applying z-score normalization

becomes 0 and the standard deviation to 1. For every value, the mean is subtracted. The result is the divided by the standard deviation. This produces a z-score, which tells us how many standard deviations the original value is from the mean. If the value is smaller than the mean, it will be negative. The value will be positive if it is above the mean. This is done for each attribute we have selected.



	traffic_volume_dec	temp_f_dec	rain_1h_dec	snow_1h_dec	clouds_all_dec
1	0.5545	0.059234	0	0	0.04
2	0.4516	0.0611780000000001	0	0	0.075
3	0.4767	0.061574	0	0	0.09
4	0.5026	0.062564	0	0	0.09
5	0.4918	0.064382	0	0	0.075
6	0.5181	0.0654260000000001	0	0	0.001
7	0.5584	0.0680360000000001	0	0	0.001
8	0.6015	0.0692780000000001	0	0	0.001
9	0.5791	0.069782	0	0	0.02
10	0.477	0.0679100000000001	0	0	0.02

Fig. 13: The first 10 rows after applying decimal scaling normalization

The last method we use to normalize the numerical attributes is decimal scaling as seen in Fig. 13. This method does not depend on mean or standard deviation at all. This method finds the largest value within the attribute. Every value is then divided by a power of 10 that results in the largest value falling between [-1,1]. As a result, every value falls within the range.

The next step of the study is to perform binning on the *temp* attribute. Binning allows us to convert continuous variables into groups to allow for further analysis. We first start by evenly splitting the temperature into 4 equal bins like Fig. 14. This method is quick and simple. Each bin contains approximately 12000 samples.

The next method we chose was to manually create a range for the temperature as seen in Fig. 15. Anything less than 40 F was considered very cold. 40-59 F was considered cold. 60-79 was considered warm and 80+ was considered very warm. The last binning method we chose was the k-means binning as seen in Fig. 16. The k-means algorithm clusters temperatures based on their similarity. This method assigns each data point to the nearest cluster. The algorithm then updates the clusters until they stabilize. The resulting clusters will then form bins accordingly. Thus, allowing the bins to follow the data distribution.

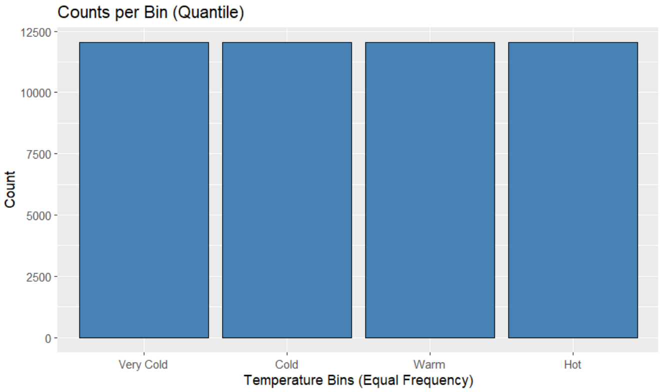


Fig. 14: Bins by evenly binning

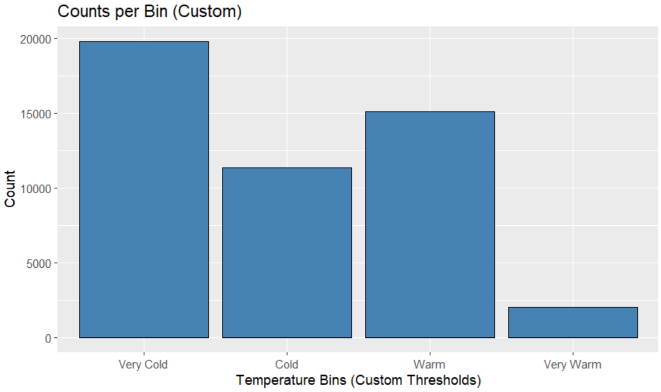


Fig. 15: Bins by manual range

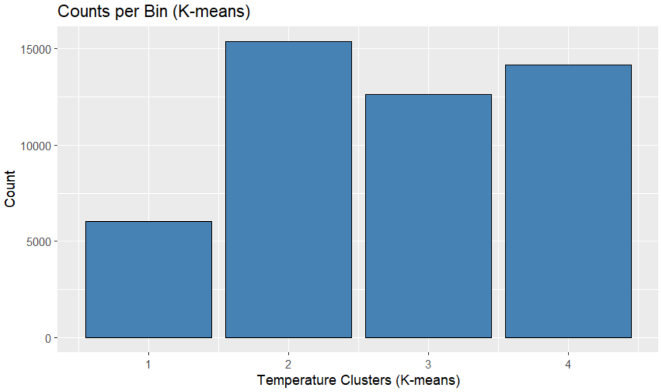


Fig. 16: K-means binning

Equal binning is fast and easy but ignores the distribution of the temperature. Manual binning allows easily interpretable groups, but depends on each human. The K-means method adapts to the temperature distribution and creates larger bins where there is more data. Thus, k-means is the best method.

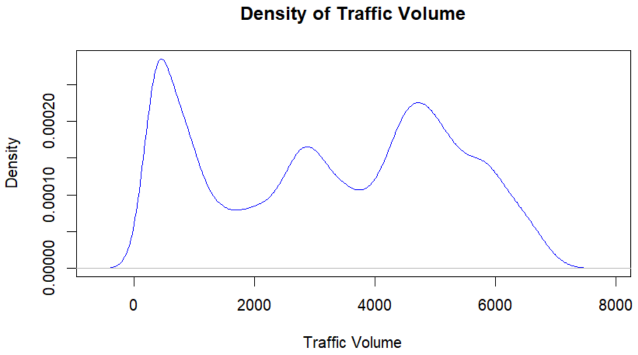


Fig. 17: Traffic volume density

The next step of the research was to find a variable that is not normally distributed and apply techniques to improve normality. We first plotted Fig. 17, and chose the attribute $traffic\_volume$ as it was right-skewed.

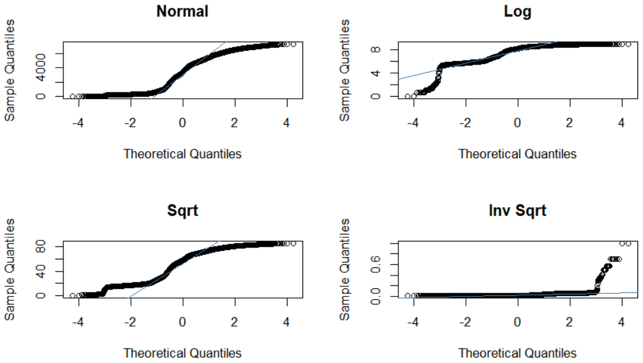


Fig. 18: Q-Q plots

We can see from the Q-Q plot that the traffic volume is heavily right-skewed. We apply the natural log, square root, and inverse square root transformations to achieve normality. From Fig. 18, we can see that the natural log greatly improved the normality of traffic volume. The right-skew is reduced and it follows the diagonal much better than the original. The square root improved the distribution slightly from the original, but worse than the natural log. The tails are still very bowed from the diagonal. The inverse square root performed very poorly and almost flattens the distribution. From this, the square root performed best in giving us a normal distribution among the four. After performing EDA and Data Preprocessing, we built and tested our models.

V. Regression Analysis

We chose to predict traffic volume using four models. These models include linear regression with ordinary least squares, decision tree regression, random forest regression, and gradient boosting regression. Each model was trained using the following attributes, temp_f, rain_1h, snow_1h, clouds_all, hour, weekday, month, and is_holiday. We used 70% of the data to train the model and the remaining 30% to test the model. To evaluate our models, we will use (R^2) and root mean square error$(RMSE)$.

A. Linear Regression

This was used as a baseline model. It attempts to predict the traffic volume by building and fitting a linear equation with coefficients that represent the contribution of each attribute. The model determines the best coefficients by minimizing the sum of squared error between the predicted and actual values.

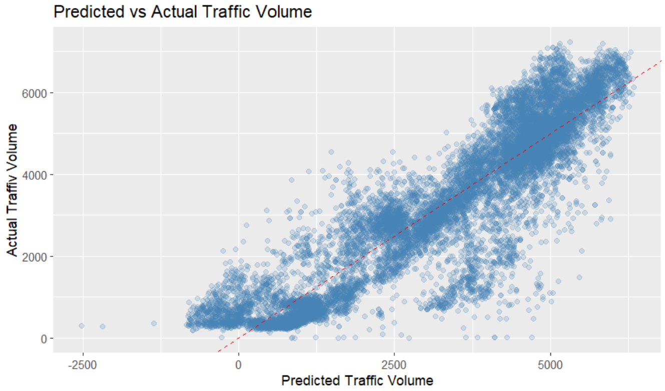


Fig. 19: Linear regression results

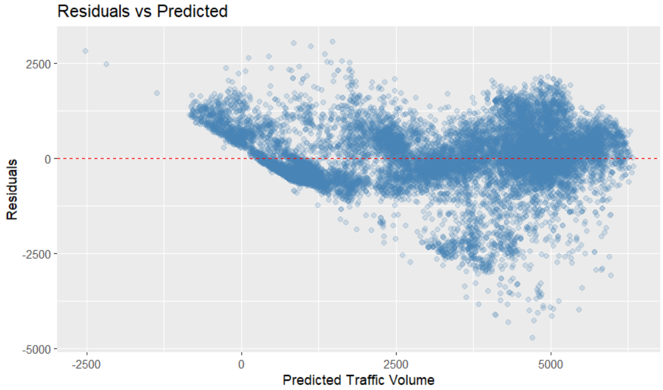


Fig. 20: Linear regression residuals

This baseline model performed fairly well. The (R^2) value is .832. This shows that about 83% of traffic volume is accounted for in our model. $(RMSE)$ was 813.176 indicating that our model's predictions are around 813 vehicles per hour away from the actual values. Fig. 19 and Fig. 20 show is the results and residuals.

B. Decision Tree Regression

The decision tree from Fig. 21 shows that *hour* is the most important attribute in the model. The tree first splits the data

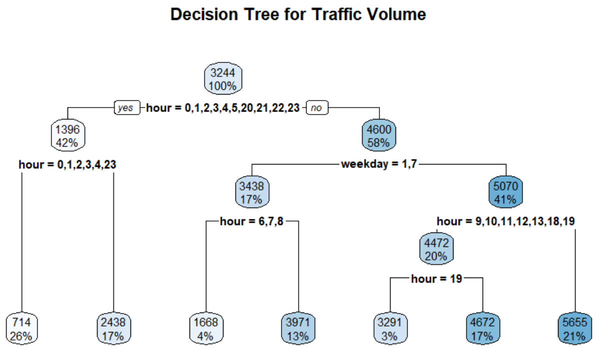


Fig. 21: Decision tree model

based on if the time of day is 5 a.m. or earlier or if it is after 8 p.m. $Weekday$ is the second most important attribute. The tree only splits the data up more after it has decided whether the time of day is early morning or night. If the $hour$ is not morning or night, the model then determines if the $weekday$ is Saturday or Sunday. If it is during the week, it then splits up the data again based on $hour$. The decision tree outperformed the linear regression model baseline. The (R^2) value is .878.

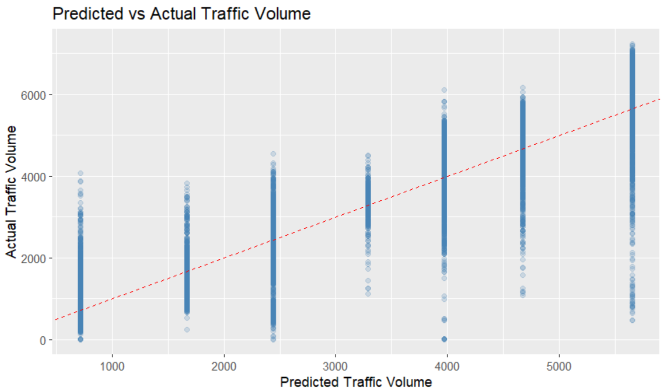


Fig. 22: Decision tree results

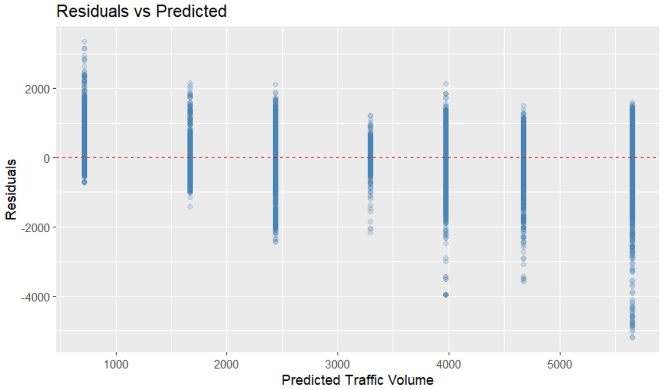


Fig. 23: Decision tree residuals

This shows that about 87.8% of traffic volume is accounted for in our decision tree regression model. The $RMSE$ is 693.183 indicating that our model's predictions are around 693 vehicles per hour away from the actual values. The results are shown in Fig. 22 and Fig. 23. Compared to the linear regression model, we have already improved (R^2) by .046 and $RMSE$ by 119.993.

C. Random Forest Regression

Random Forest builds many decision trees independently. Each tree is trained on a random subset of the data, and therefore gives its own prediction. The predictions are then averaged and the result becomes the final prediction. Due to the building of multiple decision trees, this approach is expected to give a more accurate prediction of traffic volume. The left figure in Fig. 24 shows how much the % increase in mean squared error when each attribute is permuted. The right figure in Fig. 24 shows how much each attribute helps the variance when splitting the data. Once again, we see that $hour$ is the most important attribute, followed by $weekday$. The (R^2) value is .936. Meaning that about 93.6% of traffic volume is accounted for in our random forest regression model. The $RMSE$ is 502.596 indicating that our model's predictions are around 503 vehicles per hour away from the actual values. The

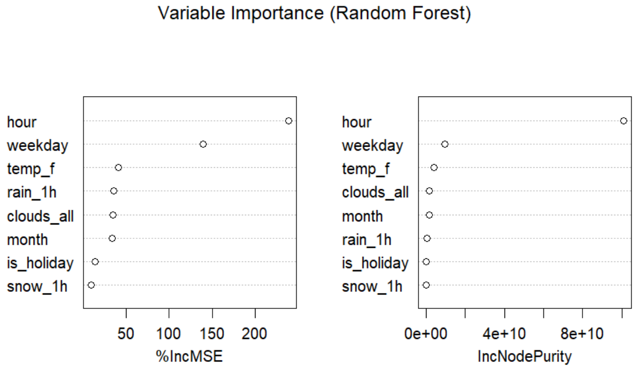


Fig. 24: Random forest summary

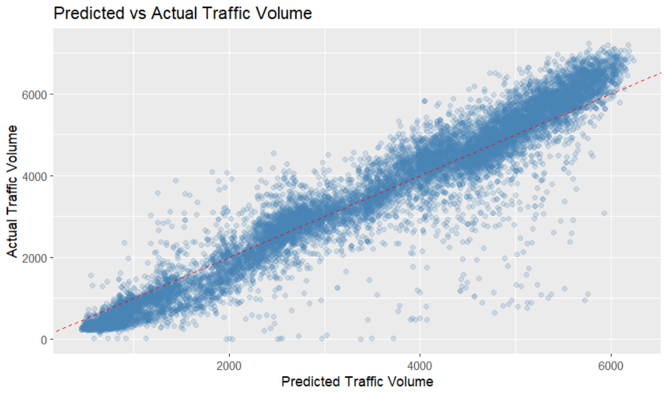


Fig. 25: Random forest results

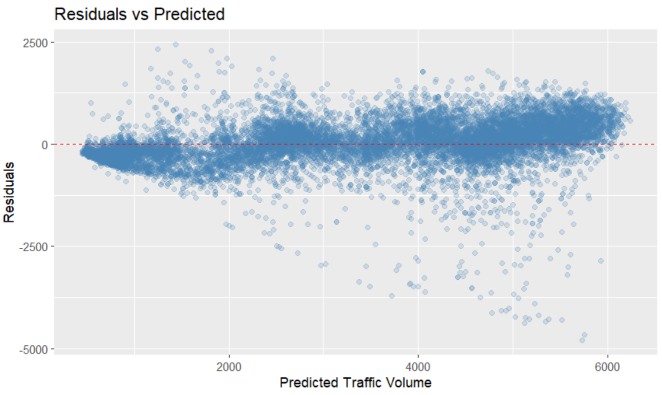


Fig. 26: random forest residuals

results are shown in Fig. 25 and Fig. 26. Compared to the linear regression model and decision tree, we have improved (R^2) by .104 and .058 respectively. We have also improved the $RMSE$ by 310.580 and 190.587 as well.

D. Gradient Boosting Regression

In Gradient Boosting Regression, there is a sequential learning process. The first trees are trained on the original data. The residuals are then used to train the next trees. Every new tree corrects the error from the previous tree, which improves accuracy with each iteration. The (R^2) value for this model is .926. Meaning that about 92.6% of traffic volume is accounted for in our gradient boosting regression model. The $RMSE$ is 541.116 indicating that our model's predictions are around 541 vehicles per hour away from the actual values, a slight decrease from our random forest model. The results are shown in Fig. 27 and Fig. 28. Compared to the linear regression model and decision tree, we have improved (R^2) by .094 and .048 respectively. However, we have decreased compared to our random forest model by .010. We have improved the $RMSE$ by 272.060 and 152.067 compared to our linear regression and decision tree models respectively. The $RMSE$ slightly decreased by 38.520 compared to our random forest model. However, the runtime for our gradient boosting model was quicker than our random forest method.

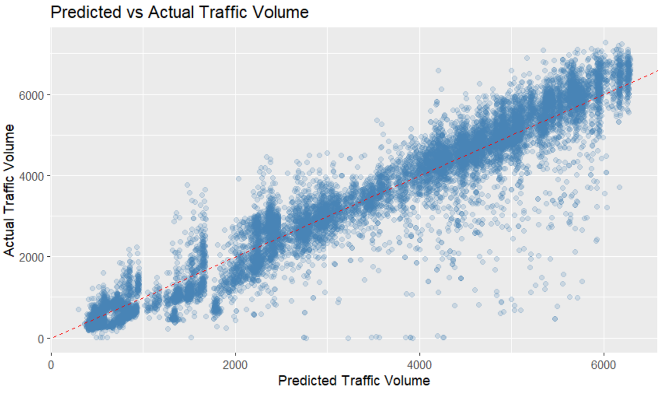


Fig. 27: Gradient boosting results

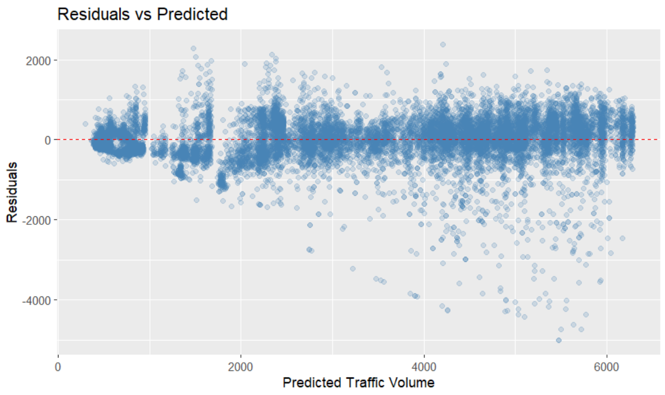


Fig. 28: Gradient boosting residuals

VI. Conclusion

In this project, we used several regression models to predict the traffic volume along I-94. We performed EDA to understand the relationships between the attributes. We also did some data preprocessing to attempt to clean and normalize the dataset. From there, we built our models to predict traffic volume. The baseline model was the linear regression model. The next model we trained and tested was the decision tree. This model performed better than our baseline model. We then built our random forest regression model to train and test as well. This model performed better than the previous two models. The final model that we built was gradient boosting. This model performed slightly worse than the random forest model, but was faster in training and testing. The most significant attributes in determining traffic volume were $hour$ and $weekday$. Overall, the random forest model proved to produce the most accurate results for our dataset.

References

[1] Dua, D. and Graff, C. (2019). *UCI Machine Learning Repository: Metro Interstate Traffic Volume Dataset*. Available: https://archive.ics.uci.edu/dataset/492/metro+interstate+traffic+volume

[2] GeeksforGeeks, "Gradient Boosting in Machine Learning," Mar. 2023. [Online]. Available: https://www.geeksforgeeks.org/machine-learning/ml-gradient-boosting/

[3] GeeksforGeeks, "Random Forest Regression in Python," [Online]. Available: https://www.geeksforgeeks.org/machine-learning/random-forest-regression-in-python/. Accessed: Oct. 6, 2025.

[4] Scikit-learn, "Plotting a Decision Tree for Regression," [Online]. Available: https://scikit-learn.org/stable/auto_examples/tree/plot_tree_regression.html. Accessed: Oct. 6, 2025.

[5] R-Bloggers, "Random Forest in R," Apr. 2021. [Online]. Available: https://www.r-bloggers.com/2021/04/random-forest-in-r/. Accessed: Oct. 6, 2025.

[6] R Documentation, "gbm: Generalized Boosted Regression Models (R Package 'gbm')," [Online]. Available: https://www.rdocumentation.org/packages/gbm/versions/2.2.2/topics/gbm. Accessed: Oct. 6, 2025.

[7] "Gradient Boosting Algorithm Explained — Machine Learning Tutorials," YouTube, uploaded by StatQuest with Josh Starmer, [Online]. Available: https://www.youtube.com/watch?v=3CC4N4z3GJc. Accessed: Oct. 6, 2025.