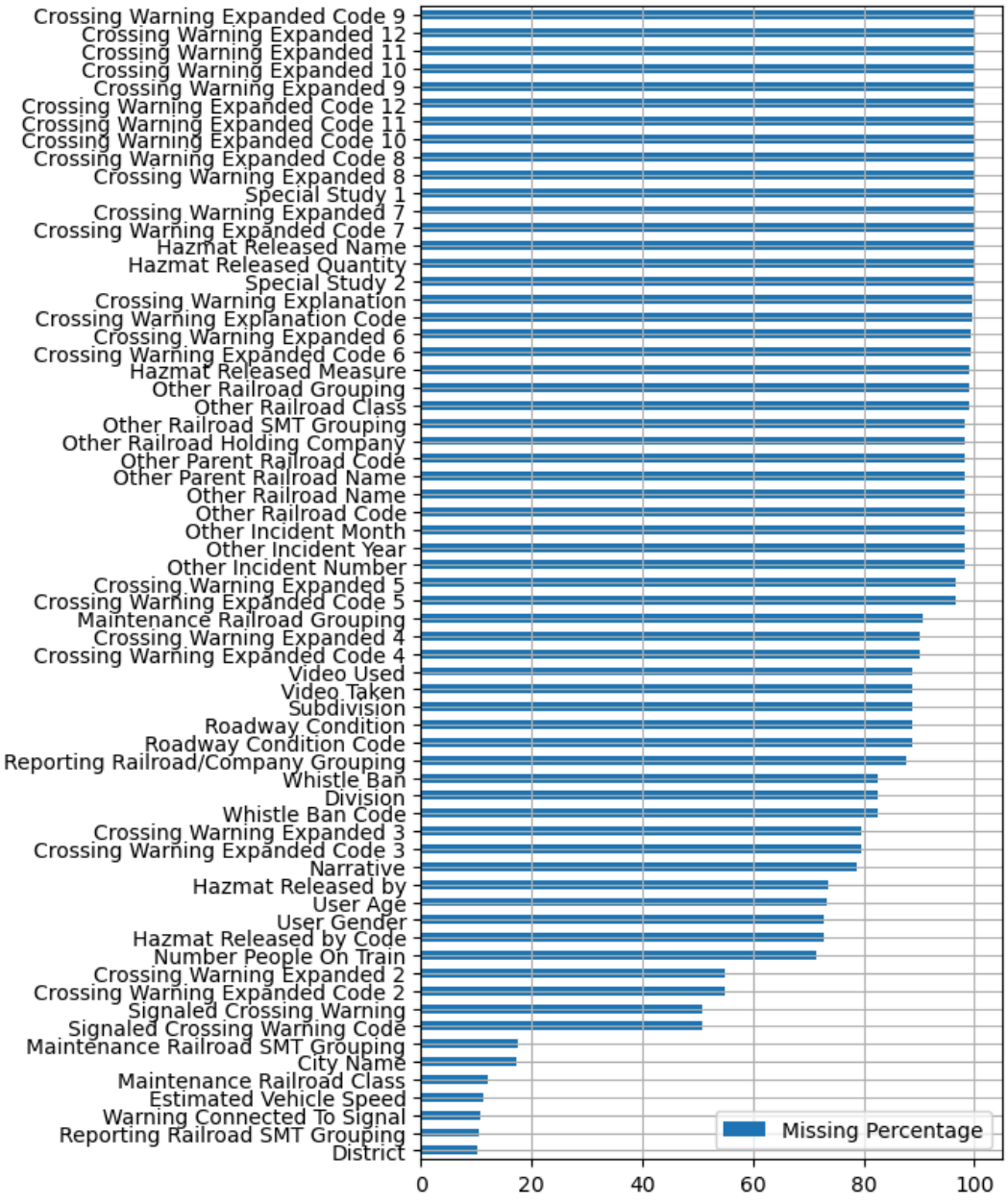
1. Task One - Data Cleaning and Preprocessing:

In the following section, we aim to load the data, preprocess it, handle issues related to missing values, and make it ready for further analysis. This dataset contains 160 columns and 24,663 rows.

We first check for missing values in the dataset by calculating the percentage of missing data in each column. We define a threshold, `tol = 10`, to show only the columns that have more than 10% missing data. The columns are then sorted by their missing values.

In the picture below, you can clearly observe the columns containing missing values.



Several columns have a very high percentage of missing data, with some columns having nearly 100% missing values (such as "Crossing Warning Expanded Code 9", "Crossing Warning Expanded 11".

Data Cleaning and Convesion:

We aim to creates a new column `DateTime` by combining the `Date`, `Hour`, `Minute`, and `AM/PM` columns into a single string. The result is a date-time string in the format `MM/DD/YYYY HH:MM AM/PM`.

We conver the created `DateTime` string into a proper `datetime` object using the specified format. The `errors='coerce'` parameter means that any rows where the conversion fails will be set to `NaN`.

Finally we remove any rows where this conversion was unsuccessful.
The strategy of drooping coulmns is described as:

Dropping Redundant Columns:

- **Date-Related Redundancies**: Columns like `'Report Year'`, `'Incident Year'`, `'Incident Month'` were dropped because their information is already contained within the `'Date'` column.
- **Sealdomly filled columns:** Some Colums like 'Hazmat Released Name', 'Hazmat Released Quantity', 'Hazmat Released Measure', 'Crossing Warning Expanded 2-12' were dropped as the majority of them were not filled.
- **Combining and Dropping Date and Time Components**:
- The columns `'Date'`, `'Month'`, `'Day'`, `'Hour'`, `'Minute'`, and `'AM/PM'` were combined into a single datetime column, after the original components were dropped to avoid duplication.

Handling Duplicates:

- Several pairs of columns were identified as duplicates, and one of each pair was removed to prevent redundant data from being stored.

Removing Sparse or Unique Columns:

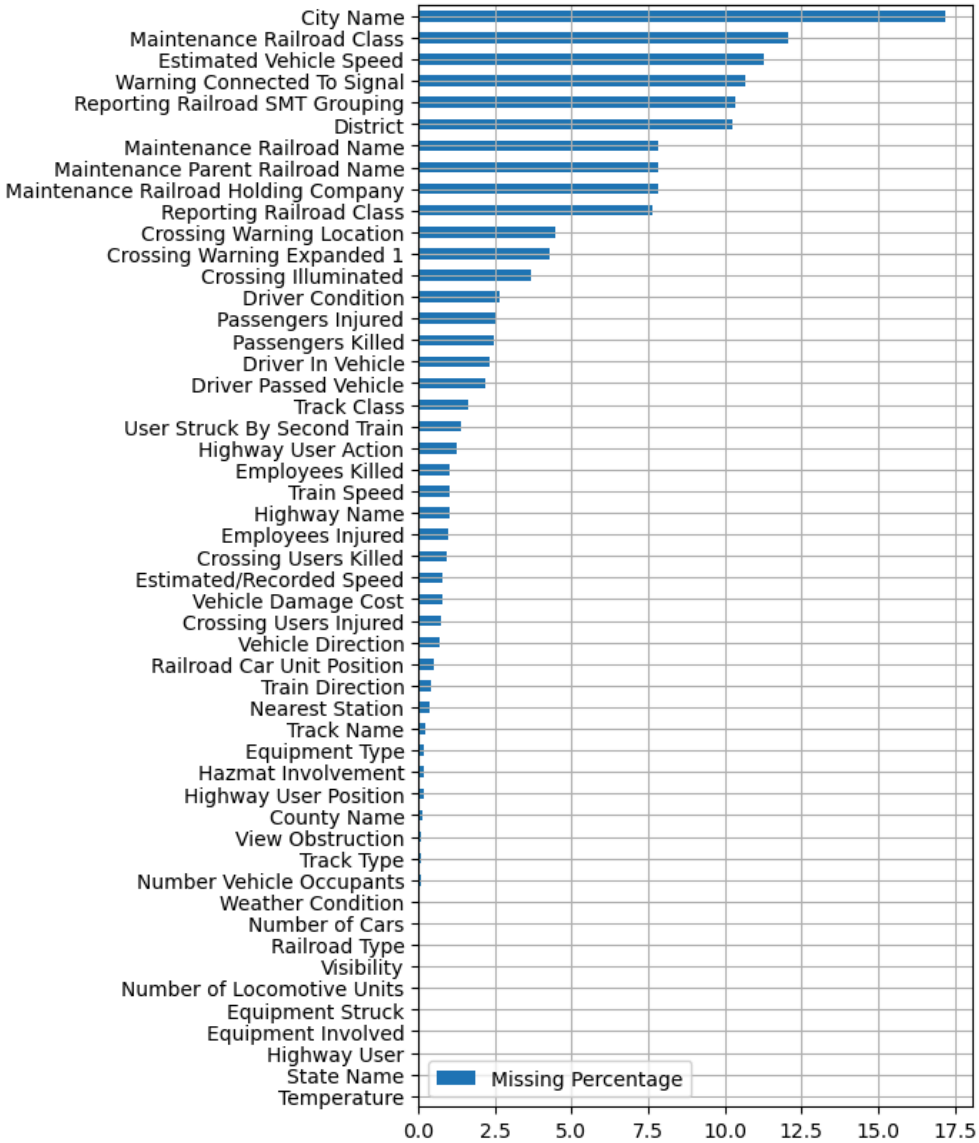
- Columns like `'Form 54 Filed'`, `'Grade Crossing ID'`, and `'Report Key'` were dropped because they were unrelated to the analysis or contained unique identifiers that did not add value to the dataset.

Finally we got two types of datas, Numerical datas and Cetegorical datas. They are listed in the notebook. And we did final check for any duplication in the modified dataset.

As it is obvious, in the beginning, the memory usage was about 301MB, but after data cleaning, it reduced to 49MB, which indicates that the process significantly reduced the dataset by removing unnecessary data.

Handeling Missing Values

We plotted the percentage of missing values for each column again, as shown in the image below.



Since we have both categorical and numerical data, we need to take different approaches to handle the missing values. We decided to handle missing categorical values by replacing them with the mode and to address missing numerical values using the BayesianRidge imputer. BayesianRidge imputer is handling missing values in numerical columns of a DataFrame using an iterative imputation strategy, where each missing value is predicted based on the relationships among other features in the dataset. The `BayesianRidge` regression model is used to make these predictions.

2. Task two - Exploratory Data Analysis (EDA):

Statistical summary of numerical columns:

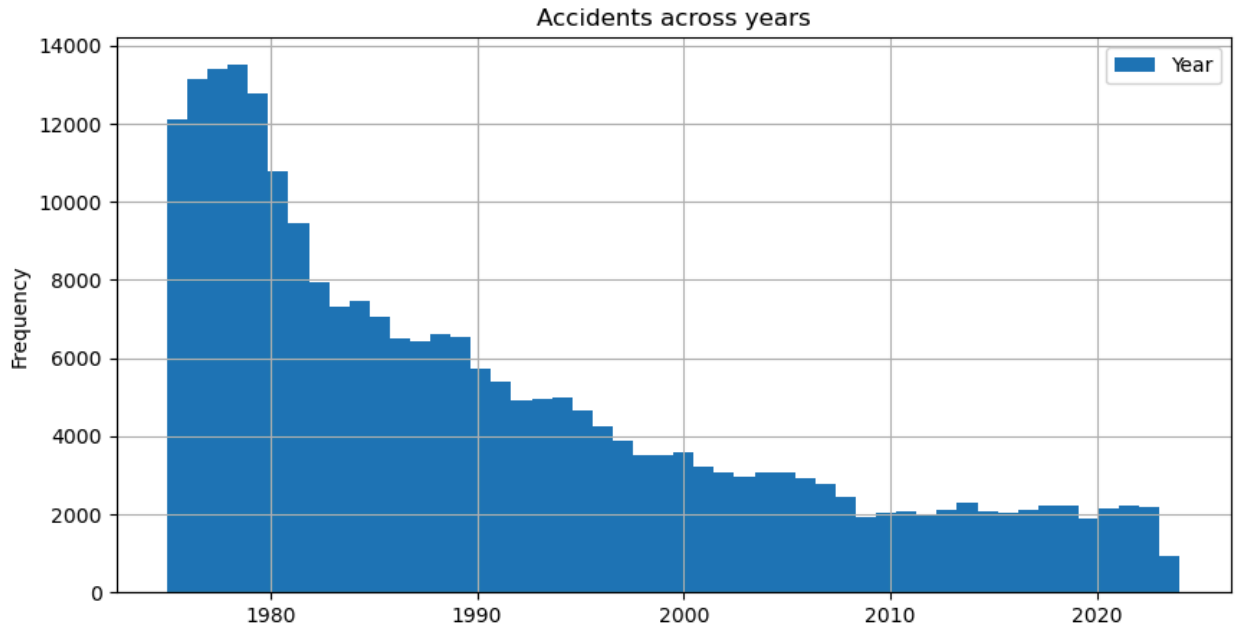
The following picture shows some of these statistics. The complete results can be observed in the notebook.

	Estimated Vehicle Speed	Temperature	Number of Locomotive Units	Number of Cars	Train Speed	Crossing Users Killed	Crossing Users Injured
count	246532.000000	246532.000000	246532.000000	246532.000000	246532.000000	246532.000000	246532.000000
mean	11.486941	56.478835	2.174939	42.117121	24.497639	0.100818	0.339010
std	12.985927	23.133218	1.591797	41.883823	18.171125	0.368664	0.705263
min	-2.544103	-99.000000	0.000000	0.000000	0.000000	-0.139765	0.000000
25%	0.000000	40.000000	1.000000	6.000000	8.000000	0.000000	0.000000
50%	9.171805	60.000000	2.000000	27.000000	22.361600	0.000000	0.000000
75%	15.000000	75.000000	3.000000	73.000000	38.000000	0.000000	1.000000
max	510.000000	870.000000	392.000000	944.000000	750.000000	12.000000	41.000000

From the showen summary we can understand that many fields, particularly those related to fatalities and injuries, have medians and lower quartiles at zero, indicating that the majority of incidents may not involve these outcomes. This suggests a skewed distribution where most entries are non-incidents, with a few records reporting significant events. Also, standard deviations are high in several columns, such as vehicle speed and the number of cars, indicating wide variability in these attributes.

Visualize the distribution of key variables and analyzing relations:

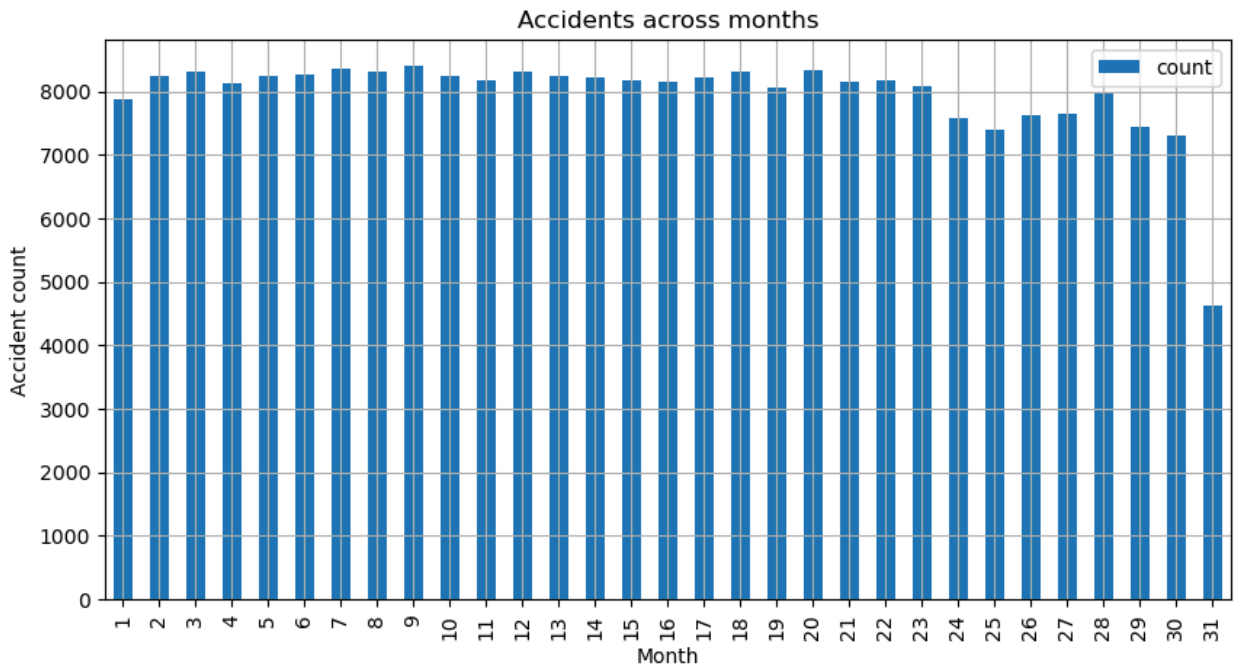
In this part we tried to plot some key variables and analyze their distribution.



The first plot shows the frequency of accidents over the years. As is obvious, the distribution is skewed, meaning that the rate of accidents was quite high around 1980 to 1990. We can then make the following observations.

Observations:

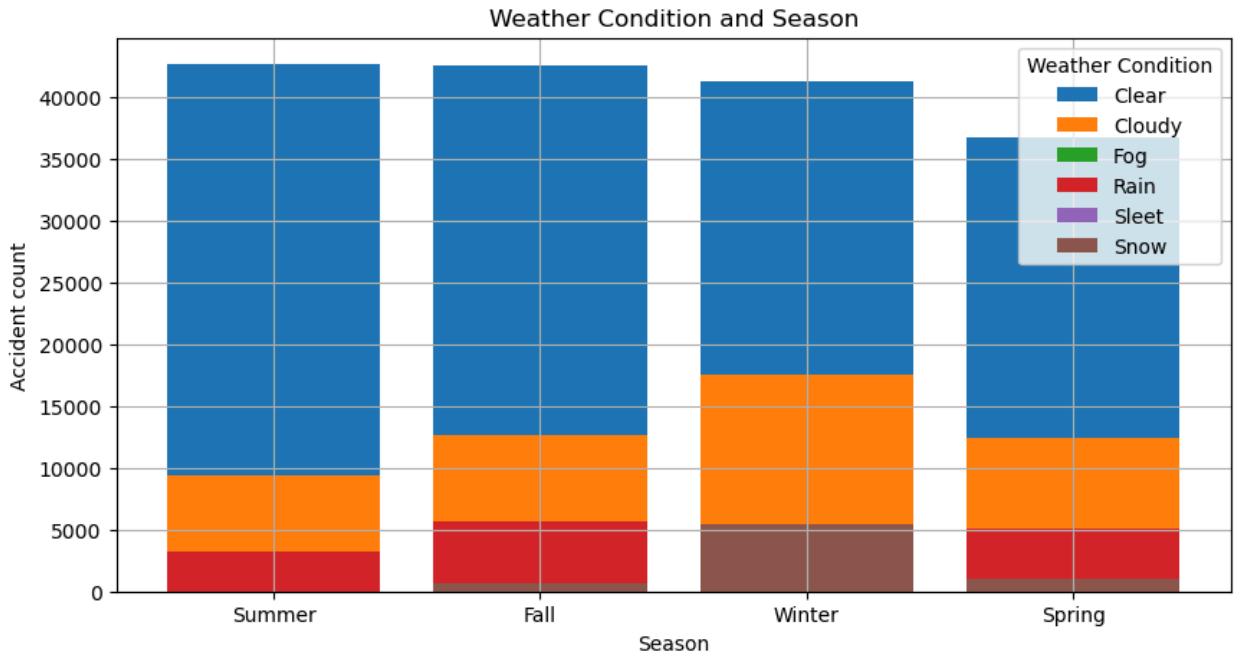
- **Decreasing Trend**: There is a clear and significant decrease in the number of railroad crossing accidents from 1975 to the early 2000s. The decline is steep, particularly in the late 1970s and throughout the 1980s.
- **Stabilization**: After around 2001, the number of accidents appears to stabilize, maintaining a relatively consistent level between 1,500 and 2,500 accidents per year.
- **Minor Fluctuations**: While the overall trend is stable after 2001, there are minor year-to-year fluctuations, with no significant upward or downward trends observed in the past two decades.
- **Recent Years:** The data for 2024 shows a lower accident count, which might be incomplete.



The second picture illustrates the accident rate across month. According to the plot We can make the following observations.

Observations:

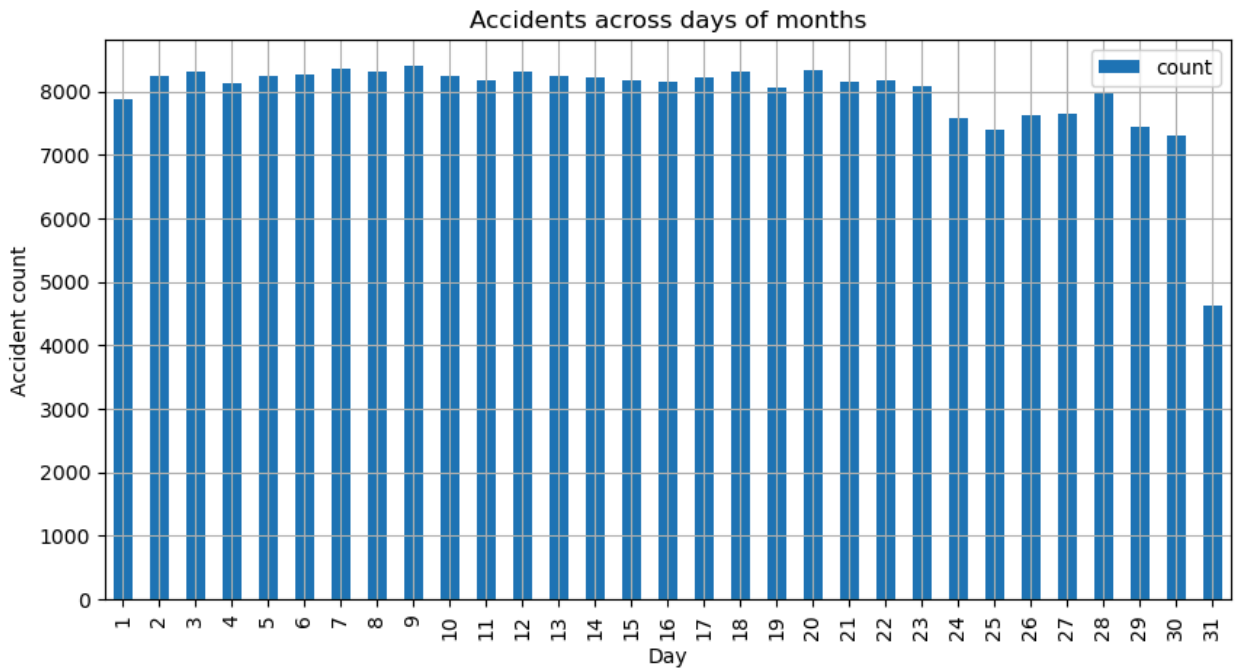
- **January and December Peaks:** The highest number of railroad crossing accidents occurs in January and December, with both months showing counts above 25,000.
- **Winter Months**: The winter months (January, February, and December) generally have higher accident counts compared to other months, which may be influenced by adverse weather conditions such as snow, ice, or reduced visibility.
- **Spring and Summer Dip**: Accident counts are generally lower during the spring and summer months (April to July), with the lowest counts observed around June and July.
- **Autumn Increase**: There is a noticeable increase in accidents starting in October and continuing through November and December, suggesting that accidents become more frequent as the year progresses towards winter.
- **Consistent Lower Counts**: The months of April through September show a relatively consistent and lower number of accidents, indicating possibly more favorable conditions during these months.



The Third picture shows the accident number with respect to the wether condition. According to the plot we can make the following observations.

Observations:

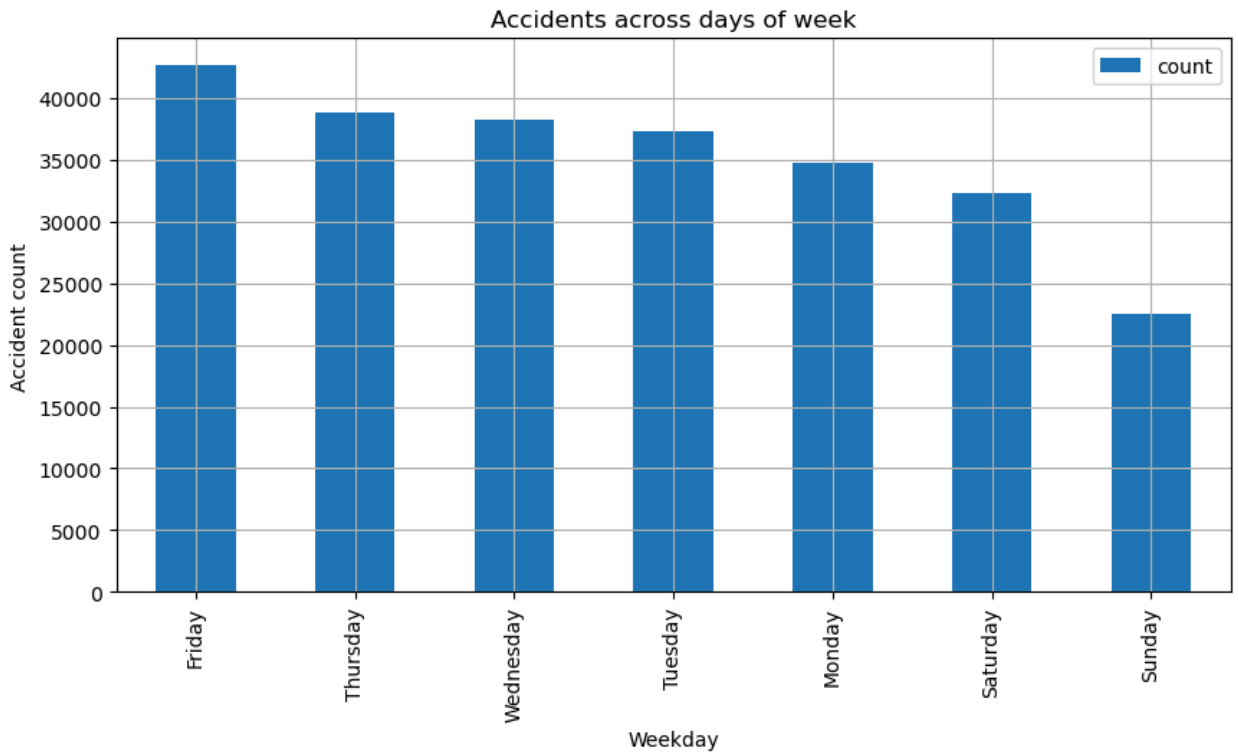
- **Seasonal Effects:** Seasons do not have a clear effect on accident numbers.
- **Clear Weather Dominance:** In all seasons, the majority of railroad crossing accidents occur under clear weather conditions, with clear weather consistently accounting for the highest accident counts.
- **Cloudy Conditions:** Cloudy weather is the second most common condition associated with accidents, following clear weather, across all seasons.
- **Winter Season Variability:** Winter shows a wider variety of weather conditions associated with accidents compared to other seasons. In addition to clear and cloudy weather, there are noticeable contributions from snow.



The following picture shows the accident rate across days of months. According to the plot we can make the following observations.

Observations:

- **General Distribution:** The number of accidents is relatively consistent across most days of the month, with counts generally around 7,500 to 8,500.
- **31st Day:** There is a significant drop in the number of accidents on the 31st day of the month, likely because many months do not have a 31st day.
- **End of Month:** A gradual decline in accidents is observed from the 25th to the 30th, before the sharp drop on the 31st.
- **Peak:** The 18th and 22nd days show slightly higher accident counts compared to other days.

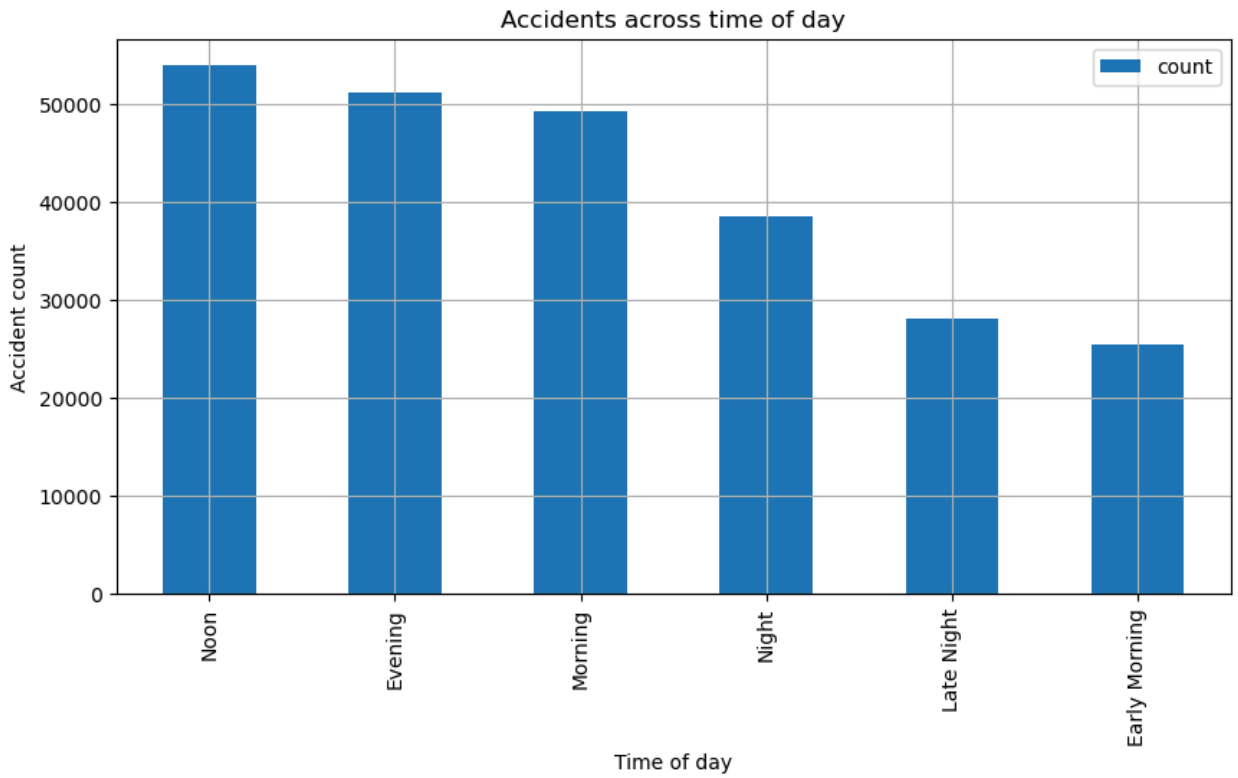


In the next step we tried to plot the accident rate across days of weeks. And we observed that:

- **Highest Accident Day:** Friday has the highest number of accidents, indicating that the end of the workweek is particularly risky for railroad crossing incidents.
- **Lower Accident Days:** Sunday has the lowest number of accidents, followed by Saturday, indicating that weekends generally see fewer incidents, likely due to reduced traffic.
- **Weekday vs. Weekend:** There is a clear distinction between weekdays and weekends, with significantly more accidents occurring on weekdays, reflecting higher levels of activity and possibly more frequent train and vehicle interactions.

We then tried to analyzed the plot deeper, so we analyze the of accident rates at different times of the day.

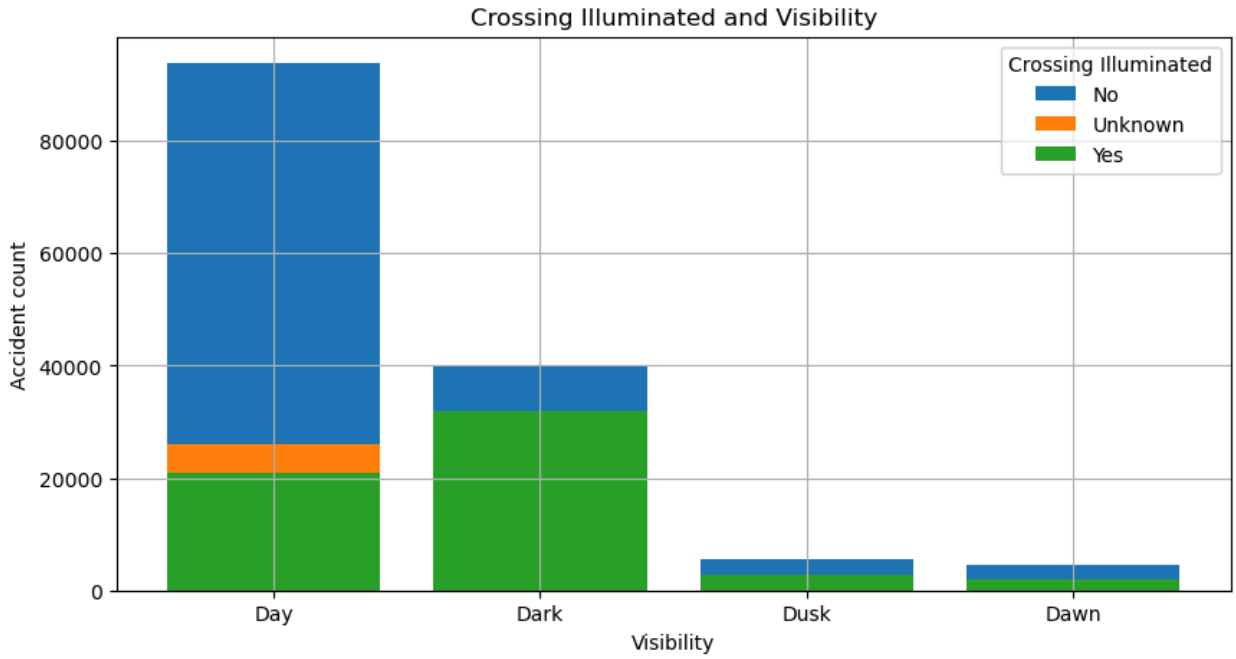
The following picture depicts the accident rate in a different time of a day.



Observations:

- **Peak Accident Times:** The highest number of accidents occurs at noon, followed closely by the evening.
- **Morning and Night:** The morning and night time periods have moderately high accident counts, slightly lower than noon and evening but still significant.
- **Noon and Evening High Accident Count:** High traffic volumes during midday and evening rush hours likely contribute to the increased accident rates during these times.
- **Late Night and Early Morning Low Accident Count:** Reduced traffic and potentially fewer trains operating during these times may account for the lower accident rates.

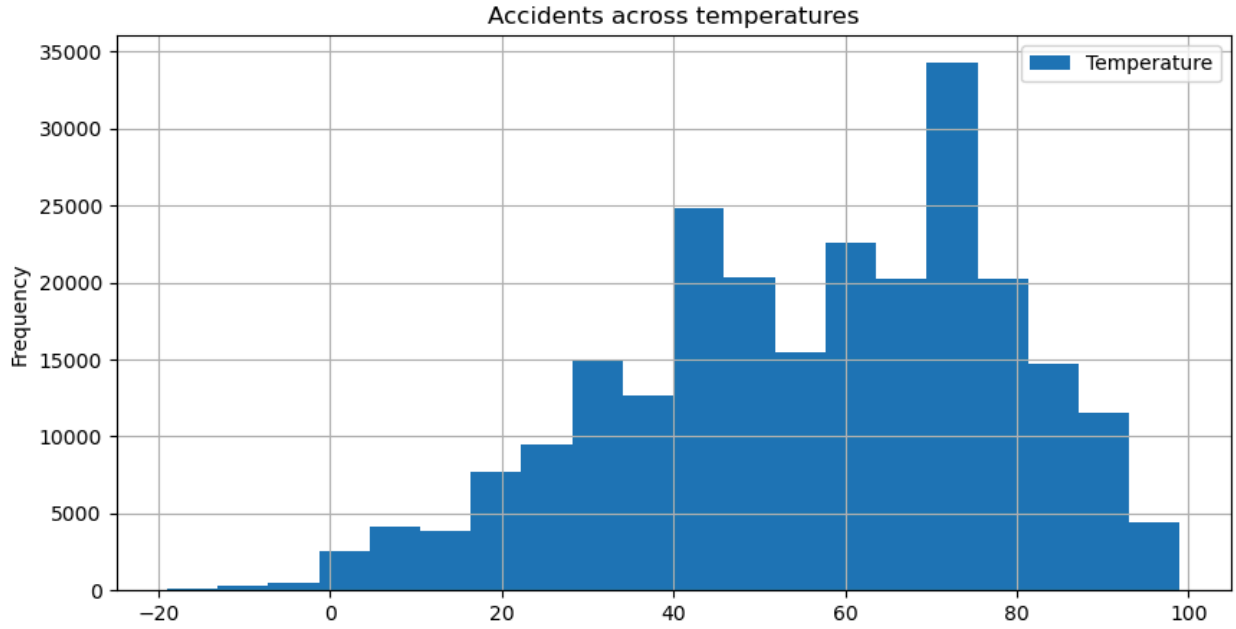
In the next step, we tried to analyze the relationship between visibility conditions (Day, Dark, Dusk, Dawn) and whether crossings were illuminated during accidents.



Observations:

- **Daytime Accidents:** The majority of accidents occur during the day, with a large portion happening at crossings that are not illuminated. This suggests that daylight visibility alone does not prevent accidents.
- **Dark Conditions:** During dark conditions, a significant number of accidents occur even at illuminated crossings, indicating that illumination alone may not be sufficient to prevent accidents in low visibility situations.
- **Illumination Impact:** The presence of illumination at crossings does not show a clear reduction in accidents, particularly at night, which might indicate that other factors, such as driver attentiveness or crossing design, play a more critical role in accident prevention.

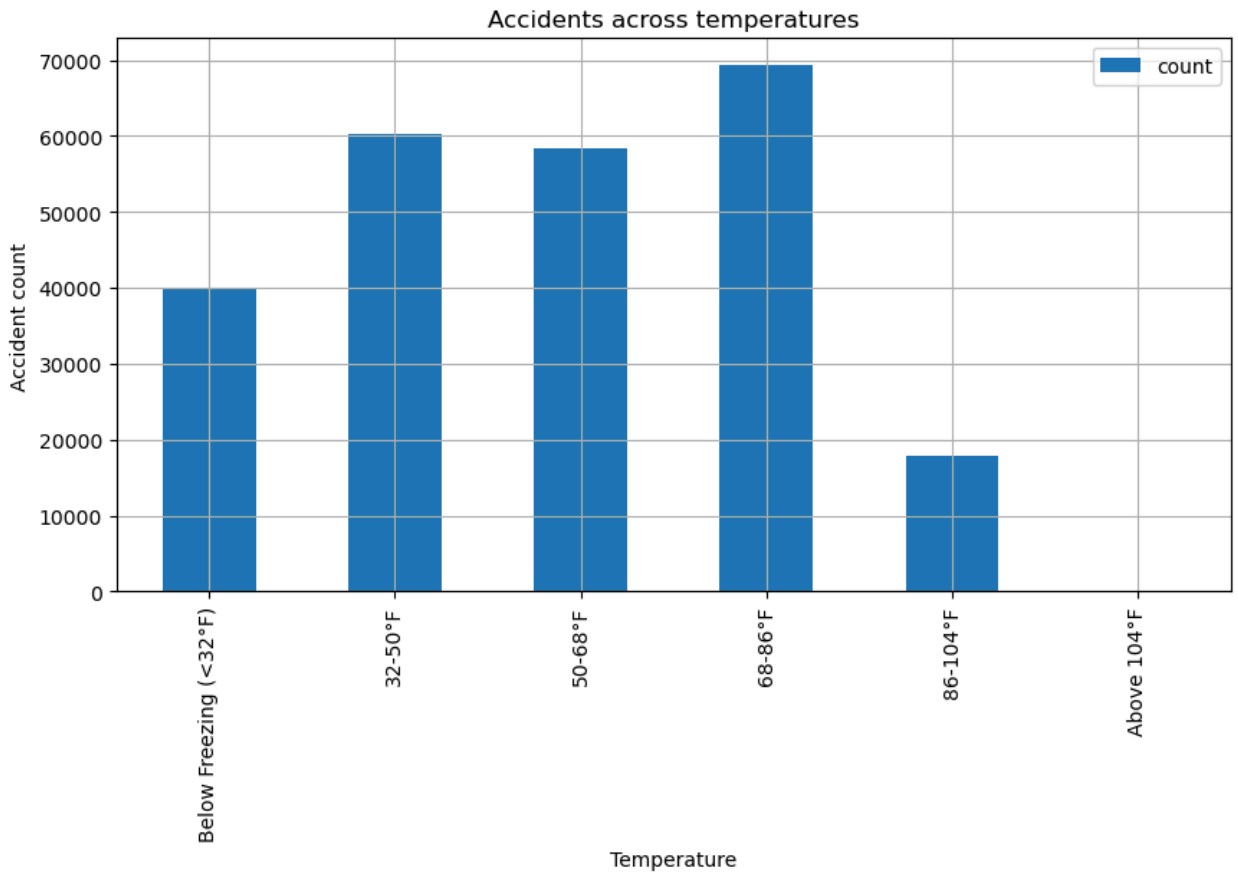
We then aim to analyze the effects of temperatures on accident rate.



To enhance our understanding, we created a categorical representation of temperature data and visualized the distribution of accidents across these temperature categories. We grouped the temperatures into the following defined bins and assigned labels to each:

- `bins = [-10, 32, 50, 68, 86, 104, 122]`
- `labels = ['Below Freezing (<32°F)', '32-50°F', '50-68°F', '68-86°F', '86-104°F', 'Above 104°F']`

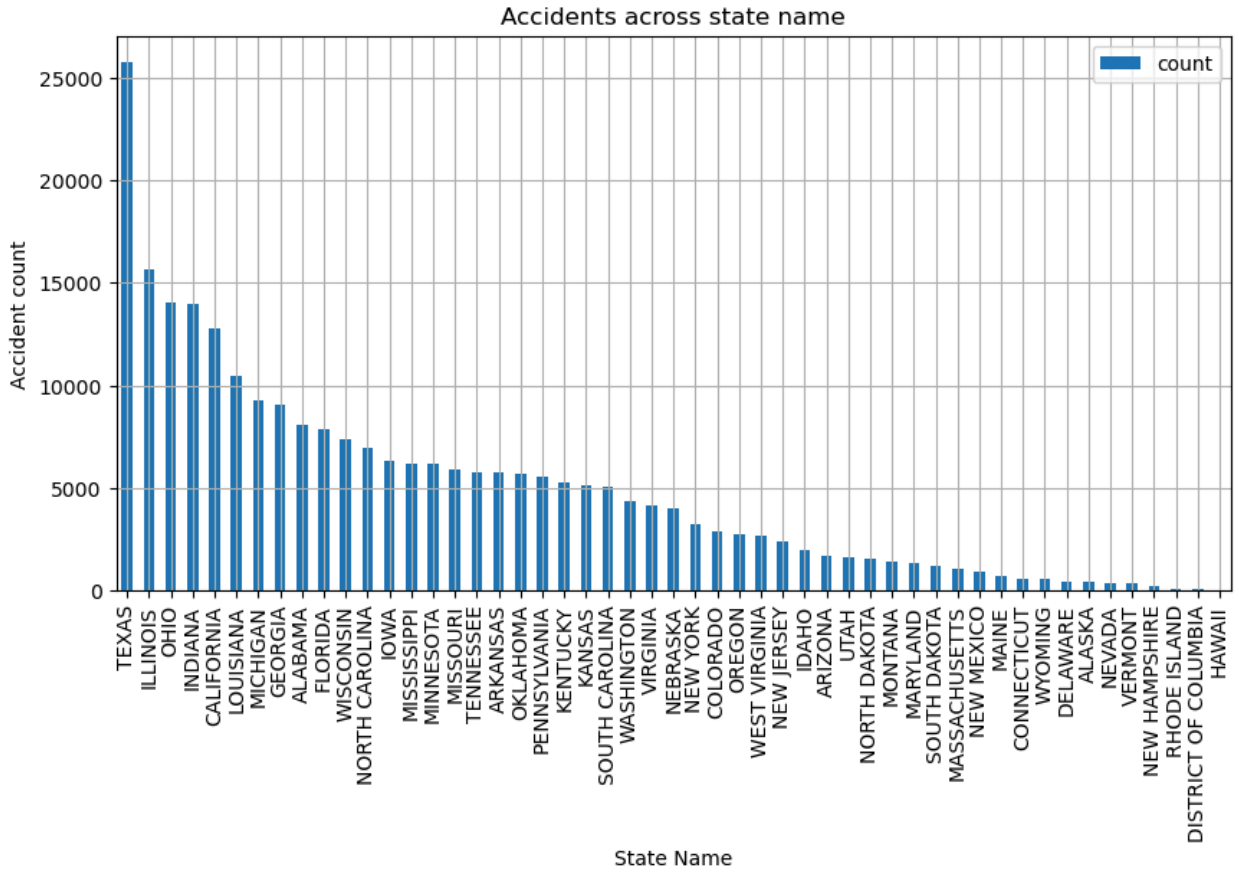
In the following picture we can observe the result.



Observations:

- Most Common Accident Temperatures: The highest number of accidents occurs in the temperature range of 68-86°F, indicating that warmer weather might correlate with increased accident frequency.
- Extreme Heat: Very few accidents occur at temperatures above 104°F, suggesting that extreme heat might lead to reduced travel or better adherence to safety due to uncomfortable conditions.
- Temperature Impact: The data indicates that railroad crossing accidents are more common in moderate to warm temperatures, with a decrease in very cold and very hot conditions. This could be linked to general traffic patterns, weather-related factors, or behavioral changes in extreme temperatures.
- Outliers: There seems to be some outliers in temperatures above 100°F and under -20°F

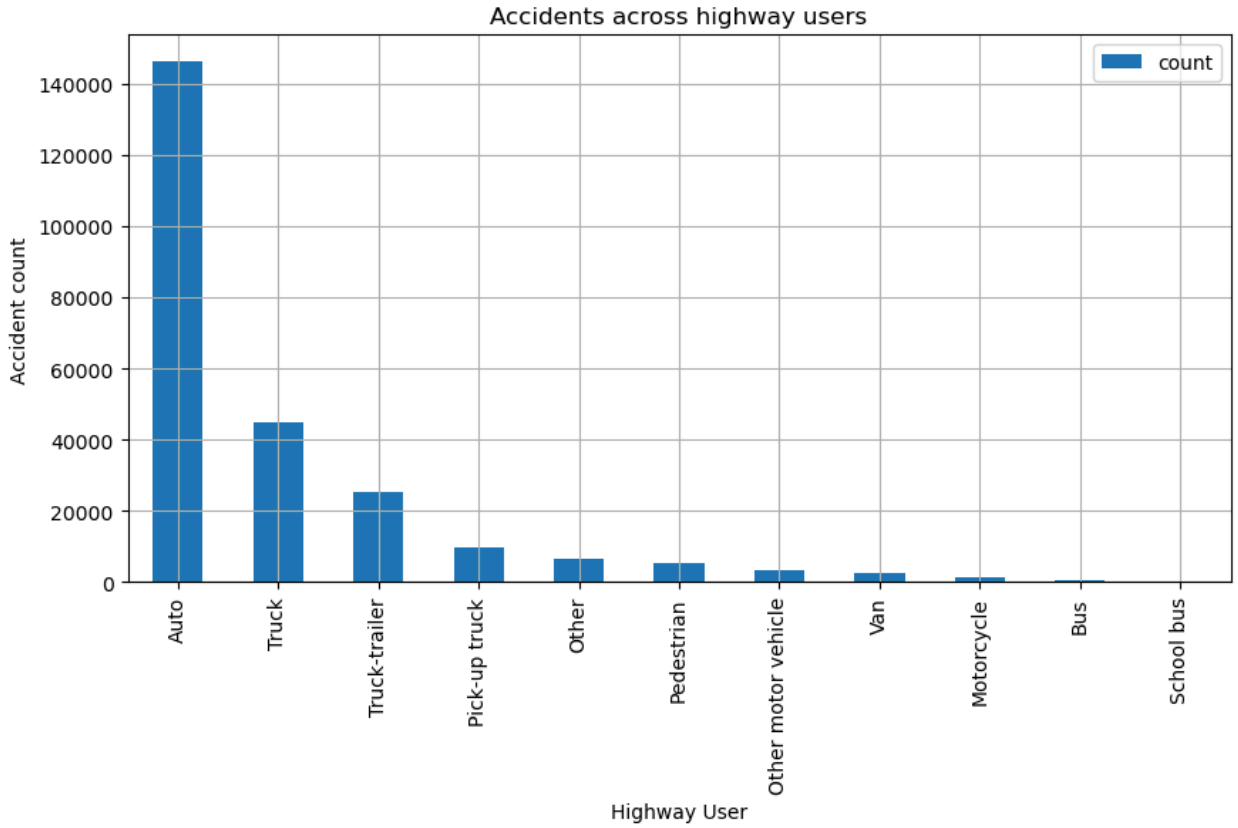
Since it is clear taht accident rate is not consistent across different state, we tried to find the accident rate in each state.



Observations:

- **Top States:** Texas has the highest number of railroad crossing accidents, followed by Illinois, Ohio, and Indiana. Texas, has a large land area and population, which could contribute to a higher number of crossings and thus more accidents
- **High Concentration:** The top 10 states (Texas to Georgia) have significantly higher accident counts compared to other states, with a steep drop-off after the top few states.
- **Distribution:** The distribution is heavily skewed, with a small number of states contributing to the majority of accidents, while most states have relatively few incidents.

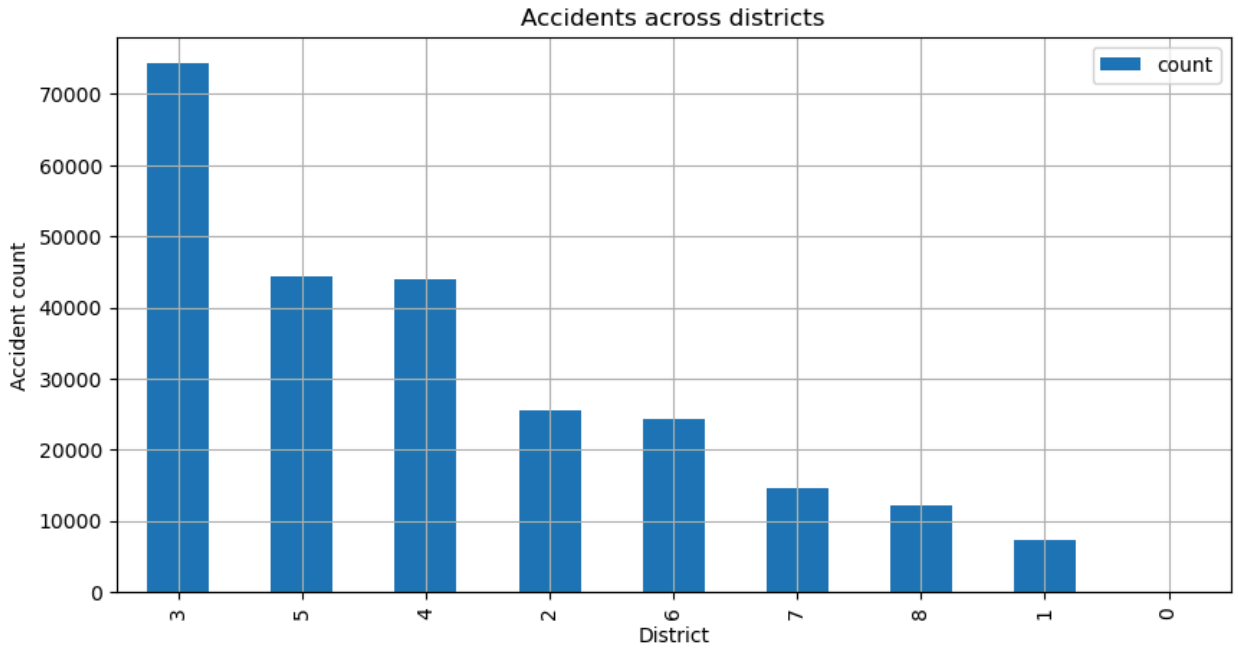
In the next step we aim to find the accident rate with respect to each vehicel, we believe that types of vehicle may have an effect of accident occurance.



Observations:

- **Dominance of Autos:** The vast majority of accidents involve "Auto" vehicles, significantly higher than any other category.
- **Trucks and Truck-Trailers:** Trucks and truck-trailers are the next most common vehicle types involved in accidents, though their counts are much lower than autos.
- **Steep Drop-off:** There's a steep drop-off in accident counts after the top three categories (Auto, Truck, Truck-trailer), with all other categories having relatively minimal impact.
- **Infrastructure Implications:** The data suggests autos are most at risk at railroad crossings, possibly due to their higher numbers on the road compared to other vehicle types.

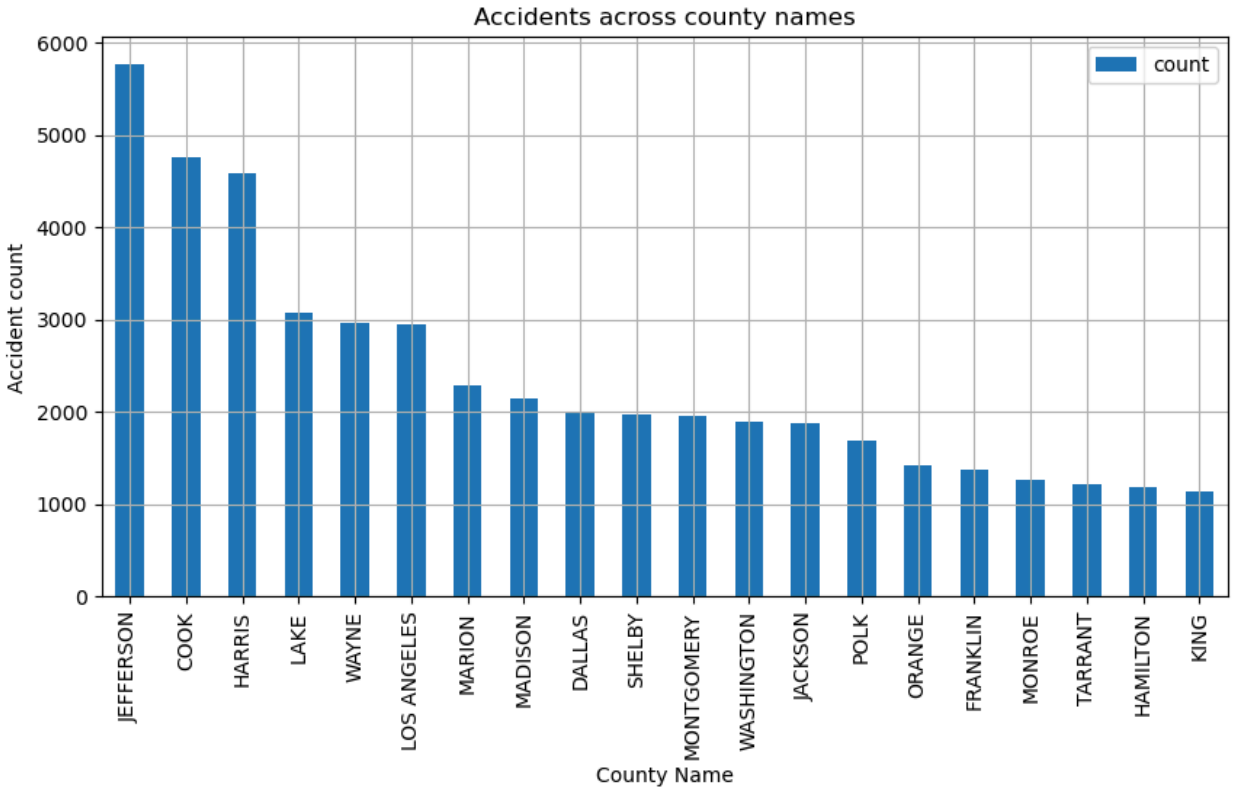
In the previous steps, we analyzed the accident rate in each state. Now, in the following step, we are focusing on finding the accident rate in different districts.



Observations:

- District 3: This district has the highest number of accidents, significantly higher than any other district.
- High Accident Districts: Districts 5 and 4 also have high accident counts, indicating they are major areas of concern after District 3.

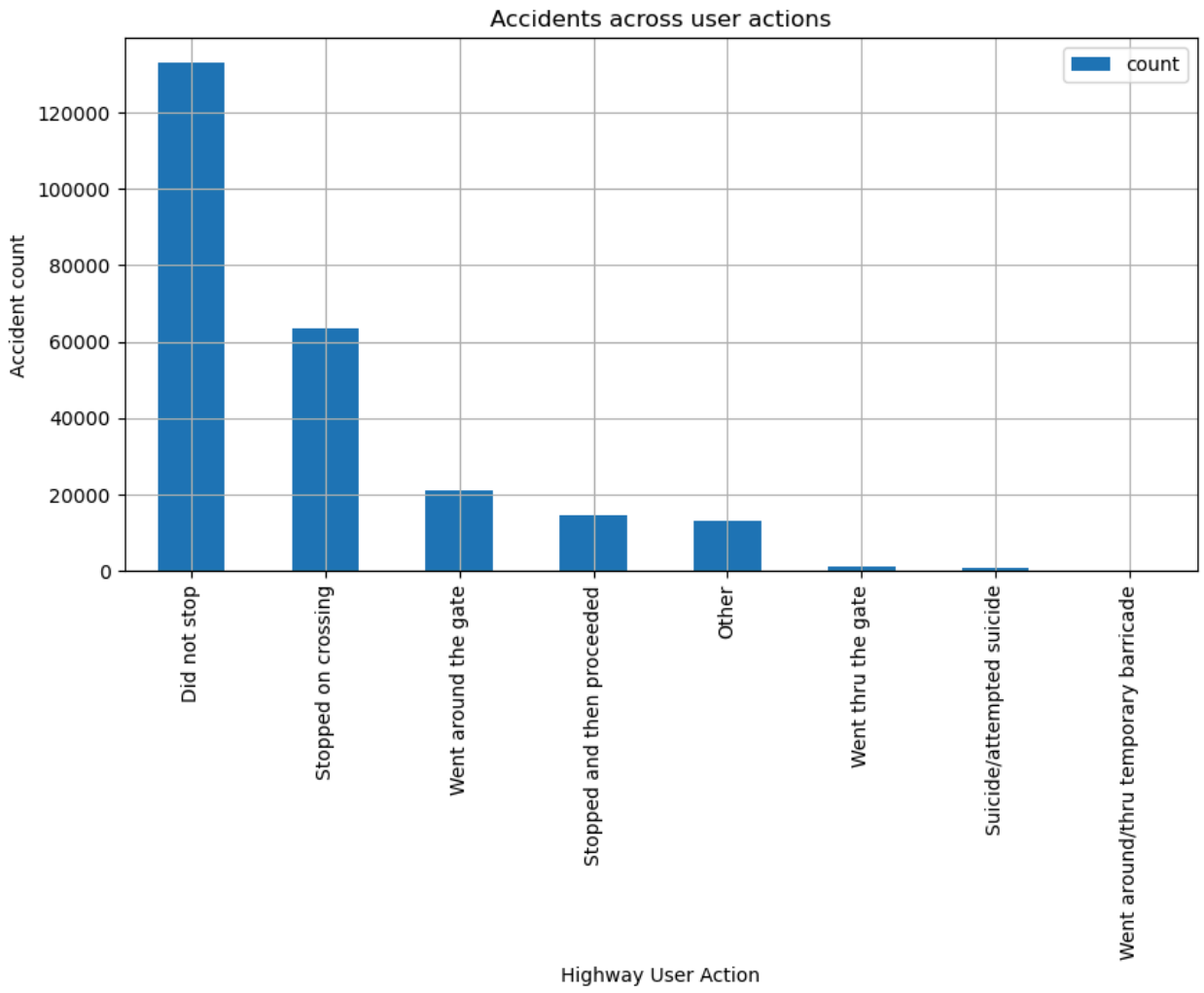
We continued our analyze by observing the relation of accident rate in each county.



Observations:

- **Jefferson County:** This county has the highest number of railroad crossing accidents, with nearly 6,000 incidents.
- **Top Counties:** Cook, Harris, and Lake counties also have high accident counts, all exceeding 4,000 incidents.

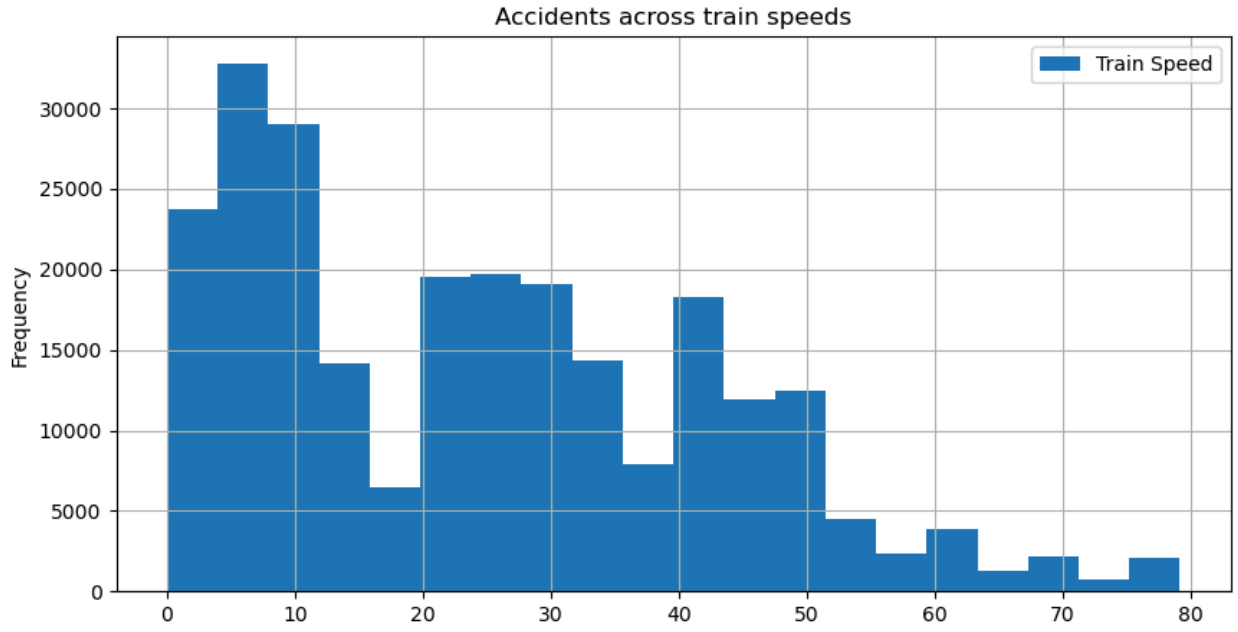
Since the actions of a car driver are crucial in the occurrence of accidents, we also analyzed them. In the following picture we can observe the result.



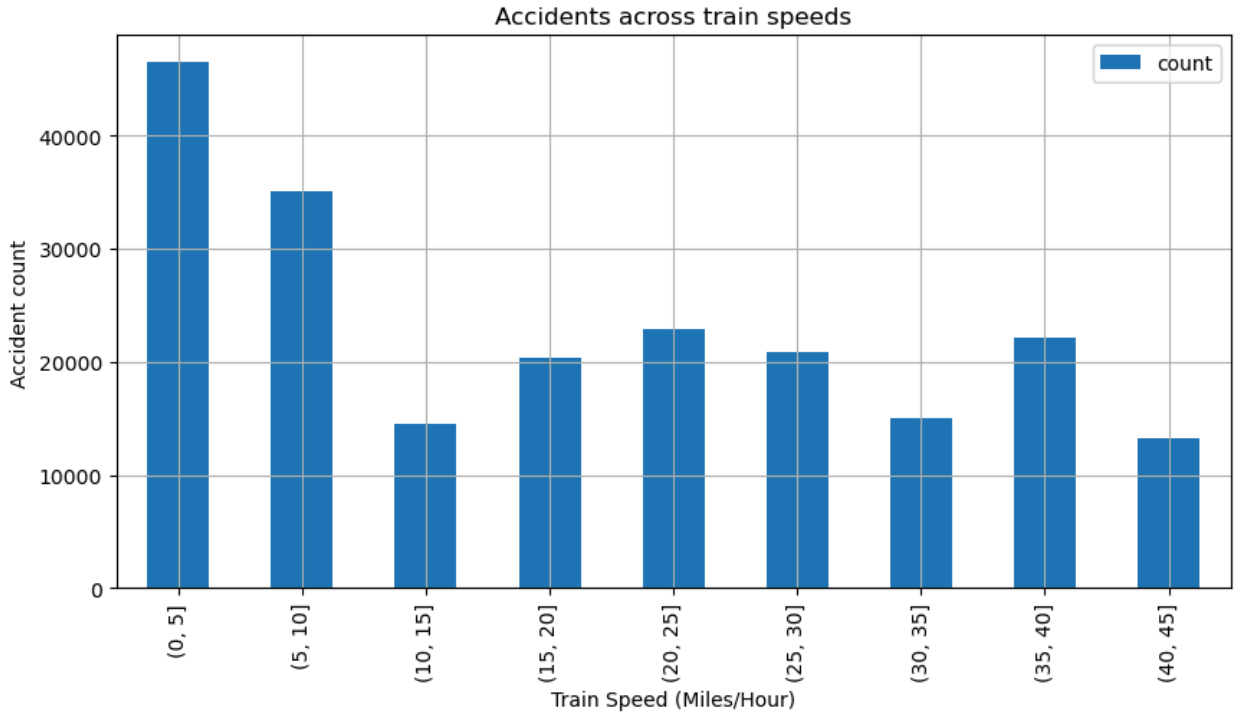
Observations:

- **"Did not stop":** The majority of accidents, over 120,000 incidents, are associated with highway users who did not stop at the crossing, hinting as the leading cause of accidents.

Another important variable worth analyzing is train speed. In the following image, we can observe the accident rate in relation to train speed.



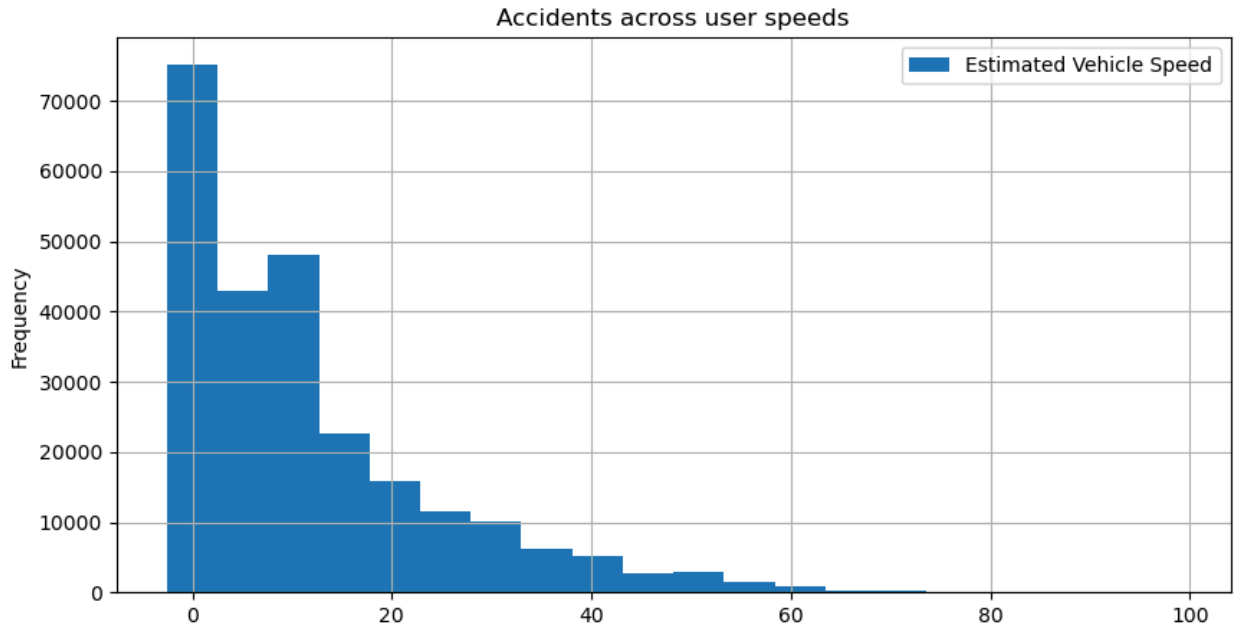
Like the previous step, we try to group speed and creating a categorized representation of train speed.



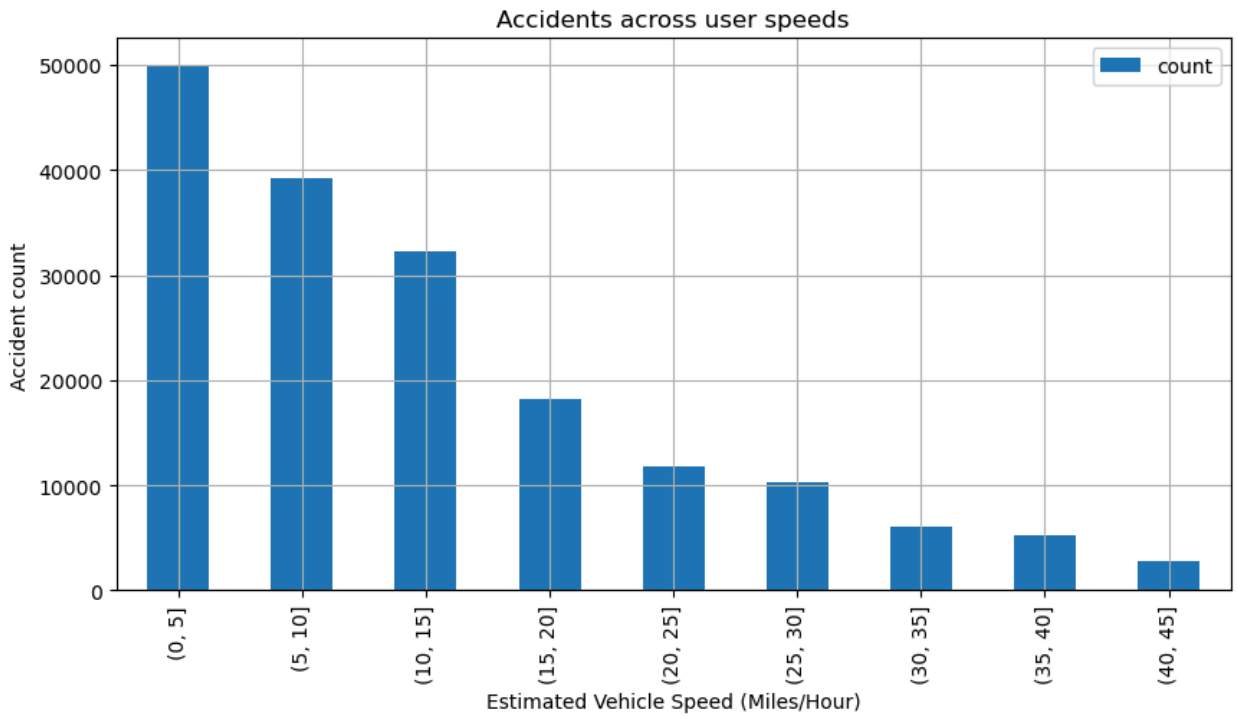
Observations:

- **Low-Speed Accidents:** The highest number of accidents occurs at train speeds between 0-5 miles per hour, with over 40,000 incidents. This suggests that slow-moving trains might be involved in more accidents, possibly due to increased train crossings or urban environments where slower speeds are common.
- **Decreasing Trend:** Accident counts generally decrease as train speeds increase, with the lowest number of accidents occurring in the 10-15 mph range.
- **Mid-to-High Speed Accidents:** As train speeds increase from 15-40 mph, there is a noticeable but smaller increase in accident counts, though these are still lower than the counts at very low speeds.
- **Outliers:** There seems to be some outliers in speeds above 80 mph.

We aply the same approach to the users' speed, as this is also an important factor to consider.



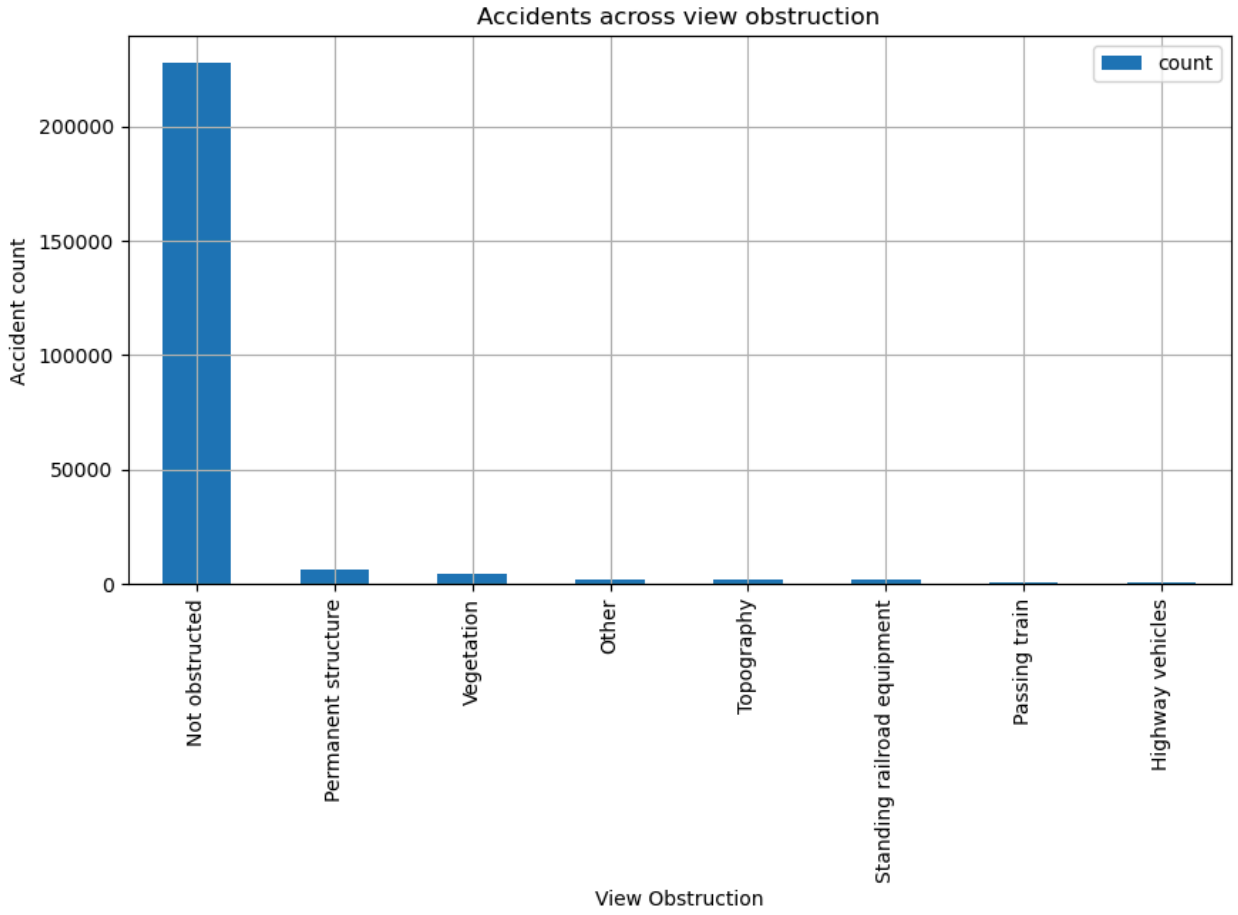
Same as previous step, we grouped each speed.



Observations:

- **Slow Speeds:** In most accidents the estimated vehicle speed was around 5 to 15 miles per hour.
- **Decreasing Trend:** There is a general decreasing trend in accident counts from low speeds to high speeds.
- **Outliers:** There seems to be some outliers in speeds above 70 mph.

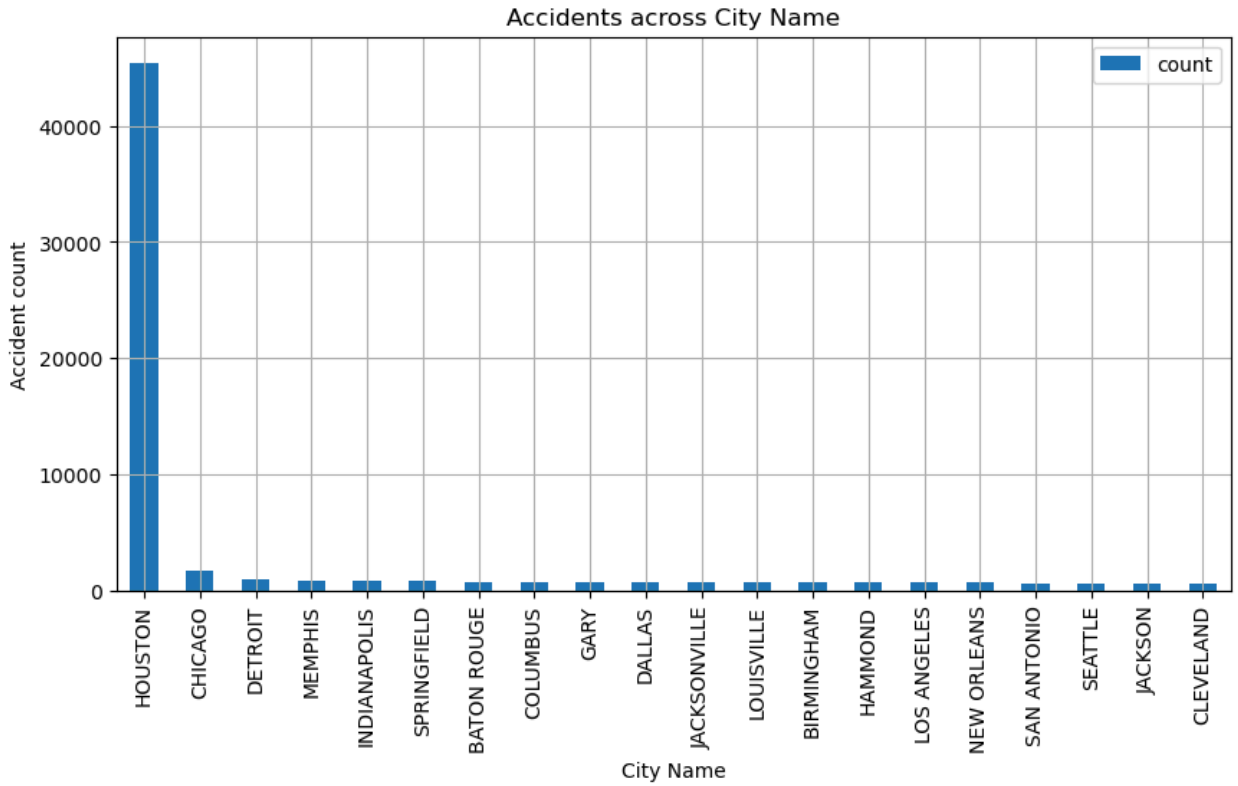
Obstacles are another important factor to consider. In the following picture we can observe their relation to the accident rate.



Observations:

- **Not Obstructed:** View obstruction is not the cause of most accidents, suggesting factors other than view are primary causes.

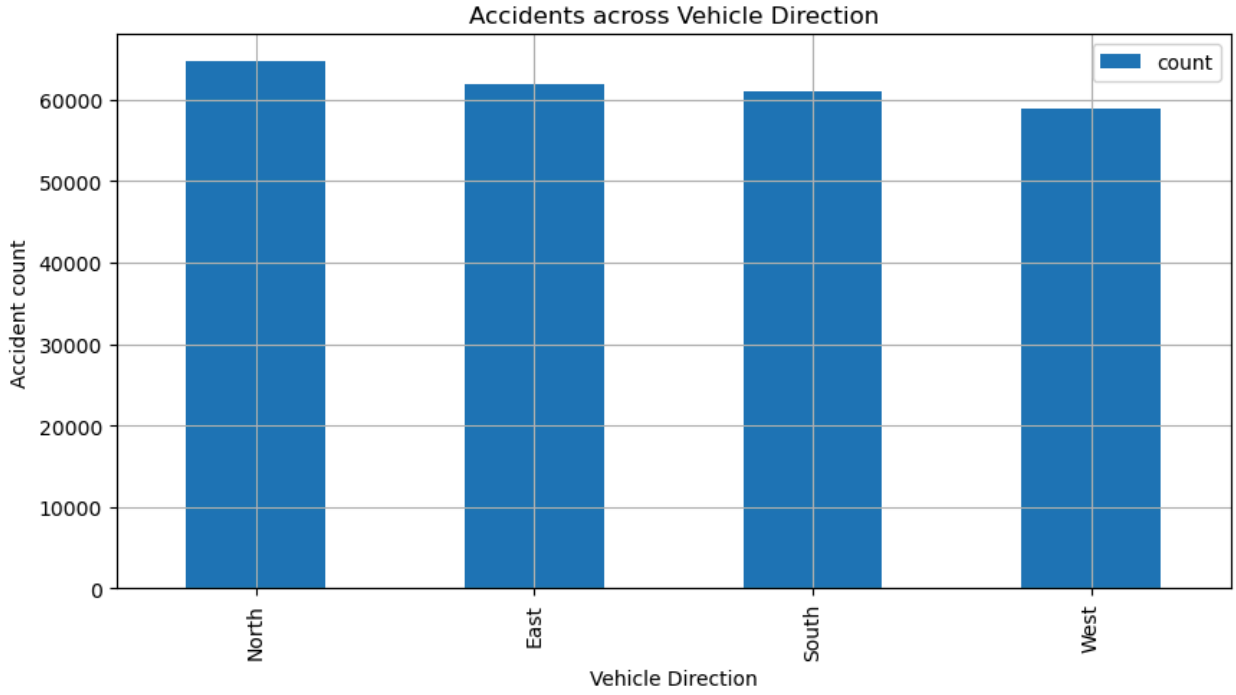
In the previous steps, we analyzed the accident rate in each state, now it is time to analyze it according to each city.



Observations:

- **Houston:** Houston has an exceptionally high number of railroad crossing accidents, far surpassing top 20 cities.

In the next step we tried to analyze the effect of vehicle direction on accident rate.

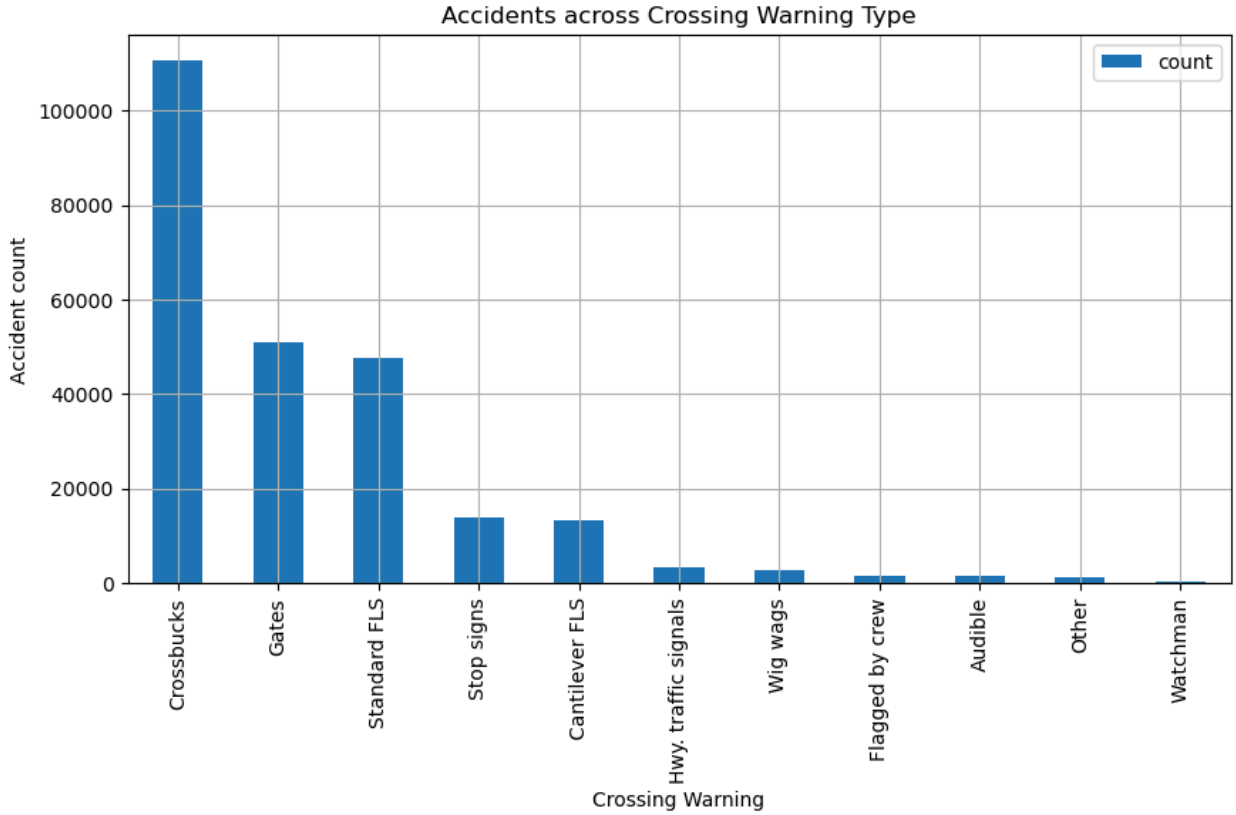


Observations:

- Uniform Distribution: Vehicle Direction is uniformly the same in all accidents.

The chart suggests that the direction of the vehicle does not play a significant role in accident occurrence, as the distribution of accidents is fairly even across all directions.

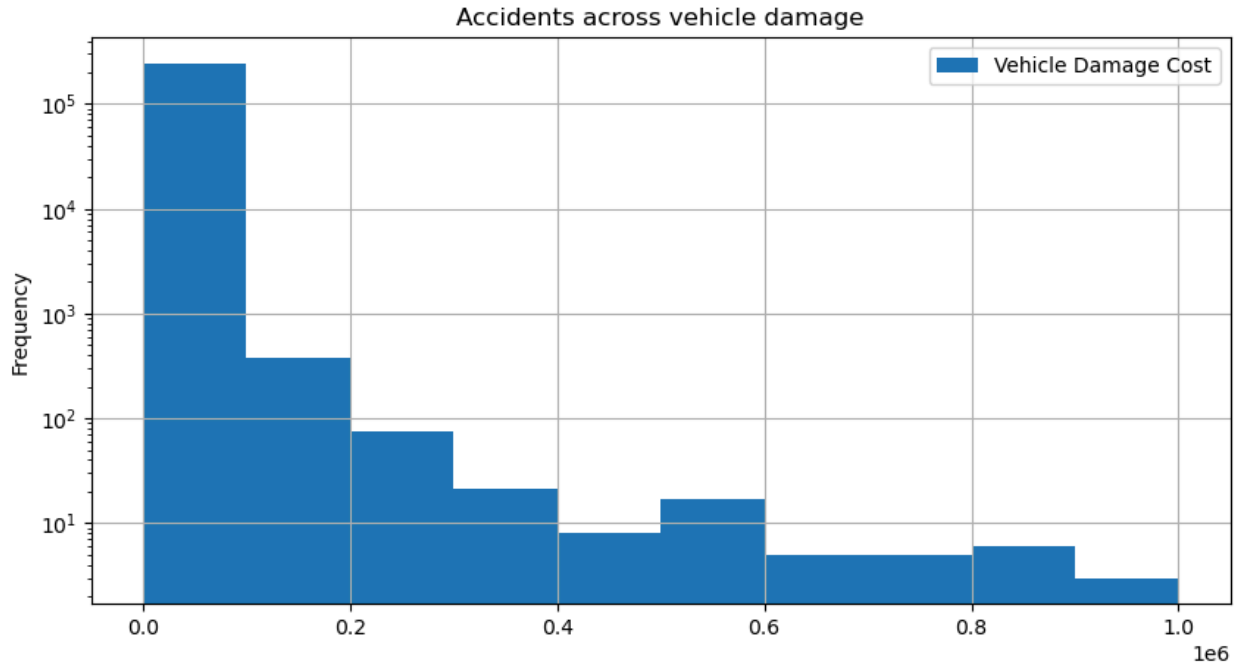
Another factor is the type of crossing warning and their effets in accident raate.



Observations:

- **Crossbucks:** The highest number of accidents occur at crossings with suggesting that this warning type may be less effective or more commonly used at high-risk crossings.
- **Gates and Standard Flashing Light Signals (FLS):** These warning types also account for a significant number of accidents indicating that even more advanced warning systems are associated with a substantial number of incidents.

Finally, we considered the vehicle damage cost. It is important to note that the x-axis is in logarithmic scale to provide a clearer understanding.



Observations:

- **Peak Damage Range:** The majority of accidents result in vehicle damage costs between $400 and $1200, with the highest frequency in the $400-$800 range.
- **Potential Inflation Impact:** The analysis of these damage costs should consider that these values might not account for inflation, meaning that the apparent concentration of costs in lower ranges could reflect older data or lower relative values rather than a true representation of accident severity.

3. Feature Engineering

In this section, we aim to select the most effective and important features for further analysis. We based our selection on observations from previous tasks, choosing features with strong correlations and excluding those with little or no relationship.

We select the following features:

- **selected_features** = ['Time of day', 'Weather Condition', 'Crossing Illuminated', 'Highway User', "Temperature', 'Visibility', 'Weather Condition', 'Number of Locomotive Units', 'Number of Cars', 'Train Speed', 'View Obstruction', 'Vehicle Damage Cost', 'Month', 'Season', 'Estimated Vehicle Speed', 'State Name', 'Highway User Action', 'Number Vehicle Occupants', 'District', 'Track Class', 'Driver Condition', 'Railroad Name', 'Public/Private', 'Crossing Warning Expanded 1', 'Estimated/Recorded Speed']

- **ordinal_features** = ['View Obstruction', 'District', 'Highway User', 'Track Class', 'Driver Condition', 'Railroad Name', 'Public/Private', 'State Name']

- **one_hot_features** = ['Time of day', 'Crossing Illuminated', 'Visibility', 'Weather Condition', 'Season', 'Highway User Action', 'Crossing Warning Expanded 1', 'Estimated/Recorded Speed']

Categorical Data Endocing

We follow soem steps to creat a new dataframe named “df_extracted”, which includes calculations like report age, total casualties, speed differences, weekend occurrence, weather impact score, and identification of peak traffic hours. These new features can be useful for further analysis or predictive modeling. The encoding is defined within the notebook, and the following explanation outlines its logic.

1. Feature Extraction and Calculation:

- **Report Age:** Calculates the age of the report by subtracting the year of the incident from the current year.
- **Total Damaged People:** Sums up the number of people killed or injured (both passengers and employees) in each incident.
- **Speed Difference:** Computes the difference between the estimated vehicle speed and the train speed.
- **Weekend:** Identifies whether the incident occurred on a weekend (Saturday or Sunday).

2. Weather Impact Score:

- **Temperature Score:** Normalizes the temperature data to a scale between 0 and 1.
- **Visibility Score:** Maps visibility conditions to predefined scores.
- **Weather Scores:** Maps various weather conditions to predefined scores.
- **Weather Impact Score:** Combines the temperature, visibility, and weather scores into a composite score using weighted averages.

3. Peak Traffic Hours:

- **Peak Traffic Hours:** Identifies whether the incident occurred during peak traffic hours (morning and evening rush hours).

4. Crossing Risk Score:

- **Crossing Warning Scores:** Assigns scores to different types of crossing warnings.
- **Illumination Scores:** Scores whether the crossing was illuminated.
- **View Obstruction Scores:** Assigns scores based on the type of view obstruction present.

- **Crossing Risk Score:** Combines the above scores into a composite risk score for crossings.

5. Severity Score:

- **Severity Score:** Calculates a composite score to represent the severity of the incident based on fatalities, injuries, and vehicle damage cost.

6. High-Risk Locations:

- **High Risk City/State:** Flags whether the incident occurred in a high-risk city or state based on the frequency of incidents.

7. Vehicle Involvement Score:

- **Position Scores:** Assigns scores to different positions/actions of highway users during the incident.
- **Vehicle Involvement Score:** Combines the number of vehicle occupants, vehicle damage cost, and position scores into a composite score.

Outlier Removal

In the previous task, we plotted the relationship between accident rates and various attributes. However, some distributions showed outliers, suggesting possible errors in data collection. In this step, we aimed to remove those outliers based on the insights we gathered.

1. Temperature Outliers:

- The code filters out any data where the temperature is unrealistically low (below -20) or excessively high (above 120). These thresholds are chosen to remove potential data entry errors or outliers that could skew the analysis.

2. Estimated Vehicle Speed Outliers:

- It removes records where the estimated vehicle speed is either negative or above 70. Negative speeds are impossible, and speeds above 70 are considered rare.

3. Train Speed Outliers:

- The code excludes rows where the train speed is less than 0 or greater than 80. Negative speeds are invalid, and speeds over 80 is outliers.

4. Vehicle Damage Cost Outliers:

- It filters out entries where the vehicle damage cost is either negative or exceeds 50,000. Negative costs are not possible, and very high costs is outliers.

4. Task Four - Predictive Modeling:

In this section we aim to predict Driver Condition and Total Damaged People Prediction. We use Classification and Regression respectively.

Driver Condition Prediction

Feature Scaling:

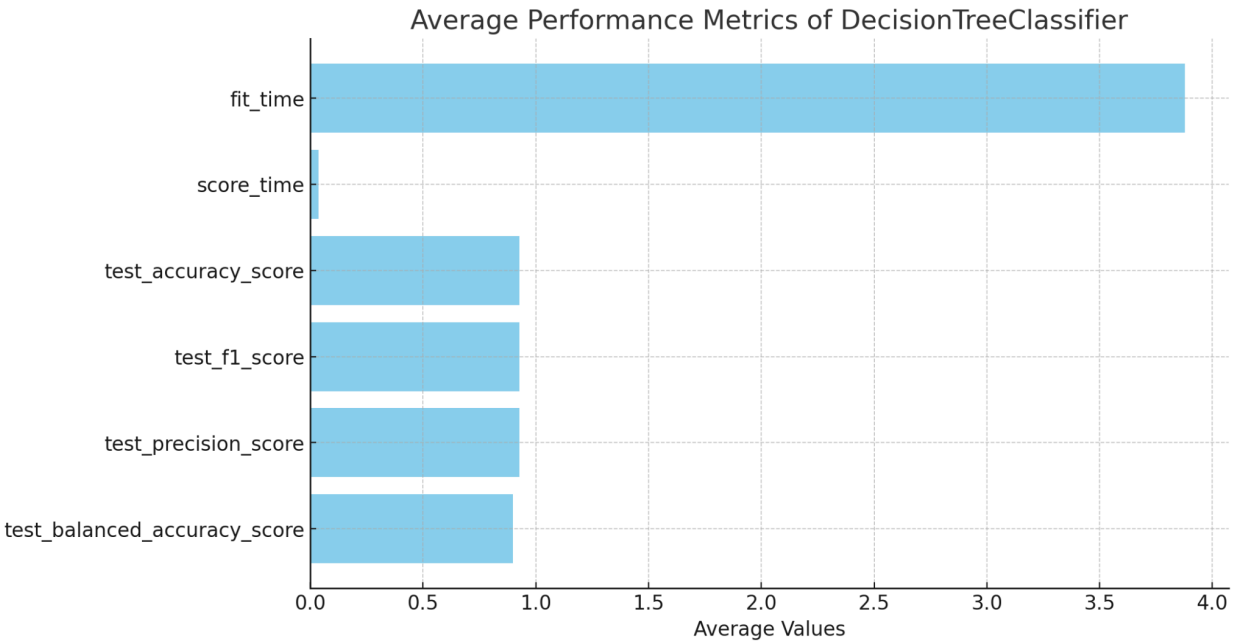
- The target variable for prediction is defined as Driver Condition and features are all other columns in the dataset df_predictive except for the prediction_column.
- To better train the model we tried to normalize the variables. So, we use Min-Max Scaler for Temperature', 'Train Speed', 'Estimated Vehicle Speed', 'Severity Score', and 'Vehicle Involvement Score'.
- We use Z-Score Scaling for Features like 'Number of Locomotive Units', 'Number of Cars', 'Speed Difference', and 'Total Damaged People'.
- It is worth mentioning that The Vehicle Damage Cost feature is transformed using a logarithmic scale to handle skewness in the data.
- We also use cross validation technique with the parameter cv = 5.

Model Selection:

We tried to test different classification models and compare their performance.

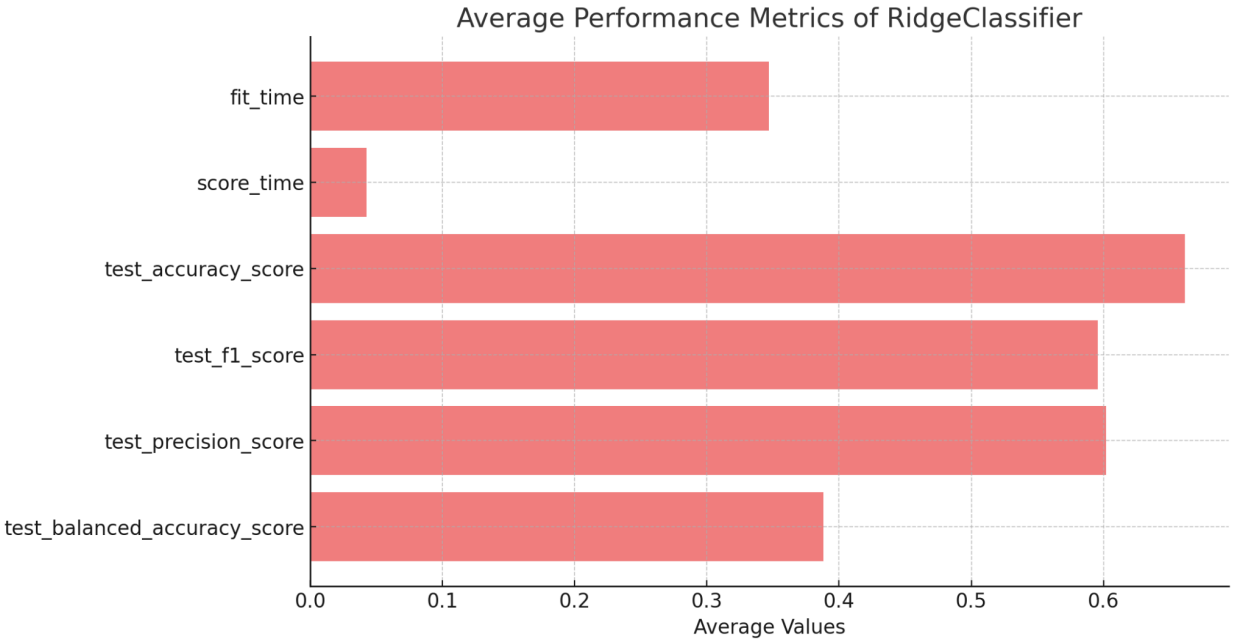
1. DecisionTreeClassifier
2. RidgeClassifier
3. RandomForestClassifier
4. XGBClassifier
5. KNeighborsClassifier
6. BernoulliNB

1. DecisionTree



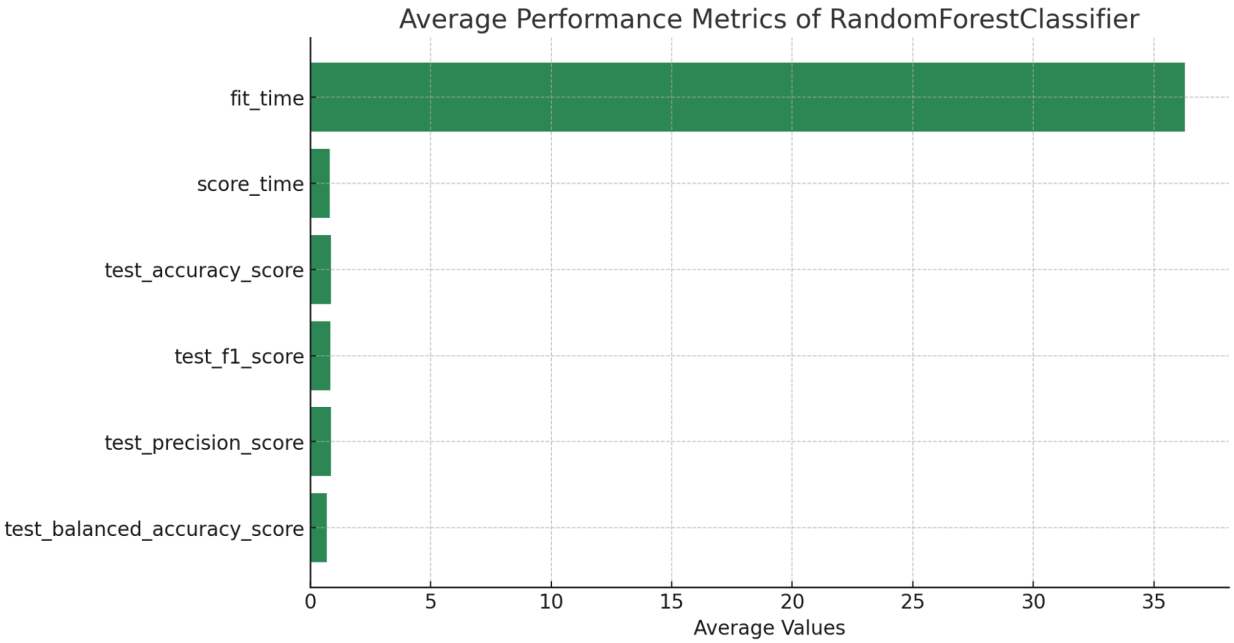
METRIC	AVE	STD	MIN	MAX
fit_time	3.8784	0.3095	3.5743	4.2668
score_time	0.0376	0.0116	0.0310	0.0584
test_accuracy_score	0.9265	0.0137	0.9045	0.9424
test_f1_score	0.9266	0.0136	0.9047	0.9424
test_precision_score	0.9267	0.0135	0.9051	0.9424
test_balanced_accuracy_score	0.8979	0.0164	0.8753	0.9169

2. RidgeClassifier



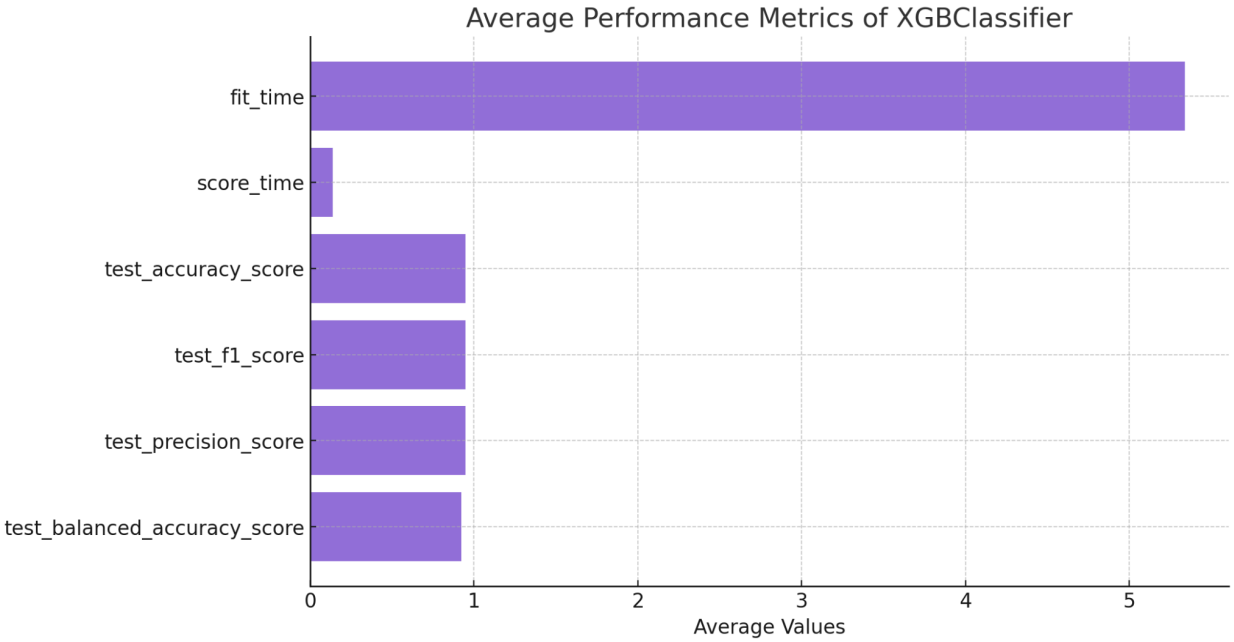
METRIC	AVE	STD	MIN	MAX
fit_time	0.3471	0.0728	0.2666	0.4596
score_time	0.0424	0.0086	0.0331	0.0519
test_accuracy_score	0.6616	0.0023	0.6585	0.6645
test_f1_score	0.5957	0.0095	0.5844	0.6058
test_precision_score	0.6021	0.0136	0.5875	0.6192
test_balanced_accuracy_score	0.3883	0.0079	0.3792	0.3973

3. RandomForestClassifier



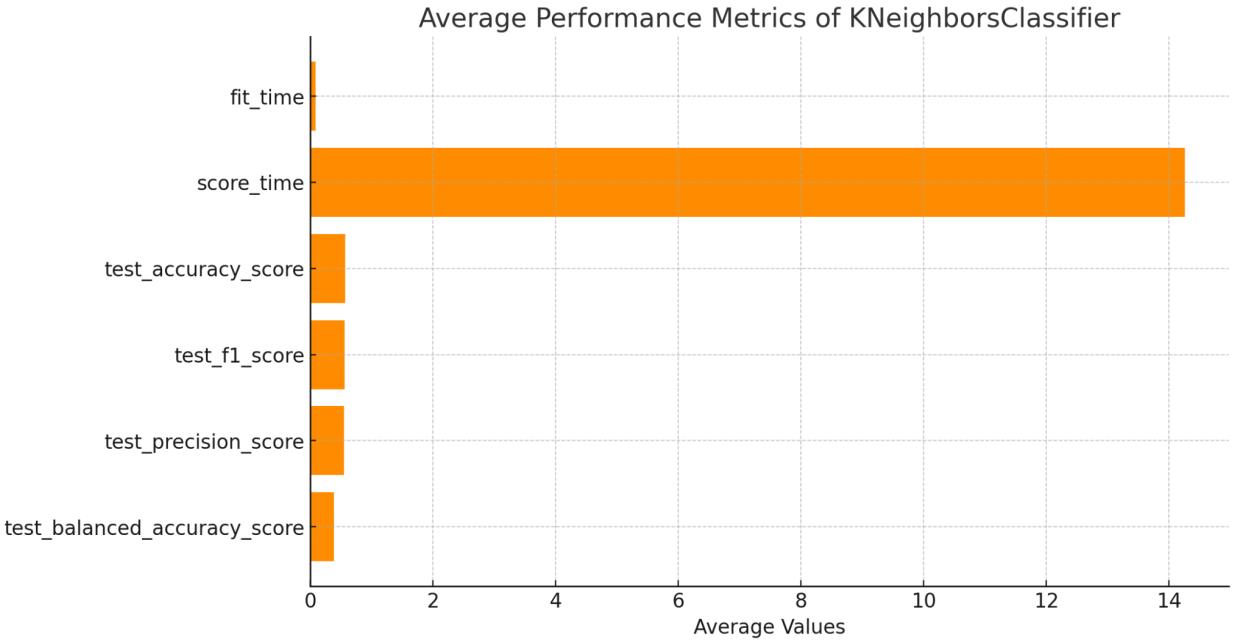
METRIC	AVE	STD	MIN	MAX
fit_time	36.2926	1.5109	34.87	38.5
score_time	0.8174	0.0511	0.78	0.90
test_accuracy_score	0.8508	0.0114	0.84	0.87
test_f1_score	0.8429	0.0103	0.83	0.86
test_precision_score	0.8518	0.0123	0.84	0.87
test_balanced_accuracy_score	0.6911	0.0227	0.66	0.72

4. XGBClassifier



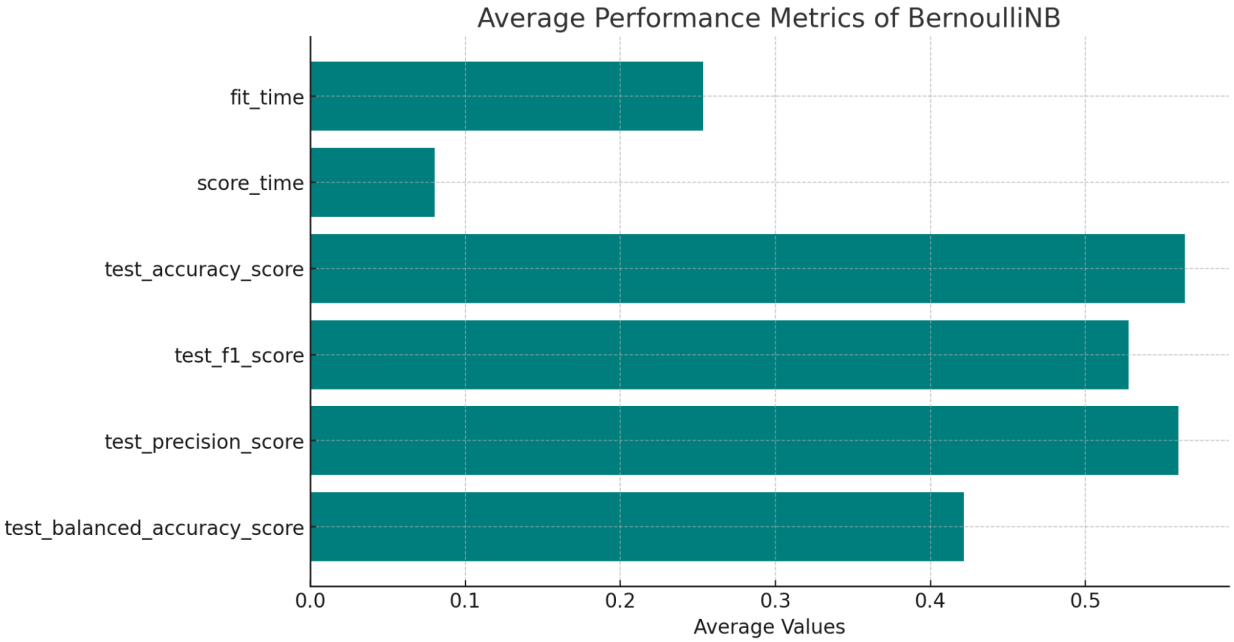
METRIC	AVE	STD	MIN	MAX
fit_time	5.3401	0.1295	5.2157	5.5324
score_time	0.1370	0.0077	0.1305	0.1497
test_accuracy_score	0.9483	0.0056	0.9410	0.9548
test_f1_score	0.9484	0.0056	0.9408	0.9547
test_precision_score	0.9491	0.0055	0.9412	0.9550
test_balanced_accuracy_score	0.9218	0.0143	0.9000	0.9347

5. KNeighborsClassifier



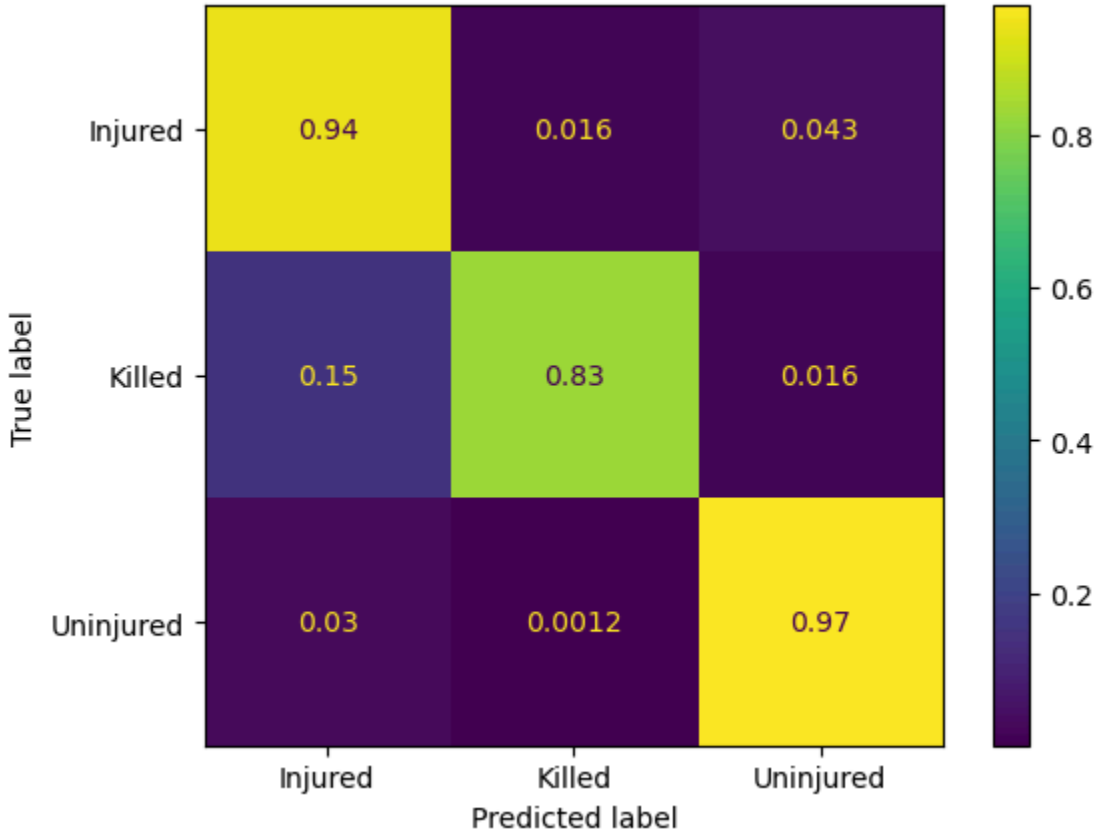
METRIC	AVE	STD	MIN	MAX
fit_time	0.0915	0.0095	0.0794	0.10
score_time	14.2595	0.2583	13.9133	14.49
test_accuracy_score	0.5739	0.0363	0.5154	0.60
test_f1_score	0.5576	0.0176	0.5286	0.57
test_precision_score	0.5503	0.0047	0.5457	0.55
test_balanced_accuracy_score	0.3841	0.0036	0.3788	0.38

6. BernoulliNB



METRIC	AVE	STD	MIN	MAX
fit_time	0.2533	0.0184	0.2334	0.27
score_time	0.0802	0.0128	0.0665	0.09
test_accuracy_score	0.5642	0.0191	0.5412	0.58
test_f1_score	0.5279	0.0132	0.5111	0.54
test_precision_score	0.5598	0.0043	0.5535	0.56
test_balanced_accuracy_score	0.4216	0.0114	0.4104	0.44

As the XGBclassifier achieve the best performance we observed it in details, so We also plot the confusion matrix for XGBclassifier.



CLASS	PRECISION	RECALL	F1-SCORE	SUPPORT
0	0.90	0.94	0.92	12571
1	0.92	0.83	0.87	3333
2	0.98	0.97	0.97	27735
Accuracy			0.95	43639
Macro Avg	0.93	0.91	0.92	43639
Weighted Avg	0.95	0.95	0.95	43639

Model Comparison

Best Performing Model: XGBClassifier - It consistently has the highest values across accuracy, F1-score, precision, and balanced accuracy, making it the most robust classifier in this comparison.

Worst Performing Model: BernoulliNB - It generally has the lowest values for accuracy, F1-score, and balanced accuracy, indicating it is the least effective among the models tested.

Best Time Efficiency: KNeighborsClassifier for fit time and RidgeClassifier for score time, but these do not necessarily correlate with better predictive performance.

Total Damaged People Prediction

In this part we aim to predict the total damaged people using different types of regression.

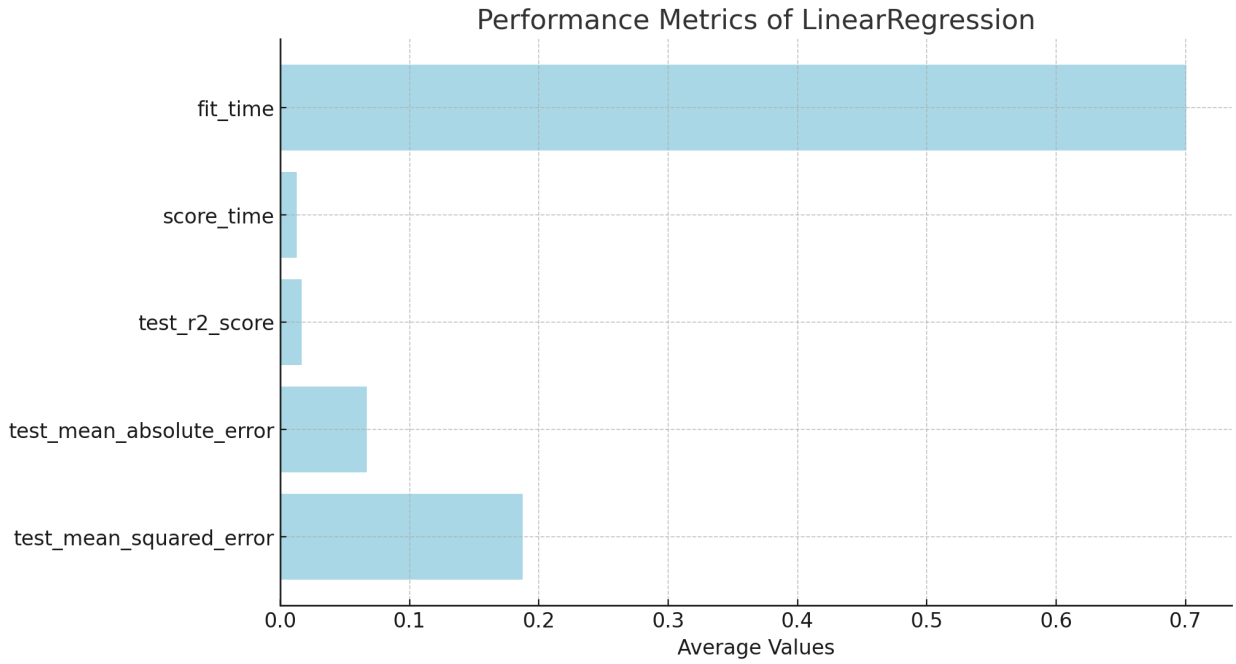
Model selection:

1. LinearRegression
2. Ridge
3. ElasticNet
4. BayesianRidge
5. XGBRegressor
6. DecisionTreeRegressor

We also use 5-fold cross validation to assess the model performance, ensuring that the evaluation is robust and less prone to overfitting.

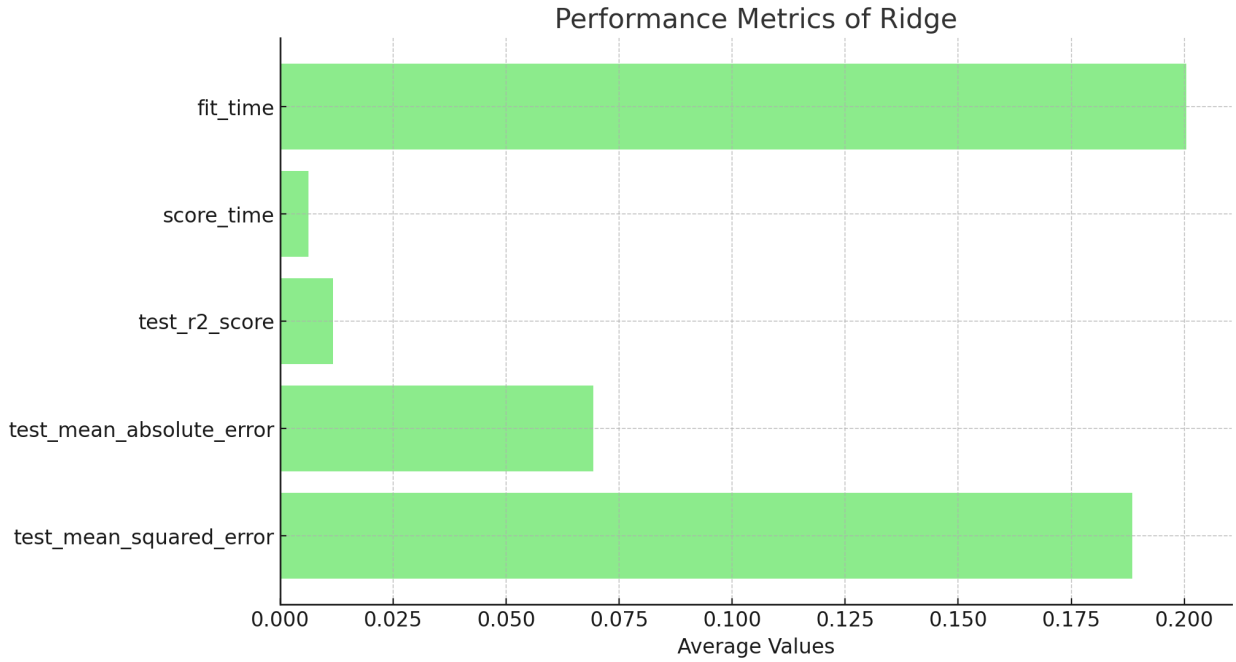
We began our analysis by LInear Regrassion, which is always first choice in regression problems.

1. LinearRegression



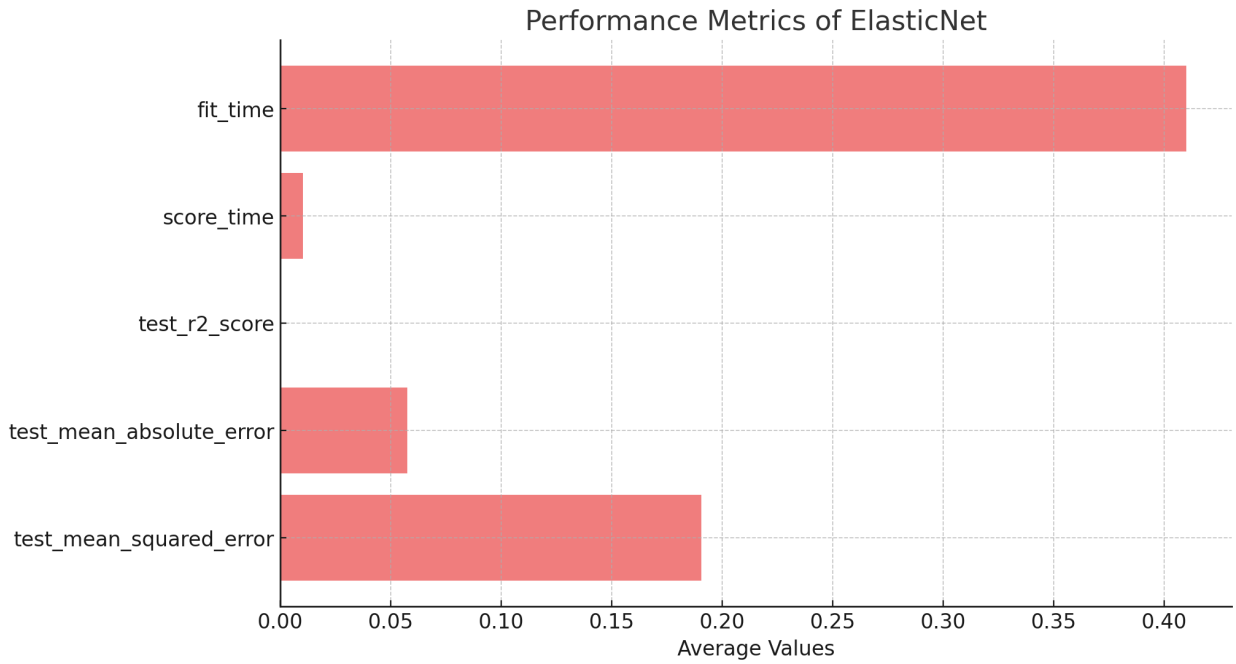
METRIC	AVE	STD	MIN	MAX
fit_time	0.7008	0.0631	0.5990	0.7497
score_time	0.0128	0.0076	0.0000	0.0193
test_r2_score	0.0167	0.0044	0.0110	0.0229
test_mean_absolute_error	0.0668	0.0062	0.0594	0.0751
test_mean_squared_error	0.1875	0.0977	0.1037	0.3380

2. Ridge



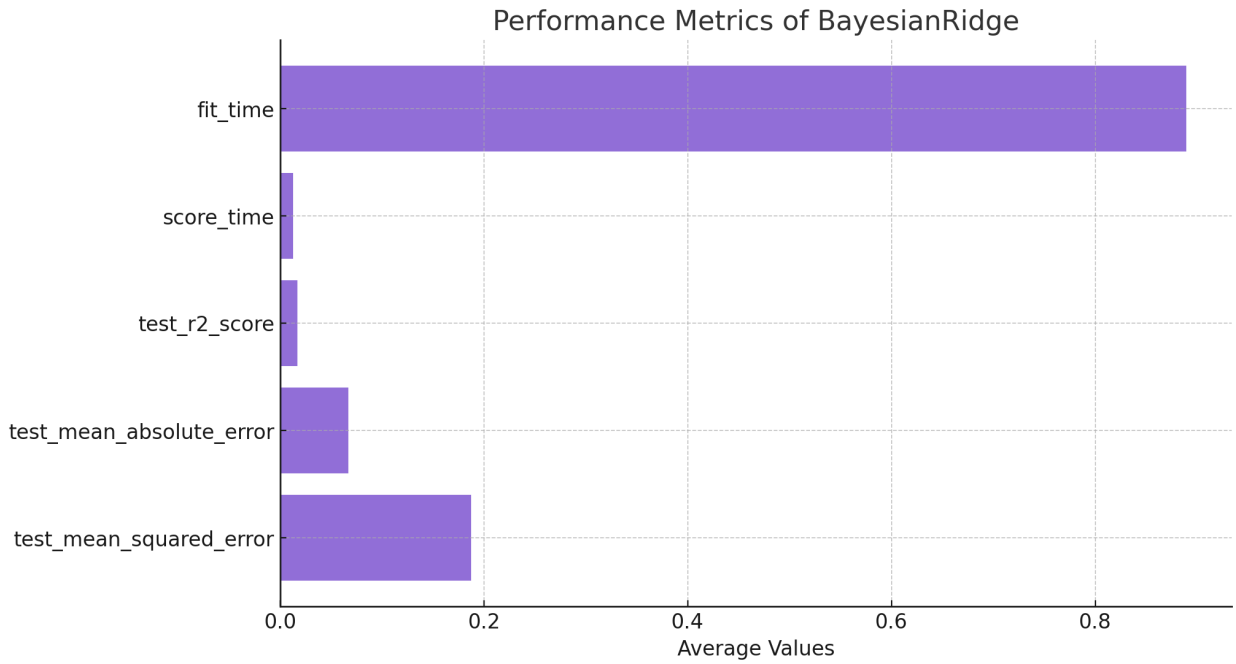
METRIC	AVE	STD	MIN	MAX
fit_time	0.2005	0.0095	0.1863	0.2131
score_time	0.0063	0.0086	0.0000	0.0159
test_r2_score	0.0116	0.0038	0.0072	0.0175
test_mean_absolute_error	0.0692	0.0056	0.0629	0.0766
test_mean_squared_error	0.1885	0.0980	0.1043	0.3393

3. ElasticNet



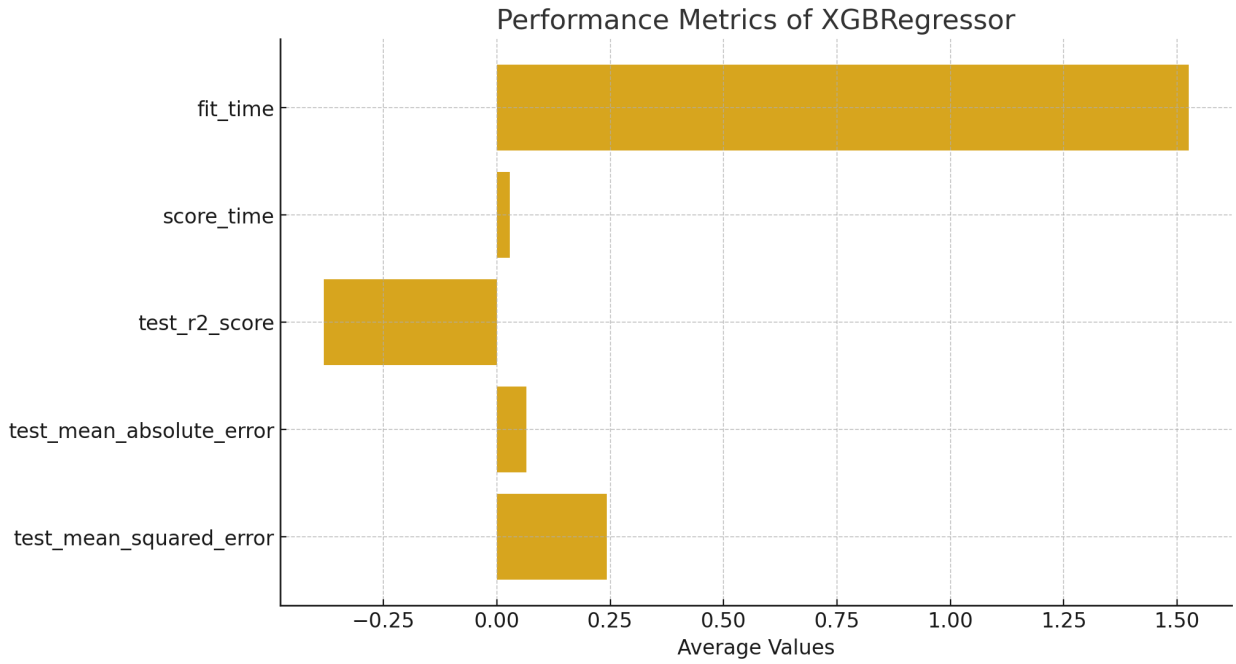
METRIC	AVE	STD	MIN	MAX
fit_time	0.4101	0.0500	0.3653	0.4864
score_time	0.0103	0.0095	0.0000	0.0200
test_r2_score	-0.0000	0.0001	-0.0001	0.0001
test_mean_absolute_error	0.0576	0.0024	0.0551	0.0605
test_mean_squared_error	0.1905	0.0986	0.1055	0.3418

4. BayesianRidge



METRIC	AVE	STD	MIN	MAX
fit_time	0.8896	0.1207	0.7660	1.0641
score_time	0.0126	0.0057	0.0030	0.0169
test_r2_score	0.0167	0.0044	0.0110	0.0230
test_mean_absolute_error	0.0668	0.0062	0.0595	0.0751
test_mean_squared_error	0.1875	0.0977	0.1037	0.3381

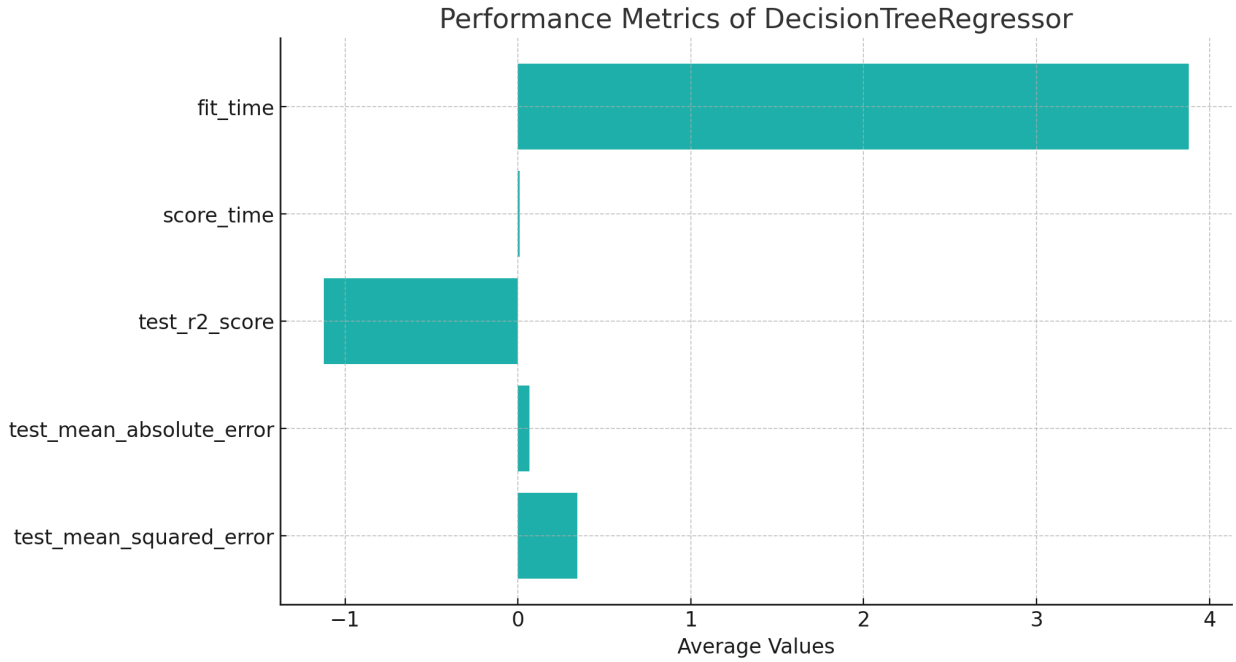
5. XGBRegressor



The negative R^2 score indicates that the model performs worse than a horizontal line (mean predictor), which suggests that this model might not be well-suited for this task.

METRIC	AVE	STD	MIN	MAX
fit_time	1.5252	0.0788	1.4461	1.6575
score_time	0.0293	0.0044	0.0213	0.0312
test_r2_score	-0.3821	0.4871	-1.2448	-0.0967
test_mean_absolute_error	0.0649	0.0075	0.0549	0.0728
test_mean_squared_error	0.2430	0.0917	0.1294	0.3759

6. DecisionTreeRegressor



METRIC	AVE	STD	MIN	MAX
fit_time	3.8788	0.0694	3.7854	3.9806
score_time	0.0106	0.0100	0.0000	0.0216
test_r2_score	-1.1241	1.6871	-4.1298	-0.2303
test_mean_absolute_error	0.0653	0.0081	0.0565	0.0756
test_mean_squared_error	0.3435	0.1805	0.1717	0.5944

Model Comparison

Best Performing Model: LinearRegression and BayesianRidge (Tie) - These models show the best performance in terms of R^2 score and mean squared error, which indicates they are better at explaining the variance in the data.

Worst Performing Model: DecisionTreeRegressor - This model has a negative R^2 score and the highest mean squared error, indicating poor predictive performance.

Best Time Efficiency: Ridge for both fit and score time, making it the most time-efficient model, though not the most accurate.