Report on US Accidents From 2016 to 2021

Group Number: 12

Group Member

Shuhan Qian	sq2235
Xinghan Gong	xg2356
Xinran Wang	xw2809
Xubo Wang	xw2808
Yizhou Zhao	yz3989
Zixue Wang	zw2767

Columbia University
Department of Statistics
Course 5291
Spring 2022

Content

1. Introduction

1.1 General Data Description

Traffic has always been an important part of a city. Factors such as weather and intersection conditions can lead to traffic accidents and thus affect people's travel. In this paper, we want to use past data to filter out the most important factors and build a suitable model to predict future accidents.

In this project, the US Accidents (2016 - 2021) dataset is used to analyze different factors such as visibility and temperature that can contribute to the traffic accident's severity. This dataset contains detailed information regarding traffic accidents that covers 49 states in the United States. There are around 2.8 million observations and 47 columns in total. The independent variable is a categorical variable reflecting the severity of the accidents at a level from 1 to 4, where 1 represents a low impact on traffic and 4 indicates a significant impact. The results can be referred to the government to make predictions and conduct traffic surveillance. As shown below, Table 1.1 gives descriptions of some representative attributes.

Severity	Shows the severity of the accident, a number between 1 and 4, where 1 indicates the least impact on traffic and 4 indicates a significant impact on traffic.
Distance(mi)	The length of the road affected by the accident.
Bump	A POI annotation indicates the presence of a speed bump or hump in a nearby location.
Start_Time	Shows the start time of the accident in the local time zone.
End_Time	Shows the end time of the accident in the local time zone. End time here refers to when the impact of an accident on traffic flow was dismissed.
Roundabout	A POI annotation indicates the presence of a roundabout in a nearby location.
Pressure(in)	Shows the air pressure (in inches).
Side	Shows the relative side of the street (Right/Left) in the address field.

Sunrise_Sunset	Shows the period of day (i.e. day or night) based on sunrise/sunset.
Start_Lat	Shows latitude in the GPS coordinate of the start point.
Start_Lng	Shows longitude in the GPS coordinate of the start point.

Table 1.1: Descriptions of Attributes

1.2 Data Source

Since February 2016, data has been regularly collected utilizing a variety of data providers, including various APIs that give streaming traffic event data. These APIs transmit traffic events gathered by a range of institutions inside the road networks, including the US and state departments of transportation, law enforcement agencies, traffic cameras, and traffic sensors [1].

2. Data Preprocessing

In this section, we process the data to make the dataset usable for models that will be implemented. We perform the data processing based on three standards, the number of missing values or errors, multicollinearity between variables, and the amount of information the variable can provide to predict the severity of the accidents.

By checking the missing values (shown in Figure 2.1), we found that the variable 'Number'(stands for specific street number) contains around 1.7 million missing values, which accounts for more than 60% of the row records. Hence, we decided to remove this column. Similarly, we also remove the 'Wind_Chill' column because it contains too many missing values and it is directly proportional to temperature(shown in Figure 2.2). 'Precipitation(in)' also contains many missing values, but nevertheless, we decided to keep it because of particular interest.

```
ID                        0
Severity                  0
Start_Time                0
End_Time                  0
Start_Lat                 0
Start_Lng                 0
End_Lat                   0
End_Lng                   0
Distance(mi)              0
Description               0
Number              1743911
Street                    2
Side                      0
City                    137
County                    0
State                     0
Zipcode                1319
Country                   0
Timezone               3659
Airport_Code           9549
Weather_Timestamp     50736
Temperature(F)        69274
Wind_Chill(F)        469643
Humidity(%)           73092
Pressure(in)          59200
Visibility(mi)        70546
Wind_Direction        73775
Wind_Speed(mph)      157944
Precipitation(in)    549458
Weather_Condition     70636
Amenity                   0
Bump                      0
Crossing                  0
Give_Way                  0
Junction                  0
No_Exit                   0
Railway                   0
Roundabout                0
Station                   0
Stop                      0
Traffic_Calming           0
Traffic_Signal            0
Turning_Loop              0
Sunrise_Sunset         2867
Civil_Twilight         2867
Nautical_Twilight      2867
Astronomical_Twilight  2867
dtype: int64
```

Figure 2.1: The Number of Null Observations for Each Variable.

From Figure 2.1, we can see there are many variables with around 70k missing values. Since we have around 2.8 million observations in total, this is relatively small. Thus, we decided to keep them for this step.

Collinearity between variables is presented in the correlation matrix shown as a heatmap in Figure 2.2. From the heatmap, we can see several pairs of variables are highly correlated, 'Wind_Chill' and 'Temperature', 'Start_Lat' and 'End_Lat', 'Start_Lng' and 'End_Lng', and 'Traffic_Calming' and 'Bump'. For these pairs of variables, we only keep one of them. 'Turning_Loop' is displayed as an invalid coefficient. This is because it is always 'False', which does not provide any information.

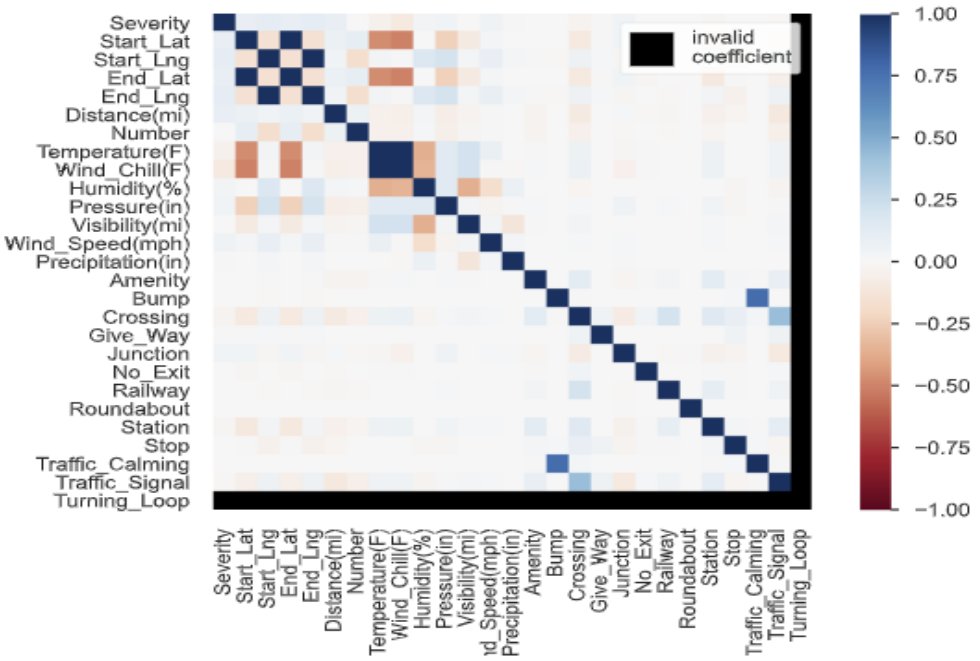

Figure 2.2: Heatmap of Pearson Correlation between the Variables

Moreover, we also dropped the variables that cannot provide much information about the severity of the accidents such as 'Zipcode', 'Airport_code', etc. In addition, we discovered that the variable 'Weather_Condition' has a very different structure. It has mixed values such as 'Light Snow / Windy', 'Snow and Thunder / Windy', 'Fog / Windy', 'Heavy Thunderstorms with Small Hail', 'Light Rain Showers', 'Light Rain / Windy', etc. These values are more descriptive than quantitative, and some of the descriptions even used different wording to convey the same meaning. There are in total 128 unique descriptions of weather conditions.

Nevertheless, the other way to represent a general weather condition is to use the numerically variables like temperature, wind_speed, humidity, etc. Thus, the variable ‘Weather_Condition’ is also being dropped.

Therefore, initially we decided to drop the following variables:

- Number: too many missing values and provide no information for the severity
- Wind_Chill: too many missing values and highly correlated with temperature
- Traffic_calming: highly correlated with Bump
- End_Lat and End_Lng: highly correlated with Start_Lat and Start_Lng
- ID: do not carry any information for the severity
- Start_Time: decomposed to year, month and day
- End_time: unnecessary variable, cannot cause effect on severity
- Description: most of the descriptions report only the location like road or exit of the accident
- Zipcode, Airport_Code, Amenity, Timezone, Weather_Timestamp: because they do not provide much useful information for our task
- Street, Side, City, County, Country: because we are more interested in the states where the accident happened
- Nautical_Twilight, Astronomical_Twilight, Civil_Twilight: because they are just different ways to describe day and night, so we decided just keep Sunrise_Sunset is enough
- Weather_condition: descriptions too complicated and can be represented by other numerical variables
- Distance: its definition is generally the same as our target, severity

Then for the categorical variables such as ‘Bump’ and ‘Crossing’ containing only values ‘True’ and ‘False’, we encode 1 = True and 0 = False.

Finally, the variables we left with are:

Severity, Temperature(F), Humidity(%), Pressure(in), Visibility(mi), Wind_Speed(mph), Precipitation(in), Bump, Crossing, Give_Way, Junction, No_Exit, Railway, Roundabout, Station, Stop, Traffic_Calming, Traffic_Signal, Sunrise_Sunset

3. Exploratory Data Analysis

3.1 Basic Analysis

3.1.1 Imbalanced Data

After basic preprocessing, we finally have a 'clean' dataset to analyze. Since it's a classification problem and we are interested in predicting severity, our first focus is the distribution of the categorical response variable 'Severity'.

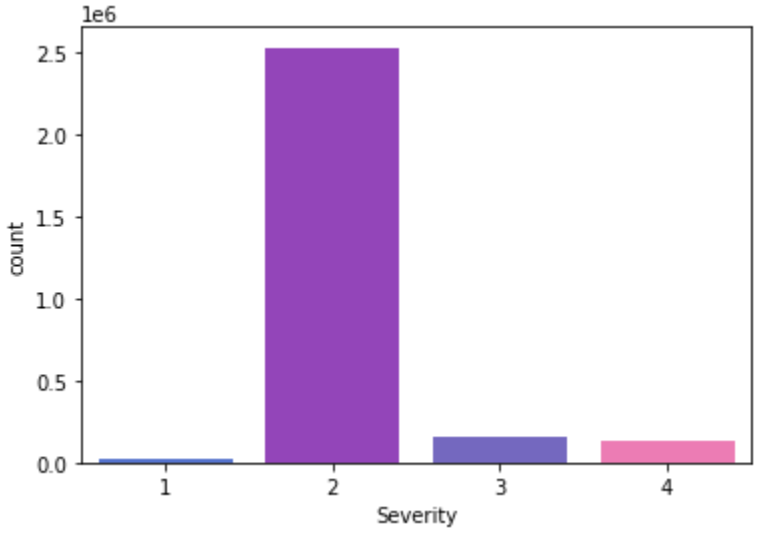

Figure 3.1: Number of Accidents under Each Severity Level

From Figure 3.1, it's noteworthy that more than 2 million out of 2.24 million accidents have a severity of level 2. Nearly 90% of data falling into a single category shows a strong signal of imbalanced data. The techniques applied to handle the problems are Undersampling, Oversampling, and SMOTE.

Undersampling is to randomly delete rows from the majority class ('Severity' 2) to match them with the minority classes ('Severity' 1, 3, 4). By doing so, we can get a balanced dataset for the majority and minority classes. Similarly, oversampling is to upsample the minority classes ('Severity 1, 3, 4") to match them with the majority classes. When all classes have a similar number of records present in the dataset, we can assume that the classifier will give equal importance to both classes.

```
data_sample_1['Severity'].value_counts()
```

```
1    23556
2    23556
3    23556
4    23556
```

```
data_sample_2['Severity'].value_counts()
```

```
1    2080349
2    2080349
3    2080349
4    2080349
```

Synthetic Minority Oversampling Technique or SMOTE is another technique to oversample the minority classes ('Severity' 1, 3, 4). Rather than simply adding duplicate records of minority classes, since it doesn't add any new information to the model, SMOTE synthesized new instances from the existing data. In other words, SMOTE looks into minority class instances and uses k nearest neighbor to select a random nearest neighbor, and a synthetic instance is created randomly in feature space.

In this report, we decide to use Undersampling to balance the data to avoid a heavy workload. Both oversampling and SMOTE produce a dataset of more than 8 million observations which is difficult to program and analyze. In contrast, an undersampled dataset has only 94224 observations.

Important Note: reworking the dataset is not always a solution in terms of improving model prediction accuracy. It's important to understand why we treat imbalanced data as a problem. It's because we assume the real-world data are balanced, which is a proportions bias since it's not always the case. In our study, nearly 90% of reported accidents have level 2 Severity, and this might be because most car accidents are neither too trivial nor too severe. Most accidents have a normal 'Severity' level. In conclusion, understanding the importance or weights of every class is critical in building a decent classifier in a classification problem.

3.1.2 Accident Count by Year

To explore more details of our imbalanced dataset, we decided to use the histogram to see the trend of accidents in Figure 3.2. It gives us information that the number of accidents has been increasing over the past six years. Also, 2021 experienced the most traffic accidents, which is almost twice as many as in 2020. Since accidents are labeled as '2' and account for the majority in all the six years, we have to deal with the imbalanced dataset before diving into constructing models.

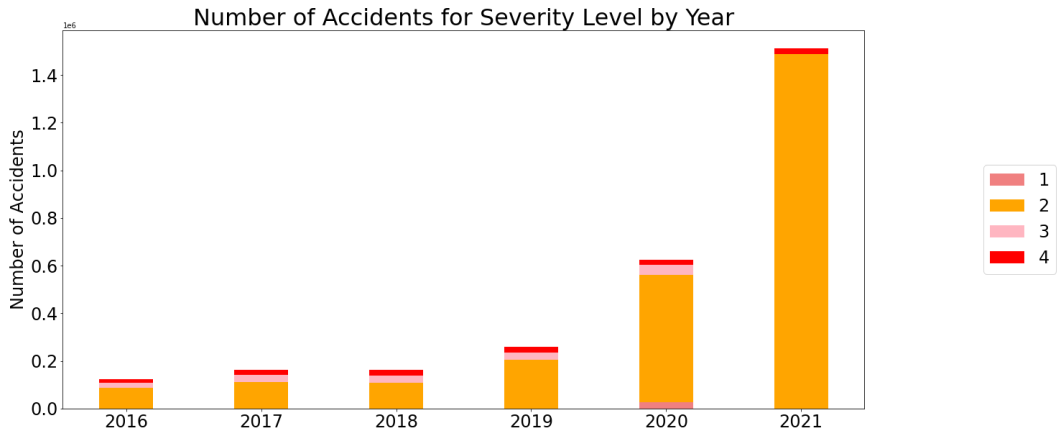

Figure 3.2: Number of Accidents under Each Severity Level by Year

3.1.3 Number of Accidents and Average Severity Level Per State

As we can see from Figure 3.3, California is the state with the highest number of accidents, then we have Florida. The other part of America experienced the generally same number of accidents over the past six years, which is around 300,000.

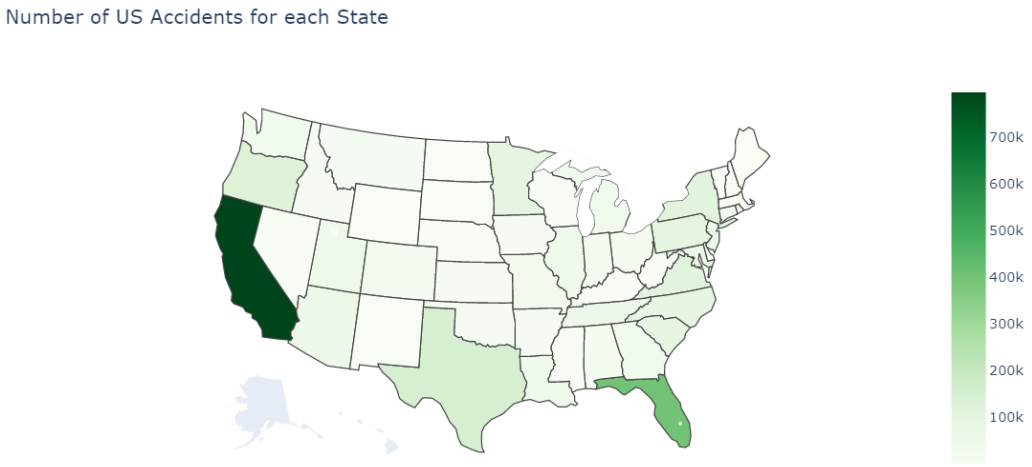

Figure 3.3: Number of US Accidents for each State

Because the number of accidents differs by state and our focus is on the severity level distribution across the United States over the last six years, it would be more useful to depict the average severity rather than the total severity. It is clear from Figure 3.4 that Wisconsin has the highest average accident severity, which is close to 2.83. Midwest America has a greater severity level than the rest of the country.

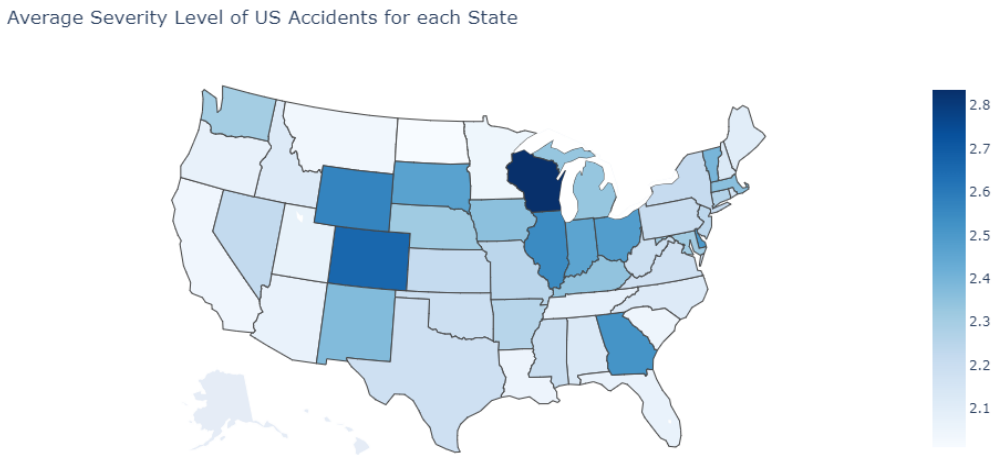

Figure 3.4: Average Severity Level of US Accidents for each State

3.2 Correlation Analysis

Correlation analysis is a way to indicate the intensity of the relationship between variables. Here, we display the Pearson correlation Table 3.1 for continuous features: Start_Lat, Start_Lng, Temperature(F), Humidity(%), Pressure(in), Visibility(mi), Wind_Speed(mph), and time diff.

	Start_Lat	Start_Lng	Temperature(F)	Humidity(%)	\
Start_Lat	1.00	-0.15	-0.48	0.01	
Start_Lng	-0.15	1.00	0.03	0.17	
Temperature(F)	-0.48	0.03	1.00	-0.37	
Humidity(%)	0.01	0.17	-0.37	1.00	
Pressure(in)	-0.23	0.21	0.14	0.14	
Visibility(mi)	-0.09	0.03	0.21	-0.36	
Wind_Speed(mph)	0.03	0.09	0.08	-0.17	
time diff	-0.01	0.00	-0.00	0.01	

	Pressure(in)	Visibility(mi)	Wind_Speed(mph)	time diff
Start_Lat	-0.23	-0.09	0.03	-0.01
Start_Lng	0.21	0.03	0.09	0.00
Temperature(F)	0.14	0.21	0.08	-0.00
Humidity(%)	0.14	-0.36	-0.17	0.01
Pressure(in)	1.00	0.04	-0.03	0.01
Visibility(mi)	0.04	1.00	0.04	0.00
Wind_Speed(mph)	-0.03	0.04	1.00	0.00
time diff	0.01	0.00	0.00	1.00

Table 3.1: Pearson Correlations between Continuous Features

Filter out correlation values greater than 0.2 and display them on the Figure 3.5:

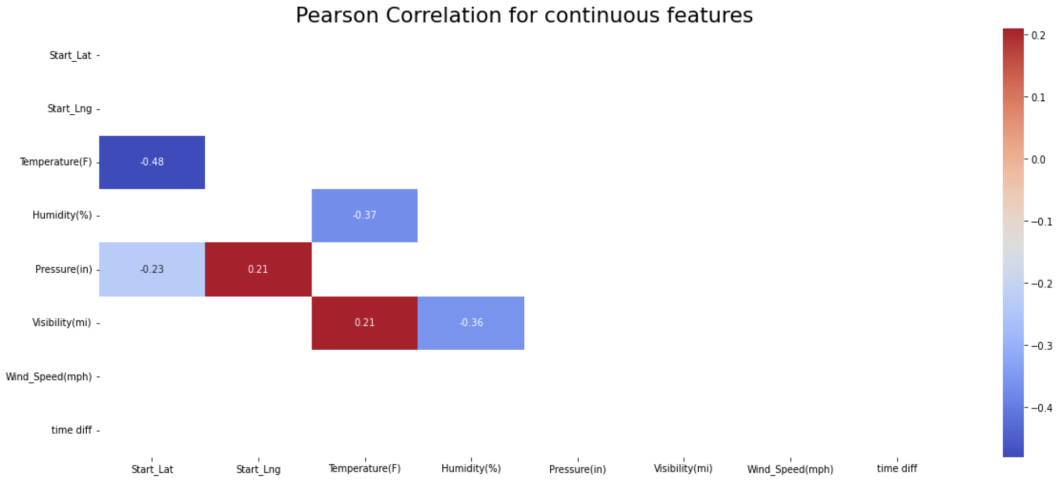

Figure 3.5: Heatmap for Pearson Correlations for Continuous Features

From Figure 3.5, we find there is a moderate relationship between Temperature and Start_Lat and weak relationships between Pressure and Start_Lat, Temperature and Humidity, Visibility and Humidity, Pressure and Start_Lng, Temperature and Visibility.

3.3 Factor Analysis

Factor Analysis is commonly used to reduce the variables to fewer numbers of factors, and to discover the hidden theoretical structure of the phenomenon. It is basically different from principal component analysis in that PCA extracts the maximum variance to form lower dimension components while factor analysis focuses on the covariances from all variables.
By implementing factor analysis, we search the influential latent variables of this dataset.

We start with scaling the numerical data to make them have an equal effect. Then we perform Barlett's Test to check if there is a correlation between the variables of the balanced dataset. The null hypothesis assumes the correlation matrix is an identity matrix with no correlation between the variables. Our goal is to reject the null hypothesis. By using the calculate_barlett_spherity function in python, we obtain the p_value around 0. Since the p_value is less than 0.05, we reject the null hypothesis. Then the correlation presented in this dataset allows us to do factor analysis. Next, we perform the Kaiser-Meyer-Olkin(KMO) test to measure the proportion of variance that can be common among the variance. The KMO score given by our data is around 0.63. This score indicates that we can continue to perform factor analysis.

The total number of factors equals the number of variables, but not all of them are important. For choosing the number of factors, we use the scree plot with eigenvalues. The value of the eigenvalues represents the amount of variance the factor can explain.

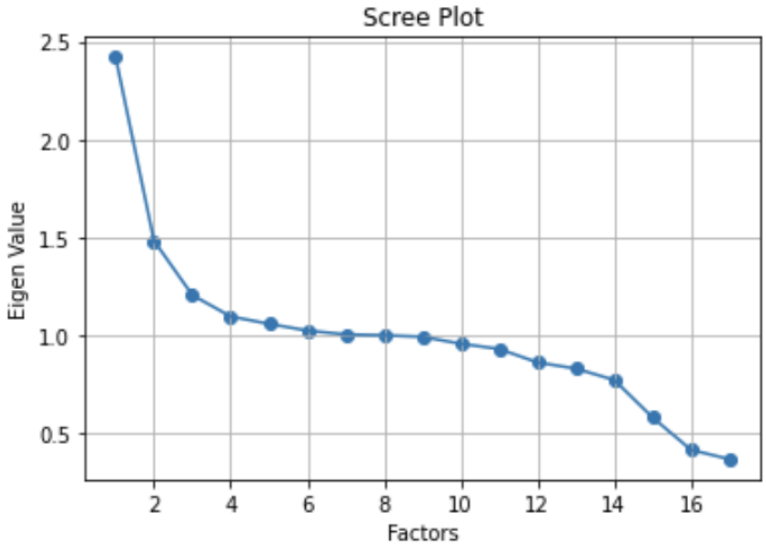

Figure 3.6: The scree plot shows the eigenvalues for each factor.

We only choose the factors that have eigenvalues that are greater than 1 because the data is standardized. If the eigenvalue is greater than 1, it indicates that the factor can explain more than one single observed variable. Thus, from the scree plot in Figure 3.6, the eigenvalue starts from the 7th factor drop below 1. This gives the information that the optimal number of factors is 6.

The factor loadings shown in table 3.2 indicates the influence of a factor to the variance. The loadings range from -1 to 1, the values close to -1 and 1 means that the factor can explain much of the variable, but if the loading is close to zero, that means the factor can explain little of the variable.

	0	1	2	3	4	5
Bump	0.122164	-0.004053	-0.214844	0.883824	0.210456	0.016338
Crossing	-0.235434	-0.013849	0.733552	-0.191001	-0.414703	-0.010537
Give_Way	-0.114871	-0.004867	0.305320	0.190707	-0.127005	0.052322
Junction	0.053096	-0.001183	-0.482658	0.176826	-0.017491	0.006262
No_Exit	0.005213	0.001338	0.017446	-0.026123	0.433294	0.008940
Railway	-0.016486	-0.003757	0.195086	0.003966	0.071711	-0.001221
Roundabout	-0.009243	0.001034	-0.005905	0.004500	-0.002694	0.004153
Station	0.691157	0.014002	-0.094780	0.069247	0.059624	0.273356
Stop	0.059565	0.061009	0.003373	0.000894	0.007477	-0.007472
Traffic_Signal	-0.199401	-0.033554	0.054362	0.001502	0.013279	-0.015885
Sunrise_Sunset	0.079072	0.014268	-0.029015	0.008852	0.004588	0.024504
Temperature(F)	0.036165	-0.007309	-0.011902	-0.009663	0.011005	0.589578
Humidity(%)	-0.004112	-0.000753	-0.004646	0.002928	0.001863	-0.000422
Pressure(in)	0.148557	0.008473	0.033399	0.020041	0.001702	0.278811
Visibility(mi)	0.008161	0.996223	-0.014843	-0.011549	-0.024917	0.039071
Wind_Speed(mph)	0.832705	-0.054681	-0.022399	0.071623	0.081779	0.090736
Precipitation(in)	0.059099	0.002815	-0.049481	0.242412	0.454138	0.003978

Table 3.2: The factor loadings for the variables

From table 3.2, we can see that the first factor has high factor loadings for Station and Wind_Speed. This is interesting because usually we would not think that Station and Wind_Speed can have common variance. Factor 2 has high loadings for junction and crossing. In this case, we can see that factor explains the common variables in road situations where there is more likely to be a fork road.

The amount of variance explained by each factor and the cumulative variance can be found in table 3.3.

	0	1	2	3	4	5
Variance	1.334363	1.001081	0.966075	0.955389	0.643085	0.513999
Proportional Var	0.078492	0.058887	0.056828	0.056199	0.037829	0.030235
Cumulative Var	0.078492	0.137379	0.194207	0.250406	0.288235	0.318470

Table 3.3: The amount of variance explained by each factor

The first row in table 3.3 shows the variance explained by each factor, and the third row is the cumulative variance. The 6 factors can together explain a total 31.8% variance. This performance is actually poor. This result may be due to two reasons, one is that we have mixed variables of both numerical and categorical, and also we have done the multicollinearity detection beforehand.

Finally, it is no harm to see the communalities that indicate the proportion of variance of each variable can be explained by the variables in table 3.4.

	Communalities
Bump	0.886803
Crossing	0.802291
Give_Way	0.161676
Junction	0.267392
No_Exit	0.188839
Railway	0.043504
Roundabout	0.000166
Station	0.569951
Stop	0.007394
Traffic_Signal	0.044273
Sunrise_Sunset	0.007998
Temperature(F)	0.349320
Humidity(%)	0.000051
Pressure(in)	0.101396
Visibility(mi)	0.995027
Wind_Speed(mph)	0.716940
Precipitation(in)	0.270970

Table 3.4: The proportions of variance explained by the factors for each variable

In table 3.4, we can see the proportion of variance of visibility and bump that can be explained by the factors being extremely high (99% and 89%), but it can hardly explain the variance of roundabout and sunrise_sunset.

4. Model Implementation

4.1 Model Selection Method

Even after data processing and exploratory data analysis, there are still around 20 features left as predictors, which is a very high dimensional feature space. Thus, we want to see which of the features are the most important ones that can be used in our machine learning model. To reduce the dimension of feature space, we implemented the sequential feature selection technique with python. The sequential feature selection selects the features based on certain criteria and functions like the mean squared error for regression problems and misclassification rate for classification problems, and the algorithm stops when either adding or reducing the features can increase the loss.

The two sequential feature selections we used are forward selection and backward selection. The sequential forward selection first selects the best single feature and then selects the feature from the rest of the features to form a pair of best features with the first chosen feature. This process is repeated until a predefined number of features are chosen. The sequential backward selection starts with all the features and removes the features from the set until the removal of further features can increase the loss.

4.2 Nonlinear Models ——— Multinomial-Logistic

Multinomial regression is a generalized linear model used to estimate the probabilities for m categories of a qualitative dependent variable Y and assign the category which has the highest probability, using a set of explanatory variables X:

$$\Pr(Y_{ik}) = Pr(Y_i = k|x_i; \beta_1, \beta_2, \dots, \beta_m) = \frac{\exp\left(\beta_{0k} + x_i\beta_k^T\right)}{\sum_{j=1}^{m} \exp\left(\beta_{oj} + x_i\beta_j^T\right)} \; with \; k = 1,2, \dots, m$$

where β_k is the column vector of regression coefficients of X for the kth category of Y.

By using sequential forward feature selection, we selected five important features (Temperature, Humidity, Pressure(in), Bump, and Crossing). Since we know that there are many accidents near stations and roundabouts, we want to keep these two variables. So, the data used to train the multinomial logistic model is based on these seven features.

4.2.1 Purely-balanced data

As mentioned above, there is a huge imbalance among the severity cases (ie, the majority of severity 2 cases and the minority of other cases). Rows were randomly deleted from the majority class to match them with the number of the minority class, which is mentioned as undersampling. 2016 - 2021 would be the first focus, and since severity 1 has minority cases, we randomly selected the same number of cases for severity 2, 3, and 4 as 1 to build the balanced data, and randomly split 70% of balanced data into training data and 30% into test data. Then we applied regularized multinomial logistic regression to the training data. Also, the penalty term is set to avoid overfitting and decrease the variance of prediction. A popular type of penalty is the l2 penalty which adds the (weighted) sum of squared model coefficients to the loss function, encouraging the model to reduce the size of the weights along with the error while fitting the model. The cost function is:

$$min_{\beta,\beta_0} \frac{1}{2}\beta^T\beta + C\sum_{i=1}^{n} \log\left(\exp\left(-y_i\left(X_i^T\beta + \beta_0\right)\right) + 1\right)$$

where C is the inverse of a regularization strength.

This means that a higher C indicates we pay more attention to the training data and a lower value of C indicates the model gives complexity more weight at the cost of fitting the data. We estimate the parameters by minimizing the cost function. Then we used grid search cross-validation to select the best C among {50, 20, 10, 5, 1, 0.5, 0.1, 0.01, 0.001} and get the best C = 5. The final step was to apply the best-selected model (C = 5) to fit the test data and the result of test accuracy is around 0.363.

Instead of the 2021 dataset, 2020 data becomes our first choice because 2020 data contains 4 types of severity, while 2021 data which has only two types of severity 2 and 4. We used the undersampling method for the 2020 data and created balanced data about 2020. (ie, 20652 cases for each severity). Likewise, we randomly split 70% of the balanced data into training data and 30% into test data. Then, based on the training data, we used grid search

cross-validation to select the best C among 50, 20, 10, 5, 1, 0.5, 0.1, 0.01, 0.001 and chose the best C = 10. After all, we applied the best-selected model to fit the test data and get a test accuracy of about 0.362. The confusion matrix is as follows:

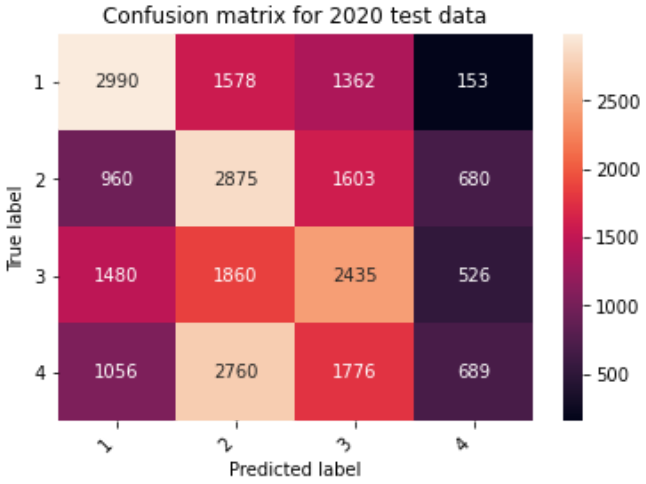

Figure 4.1: Confusion Matrix for 2020 Test Data

Figure 4.1 gave us detailed information that every type of severity is more than 50% to be predicted incorrectly, which needs to be improved.

After diving into the model performance for different time periods, California State (CA) is the new research subject since it has the most traffic accidents among all states. After the same steps as stated above, we obtained an accuracy value of the best model for California is 0.364. The above analysis can be summarized by the following table:

Purely-Balanced Data	Data information	Best C	CV-accuracy	Test-accuracy
2016-2021	23556 cases for each severity	5	0.363	0.359
2020	20652 cases for each severity	10	0.362	0.362
CA	4189 cases for each severity	0.01	0.364	0.360

Table 4.1: Results of Purely-balanced Data

4.2.2 Relatively imbalanced data

One problem of purely-balanced data (ie, the same number of cases for each severity) is that it has low test accuracy. So, instead of constructing purely-balanced data, we modified our data by randomly reducing the cases of severity 2 to 400,000 and not changing the number of other types of cases to decrease the degree of imbalance of data and increase our prediction accuracy.

Then, we fit the regularized multinomial logistic model and used the grid search to choose the best C to get the best models for relatively imbalanced 2016-2021 data, 2020 data, and CA data. The results are as follows:

Relatively-Imbalanced data	Data information	Best C	CV-accuracy	Test-accuracy
2016-2021	S1:23556 S2:400,000 S3:71130 S4:65895	20	0.715	0.714
2020*	S1:23556 S2:93909 S3:40371 S4:20652	0.001	0.55	0.55
CA	S1:4189 S2:114279 S3:6292 S4:4213	50	0.885	0.887

Table 4.2: Results of Relatively-imbalanced Data

**This relatively imbalanced data is a reduced data in severity 2 from original data. In the second column, Si stands for the number of cases for each severity i.

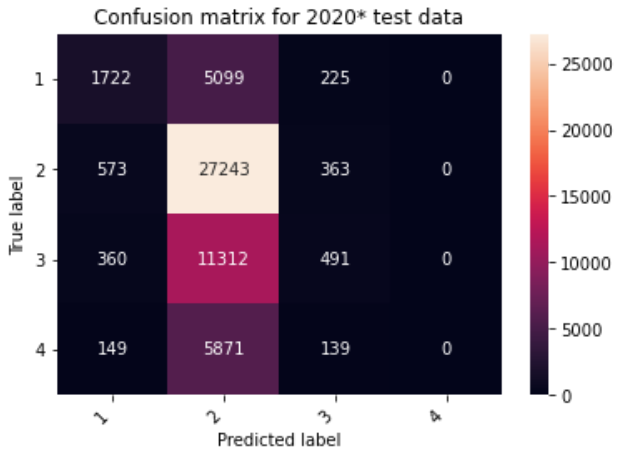

Figure 4.2: Confusion Matrix for 2020* Test Data

Figure 4.2 shows that the majority of severity-2 cases are predicted correctly and the majority of severity 1 and severity 3 are predicted incorrectly and none of severity 4 is predicted correctly. But indeed the test accuracy improves compared with balanced data because severity 2 accounts for a large portion of the data.

In conclusion, using relatively imbalanced data(more cases in severity 2), the test accuracy increases at the cost of misclassification of severity 1, 3, and 4. We prefer to use the relatively imbalanced data to build the multinomial logistic model because it can give us a state-of-art accuracy value.

4.3 Deep Learning Model —— DecisionTree

For this problem, the target for us is severity, which is a categorical variable. Considering this, despite multinomial logistic classification, we also think decision trees would be a good fit since they are relatively easy to understand and are also very effective under this situation. A decision tree's basic goal is to divide a large amount of data into smaller parts. The prediction has two stages. The tree is generated, tested, and optimized using the collected dataset in the first stage. The model is then used to forecast an unknown outcome in the second stage. Here, in the first step, we divided the dataset into training and testing data and used grid search in the training dataset to help us find the values of the best parameters for the decision tree, such as max_depth.

Moreover, since we don't need to predict future severity, the second stage would be evaluating the model in our randomly chosen test dataset.

Similar to Multinomial Logistic Regression, the severity level of accidents in the year 2020, 2016-2021, and the State of California is what we were interested in. (The reason why we chose 2020 rather than 2021 was explained in the last section) As mentioned before, our target severity is a categorical variable, so the type of our decision tree would be a classification decision tree. It is important to notice that another type is the regression tree, which is used to predict continuous quantitative data.

By using sequential feature selection, which was explained in section 4.1, the five features that will be used for building our classification decision tree model are ['Visibility(mi)', 'Bump', 'Give_Way', 'Roundabout', 'Stop']. Then, we randomly split data into training and testing sets and applied the Grid search method, and set different values for 'criterion' and 'max_depth' to find the best parameters for the decision tree. Table 4.3 shows the parameters and accuracy for models in balanced and imbalanced datasets.

Dataset	Criterion	Max_depth	Model Accuracy(%)
2016-2021(balanced)	entropy	7	37.6
2020(balanced)	entropy	8	62.38
California(balanced)	gini	8	49.54
2016-2021(imbalanced)	entropy	8	67.42
2020(imbalanced)	entropy	4	47.59
California(imbalanced)	entropy	4	86.59

Table 4.3: Accuracy of Different Dataset

From Table 4.3, we can observe that for the same subject, such as the State of California, the accuracy of the balanced dataset is higher than that of the imbalanced dataset. The reason behind this phenomenon is the same as section 4.3, which is balancing data is at the cost of decreasing accuracy since severity 2 should have been a dominant category.

We then visualized the classification tree of the state of California over the past six years for balanced data and imbalanced data. As shown in Figure 4.3, and Figure 4.4 there are 16 and 14 leaves for each decision tree. The most important feature in this tree is visibility, which makes sense because it would be more dangerous if driving under limited vision.

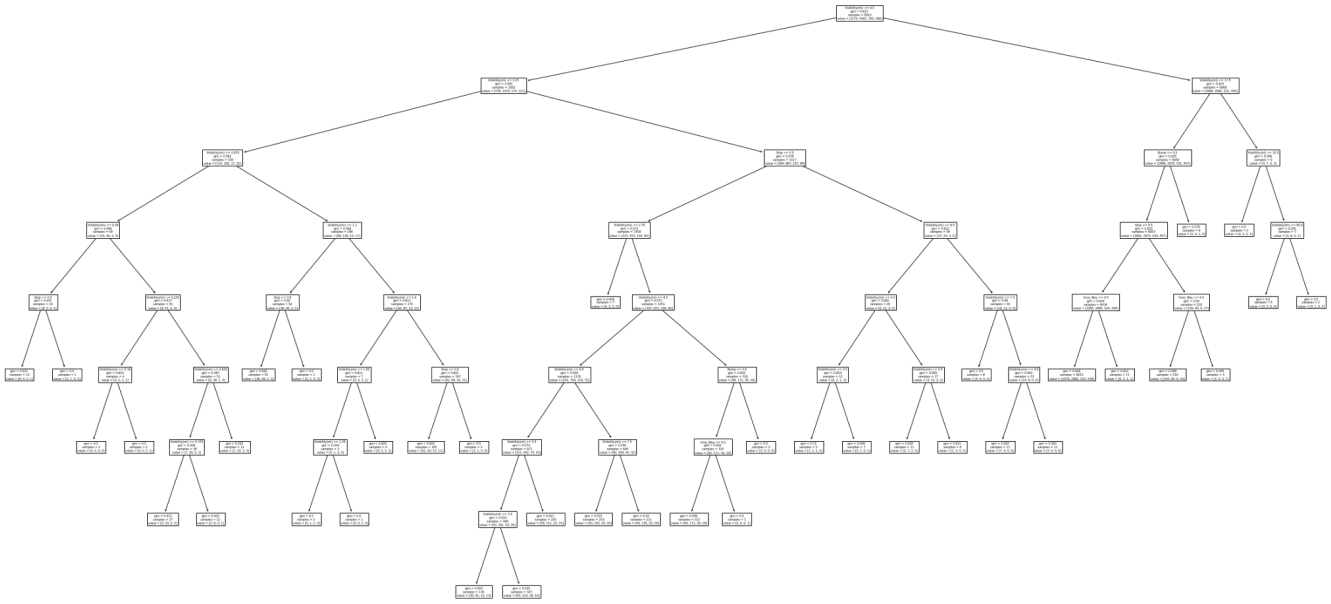

Figure 4.3: Decision Tree Visualization of Balanced Data

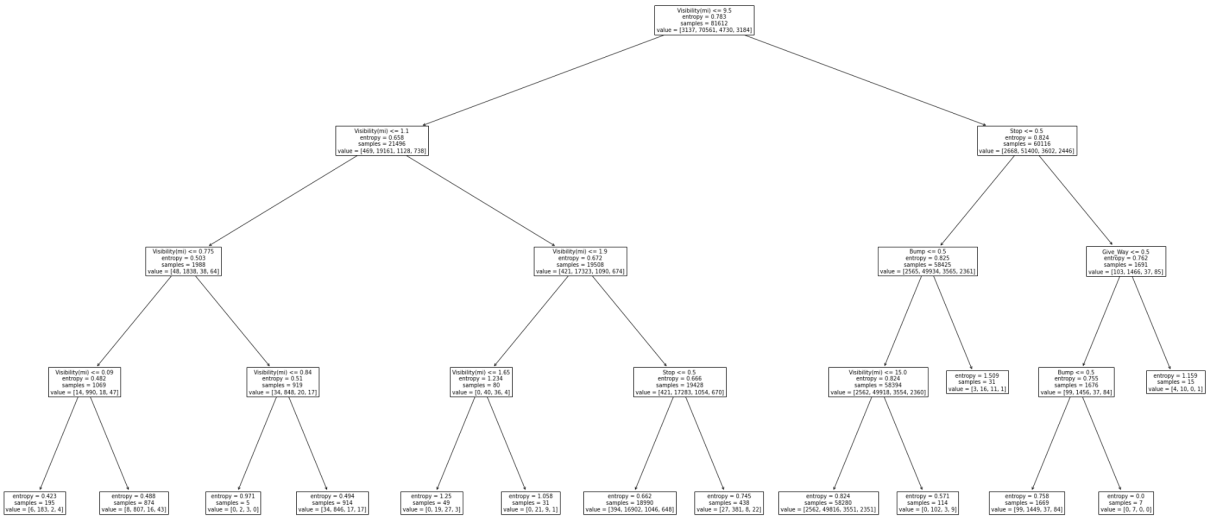

Figure 4.4: Decision Tree Visualization of Imbalanced

4.4 Times Series Model —— ARIMA

In addition to predicting the severity of accidents, forecasting the number of accidents also plays an essential role for decision-makers (DM) to deploy as well as scheduled public officials. In this part, we attempt to apply time series methods to predict the number of accidents each week and provide supplementary assistance to DM.

Auto-Regressive Integrated Moving Average (ARIMA) is the combination of two models, the auto-regressive (AR) and the moving average (MA) models. An autoregressive AR(p) model with lag p can be defined as

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + \omega_t,$$

where $\omega_t \sim N\left(0, \sigma_\omega^2\right)$, and the value of X today is dependent on today's fluctuation and previous p values of X. The moving average MA(q) model with lag q, however, is given as

$$X_t = \omega_t + \theta_1 \omega_{t-1} + \ldots + \theta_q \omega_{t-q},$$

and the value of X today is dependent on today's fluctuation and the previous q fluctuations. Then, ARMA(p,q) is a combination of AR(p) and MA(q), defined as

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + \omega_t + \theta_1 \omega_{t-1} + \theta_2 \omega_{t-2} + \ldots + \theta_q \omega_{t-q},$$

and when the time series is not stationary, we can use ARIMA(p,q,d) to achieve stationarity by taking d differences.

We take New York state as an example to examine the effectiveness of ARIMA in forecasting the number of accidents in each week. The time series, shown in Figure 4.5, is collected between 03/23/2016 and 12/31/2021 with a total of 302 weeks.

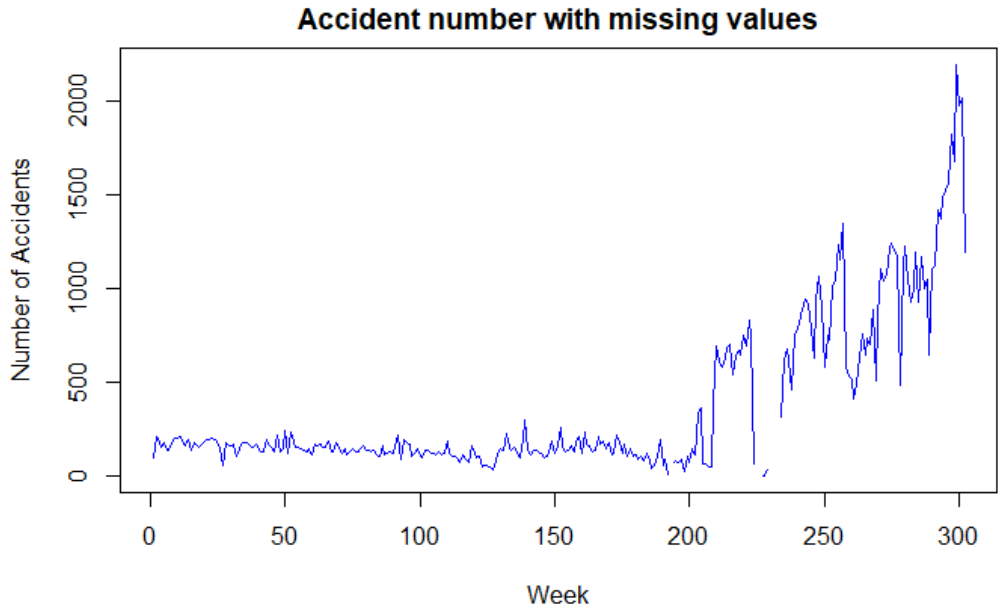

Figure 4.5: Accident Number with Missing Values

To deal with the missing values between the 180th and 230th week, we first observe that there does not exist an apparent seasonal pattern in our dataset, and thus, apply linear interpolation to fill in the missing values. Besides, in Figure 4.5, since data starting from the 200th week shows some abnormal fluctuations, it is necessary to check for possible outliers, and if so, also replace them with linear interpolation. The preprocessed data is shown in Figure 4.6.

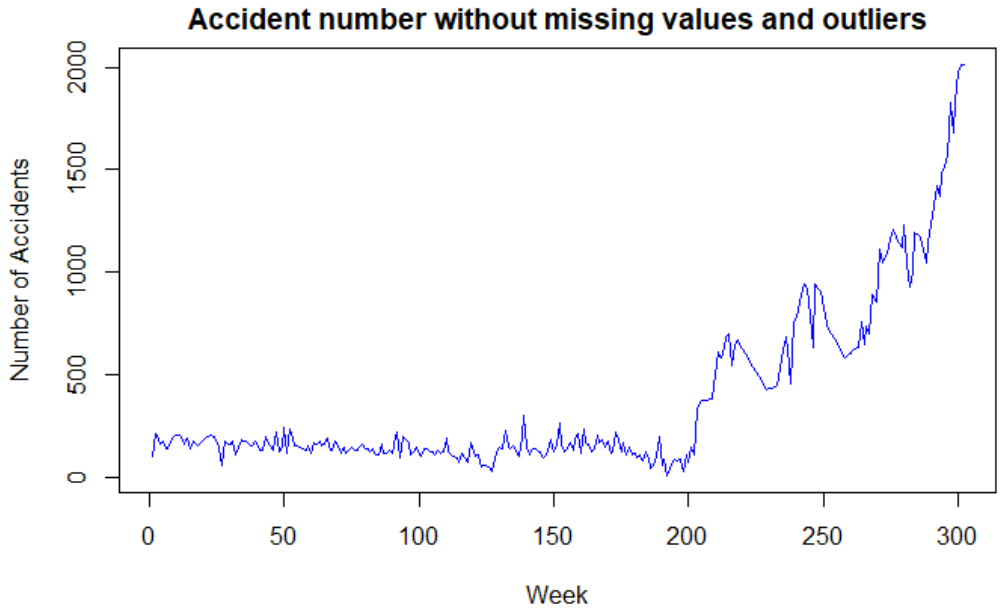

Figure 4.6: Accident Number without Missing Values and Outliers

In order to perform a successive modeling on our time series, we must assure that the data is stationary with constant mean and variance for all t, and the autocovariance function between X_{t_1} and X_{t_2} only depends on the interval t_1 and t_2. We use the Dickey Fuller Test to check for stationarity, which returns a p-value of 0.99, resulting in the acceptance of the null hypothesis. Hence, the data is not stationary. To stationize our time series, we perform differencing on our data, and the result of the first two differencing is given in Figure 4.7. Then, we once again perform the Dickey Fuller Test on these two different time series and gain a p-value of 0.01 for both tests. Therefore, since they both are stationary after differencing and have a similar pattern over time, we take the first differencing result as our final time series and set $d = 1$ in our ARIMA model.

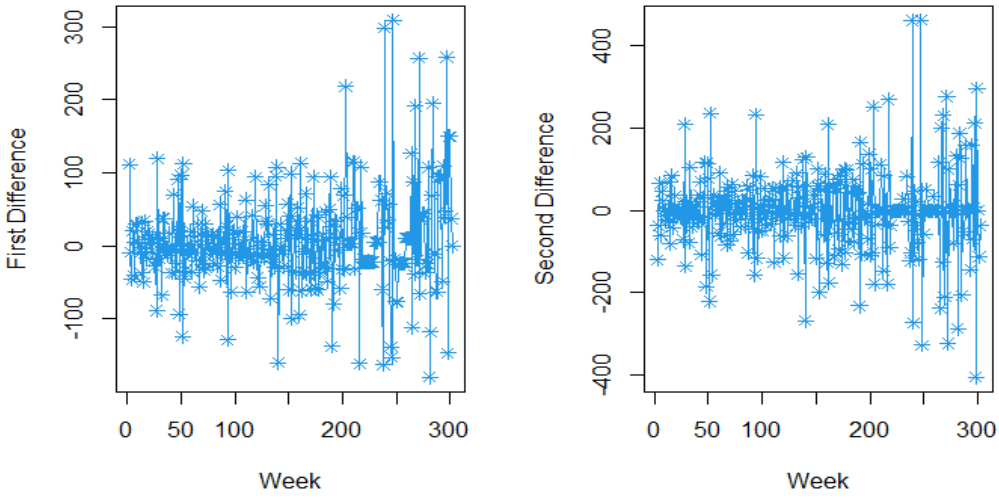

Figure 4.7: Difference Results of Time Series Data

Before our experiment, we first separate the time series data into a training set and testing set with a ratio of 8:2, and perform an automatic ARIMA model on the training set to choose the best parameters. The best model is the one with the lowest AIC score, and the final model is then ARIMA(2,1,2).. To check the assumption of our model, we first perform the Box-Ljung test to check the independence of residuals. Since the obtained p-value is 0.824 with lag = 20, we fail to reject the null hypothesis and conclude that the residuals for our time series model are independent. Besides, since the model also passes the significance test of parameters, we then use the obtained model on the test dataset. The results are shown in Figure 4.8, and we use RMSE as the measure for evaluating the forecast accuracy. The RMSE are defined as follows,

$$\sqrt{\sum_{i=1}^{n}\left(Pred_i - Actual_i\right)/n},$$

and the value of RMSE based on our test data and ARIMA predictions is 341.9946, showing that ARIMA may not be a good method for predicting our time series data.

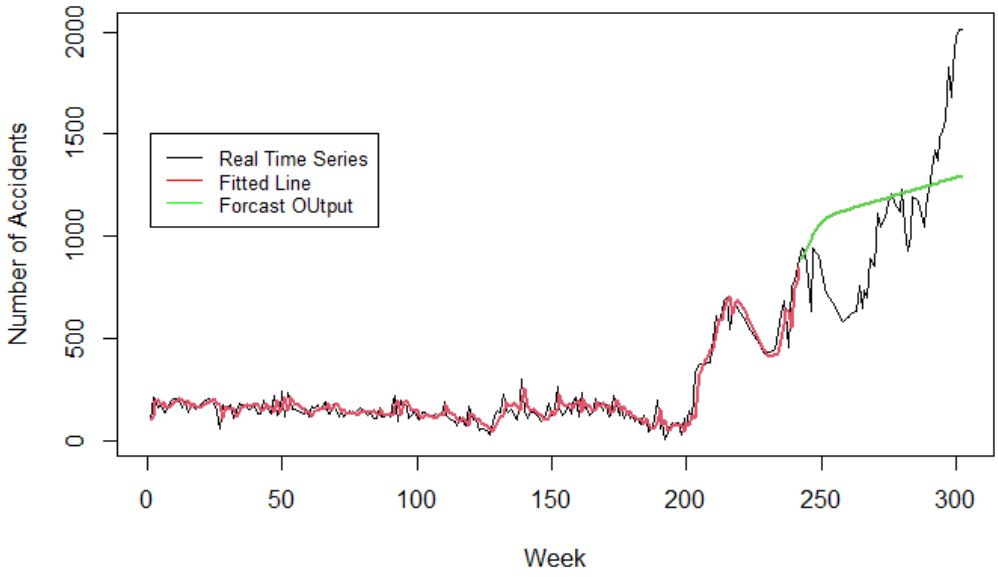

Figure 4.8: ARIMA Prediction Results

To interpret the ineffectiveness of ARIMA model in predicting the number of accidents, we conjecture that it is because of the unexpected covid epidemic, which causes the sudden surge of the accidents around the start of year 2020.

5. Conclusion

For this dataset, first, we preprocessed by removing unimportant columns and missing values. Then, we analyzed exploratory data by visualization such as the number of accidents under each severity level by year. During this process, it is obvious that this traffic dataset existed as a problem of imbalance. To solve this problem, we tried two ways of resampling: the first is randomly selecting an equal amount of data from four severity levels, and the second is randomly selecting higher volumes of data from severity 2. We did correlation analysis and drew a heatmap to check if columns have multicollinearity for further exploration. For model implementation, we applied sequential feature selection to select appropriate features and then compared the performances of a multinomial-logistic model and decision tree model. The accuracy of the decision tree model is higher than the multinomial-logistic model when the dataset is balanced, while for unbalanced data the opposite is true. Finally, we performed the ARIMA model to predict the number of accidents. It simulated well in the first three years but was not a good fit for 2020-2021, and the reason may be the breakout and spread of COVID-19.

6. Reference

[1] Moosavi, Sobhan, Mohammad Hossein Samavatian, Srinivasan Parthasarathy, and Rajiv Ramnath. “A Countrywide Traffic Accident Dataset.”, 2019.

[2] (June 21, 2021) saikat365, https://www.analyticsvidhya.com/blog/2021/06/5-techniques-to-handle-imbalanced-data-for-a-classification-problem/