Predicting Car Accident Severity

Berke Yağmur – October 2020

1. Introduction

1.1 Backround

Traffic accidents are a social issue that causes many deaths or financial losses worldwide. According to the data recorded in 2019, 36,096 fatal accidents occurred. Although fatal accidents decreased compared to the previous year, the number of fatal accidents is still unacceptable.

Thanks to the developing technology and data science, analyzes have been made and what causes traffic accidents has become understandable and classifiable. Many factors such as road conditions, weather conditions, driver attention, technical malfunctions in the vehicle during the accident lead to a significant reduction in the number of accidents. Good analysis of these factors and supporting the necessary classifications with models can help us prevent future traffic accidents.

Traffic accidents are a universal issue that directly concerns governments and some private companies. As a result of the necessary analysis studies, traffic accidents can be reduced significantly. It is important to analyze and work on this sensitive issue that can add significant meaning to human life.

1.2 Problem

This project is about developing a model and predicting new accidents using previously recorded data on traffic accidents. It is about reducing the numerical value of future traffic accidents by making analyzes with variables such as data, road condition, weather, light condition, vehicle type, driver attention.

2. Data

2.1 Data Source

The data set is seattle traffic severity data, which has been recorded since 2004 provided by the ibm course.

2.2 Data Cleaning and Description

When the data set is examined, it is seen that some variables are explanations of each other. We remove variables that we think are redundant from our data set. Then we examine the missing values and see that variables such as intkey, exceptrsncode, exceptrsndesc, pedrownotgrnt have too much missing data. So we exclude them from analysis. By using the necessary statistical methods, we take the variables that we consider important, which will directly affect our results, into our analysis.

After the necessary analysis was done, we reduced the number of variables in our data set from 38 to 12. These variables are;

- Severitycode: A code that corresponds to the severity of the collision
- Weather: A description of the weather conditions during the time of the collision.
- Lightcond: The light conditions during the collision.
- Roadcond: The condition of the road during the collision.
- Speeding: Whether or not speeding was a factor in the collision. (Y/N)
- Underinfl: Whether or not a driver involved was under the influence of drugs or alcohol
- Personcount: The total number of people involved in the collision.
- Pedcylcount: The number of bicycles involved in the collision. This is entered by the state.
- Pedcount: The number of pedestrians involved in the collision. This is entered by the state.

- Vehcount: The number of vehicles involved in the collision. This is entered by the state.
- Junctiontype: Category of junction at which collision took place.
- Crosswalkkey: A key for the crosswalk at which the collision occurred.

We see that the N value is not assigned in the **speeding** variable. First, we assigned the unassigned observations here. Then we assigned values 0 and 1 to be included in the model.

We saw the values of N, 0, Y, 1 in the **underinfl** variable. We combined the values of N and 0 and assigned the value 0. Likewise, we combined the Y and 1 values and assigned the value 1.

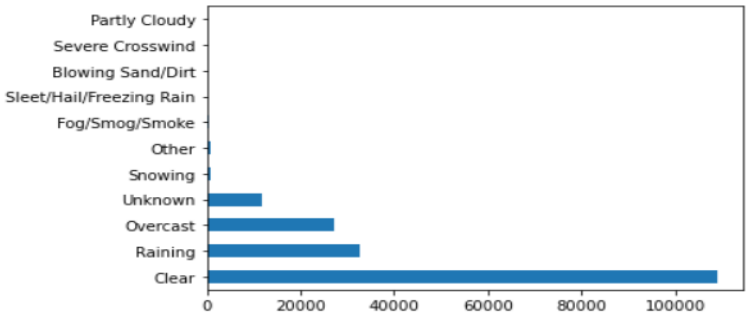
We did the missing value research again on the selected variables.

severitycode	0
weather	5081
lightcond	5170
roadcond	5012
speeding	0
underinfl	4884
personcount	0
pedcylcount	0
pedcount	0
vehcount	0
junctiontype	6329
crosswalkkey	0

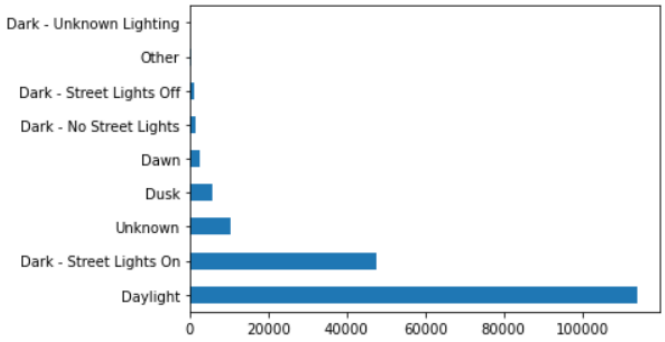
The number of observations in our data set was 194673. We deleted the missing observations line by line. The new observation number of our data set, which we prepared for analysis, was 183196.

In order to make our dataset ready for analysis, we had to convert the variables defined as objects to an integer structure. For this reason, code assignment has been made to the variables listed below.

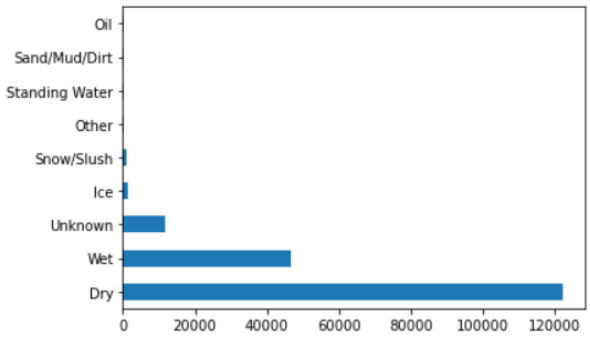
- Weather:



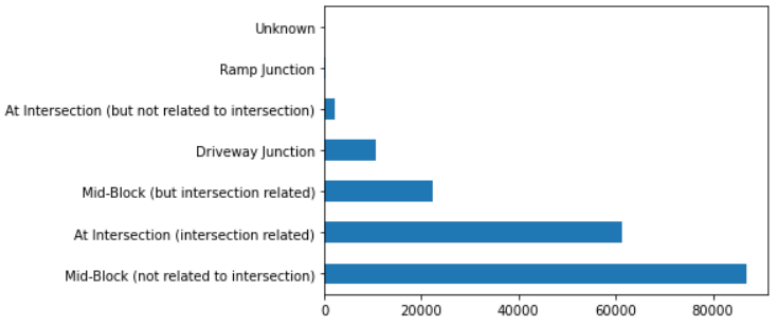
- Lightcond:



- Roadcond:

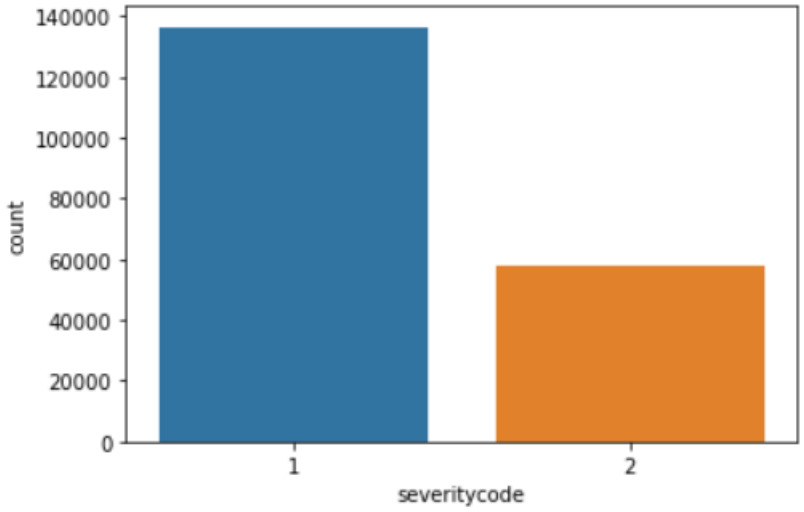


- Junctiontype:

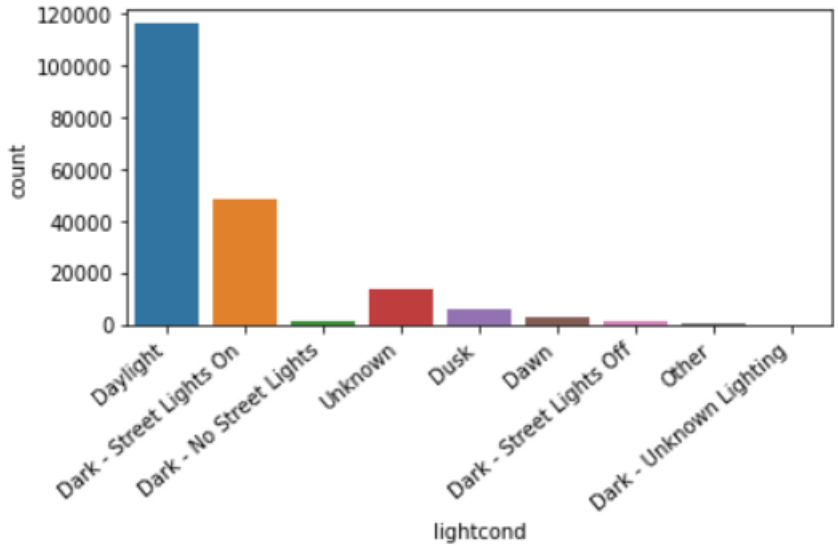


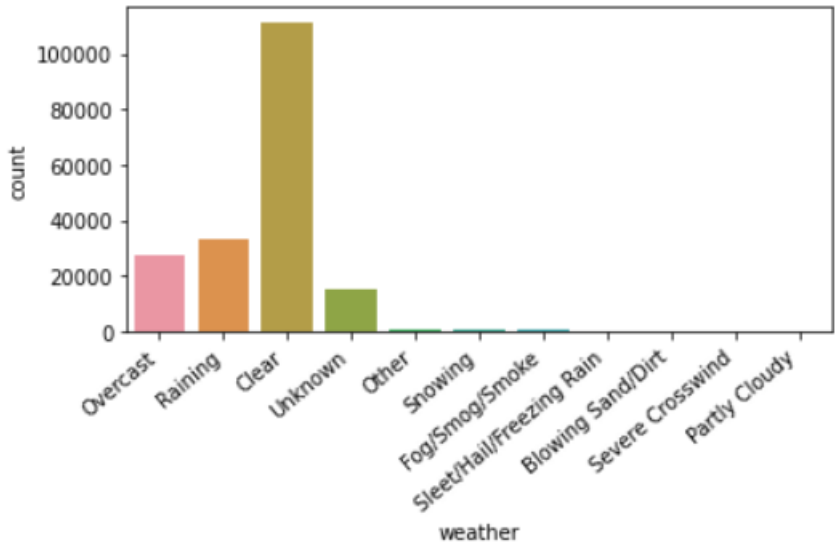
3. Explaratory Data Analysis

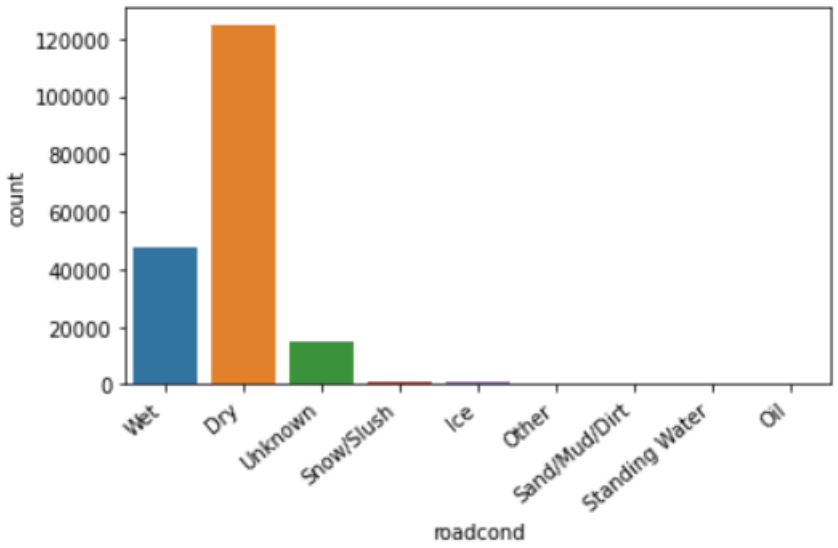
Target Variable:

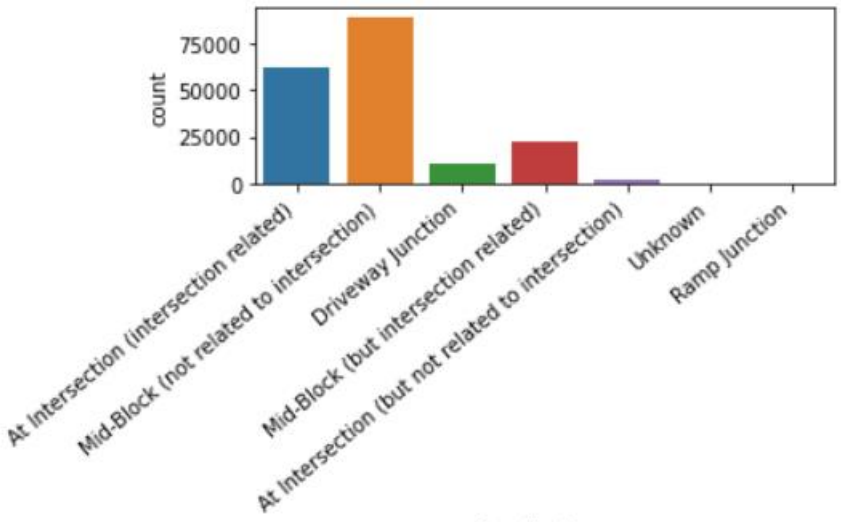


Other Categorical Variable Visualization:




100000
80000
60000
40000
20000
0
count
Overcast
Raining
Clear
Unknown
Other
Snowing
Fog/Smog/Smoke
Sleet/Hail/Freezing Rain
Blowing Sand/Dirt
Severe Crosswind
Partly Cloudy
weather


120000
100000
80000
60000
40000
20000
0
count
Wet
Dry
Unknown
Snow/Slush
Ice
Other
Sand/Mud/Dirt
Standing Water
Oil
roadcond


75000
50000
25000
0
count
At Intersection (intersection related)
Mid-Block (not related to intersection)
Driveway Junction
Mid-Block (but intersection related)
At Intersection (but not related to intersection)
Unknown
Ramp Junction

4. Model Development and Evaluation

After we have our data set ready, we can start building our model. In this section, we first divide our data set into train and test. 80% for train and 20% for testing. As a result, 146556 observations are allocated to the train set and 36640 observations are allocated to the test set. Then we apply the standardization process. We can now put our data set into the model and train it, and examine the results we get.

Here, 7 algorithms have been tried as the classification algorithm. Test and train accuracy of each algorithm has been created. Jaccard scores, f1 scores, confusion matrix have been found. Care has been taken to ensure that the data set is not overfitting or underfitting. The algorithms and their scores, respectively, are listed below.

Algorithms used:

- Logistic regression
- KNN
- DecisionTree
- XGBoost
- RandomForest
- GBM
- CatBoost

- **Logistic Regression Classification Report**

	precision	recall	f1-score	support
1	0.74	0.97	0.84	25221
2	0.80	0.24	0.37	11419
accuracy			0.74	36640
macro avg	0.77	0.61	0.60	36640
weighted avg	0.76	0.74	0.69	36640

Confusion Matrix:

```
[24540,   681]
[ 8709,  2710]
```

- **KNN Classification Report**

	precision	recall	f1-score	support
1	0.75	0.89	0.82	25221
2	0.60	0.35	0.44	11419
accuracy			0.72	36640
macro avg	0.67	0.62	0.63	36640
weighted avg	0.70	0.72	0.70	36640

Confusion Matrix:

```
[22494,  2727]
[ 7394,  4025]
```

- **Decision Tree Classification Report**

	precision	recall	f1-score	support
1	0.73	0.99	0.84	25221
2	0.89	0.19	0.31	11419
accuracy			0.74	36640
macro avg	0.81	0.59	0.58	36640
weighted avg	0.78	0.74	0.68	36640

Confusion Matrix:

```
[24955,   266]
[ 9237,  2182]
```

- **XGBoost Classification Report**

	precision	recall	f1-score	support
1	0.74	0.97	0.84	25221
2	0.79	0.26	0.39	11419
accuracy			0.75	36640
macro avg	0.76	0.61	0.61	36640
weighted avg	0.76	0.75	0.70	36640

Confusion Matrix:

```
[24422,   799]
[ 8497,  2922]
```

- **Random Forest Classification Report**

	precision	recall	f1-score	support
1	0.74	0.95	0.83	25221
2	0.72	0.27	0.40	11419
accuracy			0.74	36640
macro avg	0.73	0.61	0.62	36640
weighted avg	0.73	0.74	0.70	36640

Confusion Matrix:

```
[23982,  1239]
[ 8290,  3129]
```

- **Gradient Boosting Classification Report**

	precision	recall	f1-score	support
1	0.74	0.97	0.84	25221
2	0.80	0.24	0.37	11419
accuracy			0.75	36640
macro avg	0.77	0.61	0.61	36640
weighted avg	0.76	0.75	0.69	36640

Confusion Matrix:

```
[24550,   671]
[ 8649,  2770]
```

- **CatBoost Classification Report**

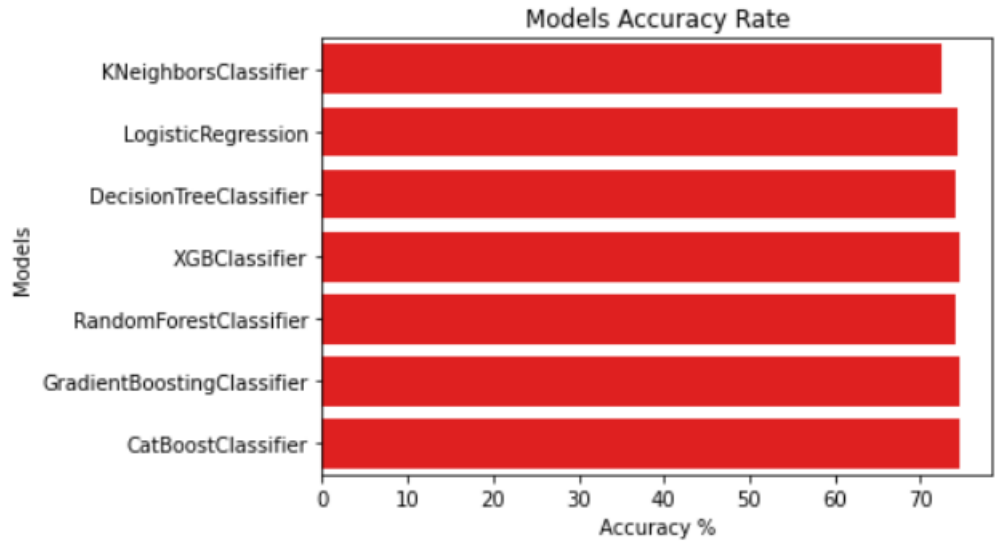
	precision	recall	f1-score	support
1	0.74	0.97	0.84	25221
2	0.78	0.26	0.39	11419
accuracy			0.75	36640
macro avg	0.76	0.61	0.62	36640
weighted avg	0.75	0.75	0.70	36640

Confusion Matrix:

```
[24383,   838]
[ 8450,  2969]
```

5. Conclusion

All the classification algorithms we used gave similar values to each other. We printed out the train and test accuracy of each algorithm to check the overfitting and underfitting status. The less the difference is, the more balanced the structure is. We have seen that all these accuracy scores are valued close to each other. So we observed that there is no high varience or high bias. I did not do the parameter tuning. My own system cannot provide enough time to do the parameter tune. Accuracy values can be improved with the best parameters found using methods such as the GridSearchValidation method.



Algorithm	Jaccard	F1-score	Train-Acc	Test-Acc
KNN	0.689	0.72	0.7357	0.7237
LogisticRegression	0.723	0.74	0.7444	0.7437
Decision Tree	0.724	0.74	0.7429	0.7406
XGBoost	0.724	0.75	0.7515	0.7462
RandomForest	0.715	0.75	0.7641	0.7399
GradientBoosting	0.724	0.75	0.7641	0.7456
CatBoost	0.724	0.75	0.7641	0.7465