Predictive Analysis of Severity of Accidents (SPD)

Capstone Project

Table of contents

Introduction to the Business Problem - Car Accident Severity

The purpose of this project is to help SPD in finding and exploring ways to predict the severity facilitate the best models to avoid future car accidents. We could also help with various relationships and within the given parameters and provide the best report.

The project aims to create an analysis of the collision dataset, measure the severity of these accidents within the dataset and related impacts. It aims to help the PD to make informed decisions in reducing the number of accidents. We will be working with a dataset of the accidents that occurred since 2004

The target audience of the project are departments of the government and the SPD. This project might to a certain extent help the insurance and transportation companies to some extent. It will help people to get awareness and real insights and possibly take action to reduce the number of accidents.

We will also be looking at attributes like weather, visibility, or road conditions that play a huge impact on accidents and how could local administration work ahead minimize the risks in the future.

Data Analysis : Data Understanding:

Dataset contains several attributes such as:

1. SEVERITYCODE
2. X
3. Y
4. OBJECTID
5. INCKEY
6. COLDETKEY
7. REPORTNO
8. STATUS
9. ADDRTYPE
10. INTKEY
11. LOCATION
12. EXCEPTRSNCODE
13. EXCEPTRSNDESC
14. SEVERITYCODE
15. SEVERITYDESC
16. COLLISIONTYPE
17. PERSONCOUNT
18. PEDCOUNT
19. PEDCYLCOUNT

- VEHCOUNT
- INCDATE
- INCDTTM
- JUNCTIONTYPE
- SDOT_COLCODE
- SDOR_COLDESC
- INATTENTIONIND
- UNDERINFL
- WEATHER
- ROADCOND
- LIGHTCOND
- PEDROWNOTGRNT
- SDOTCOLUMN
- SPEEDING
- ST_COLCODE
- ST_COLDESC
- SEGLANEKEY
- CROSSWALKKEY
- HITPARKEDCAR

Data Analysis : Data Understanding:

Data Preparation:

The target variable we will be considering for the analysis will be - **'SEVERITYCODE'** as it used to depict the severity of the accident. The same is denoted as 0 or 1 within the dataset; where

- "0" denotes Property damage
- "1" denotes Severe Injury

Attributes

We analyzed that the attributes used to describe the severity of an accident are:

- **"WEATHER"**
- **"ROADCOND"**
- **"ADDRTYPE"**
- **"COLLISIONTYPE"**
- **"LIGHTCOND"**

Data Analysis : Data Understanding:

Data Cleaning

The data needs to be cleaned as it is not fit for analysis. There are unnecessary columns and also the datatypes needs changing.

From my observation,lot of unified and imbalanced data was found. There was also presence of incomplete information. The dataset was completely analyzed and then cleaned.

Features Selected

Exploratory Analysis of the dataset:

Once the dataset was cleaned, the below features were fine tuned for the model building

- SEVERITYCODE
- WEATHER
- ADDRTYPE
- COLLISIONTYPE
- JUNCTIONTYPE
- ROADCOND
- LIGHTCOND

Meanwhile Dependent and Independent variables were also selected.

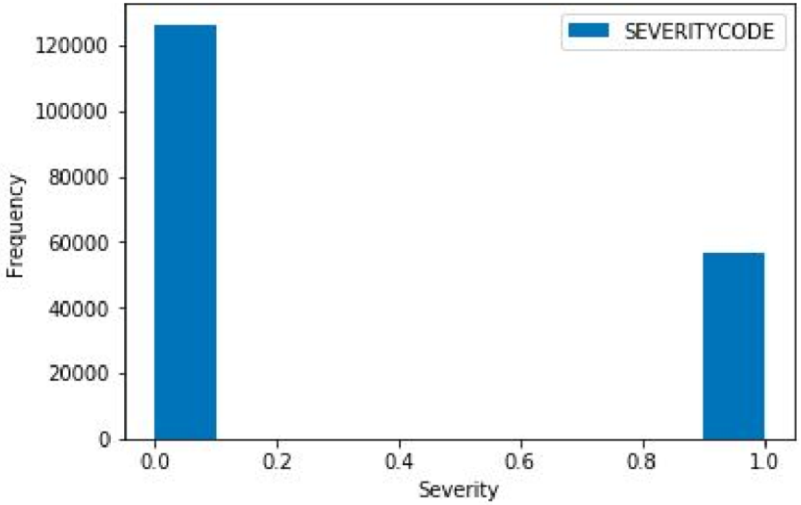
- SEVERITYCODE – Dependent Feature also known as predictor.
- Others would be our Independent features through which we will predict the Severity of the accident.

This was conducted to encode all the features in the required data frame We conducted 2 encoding steps

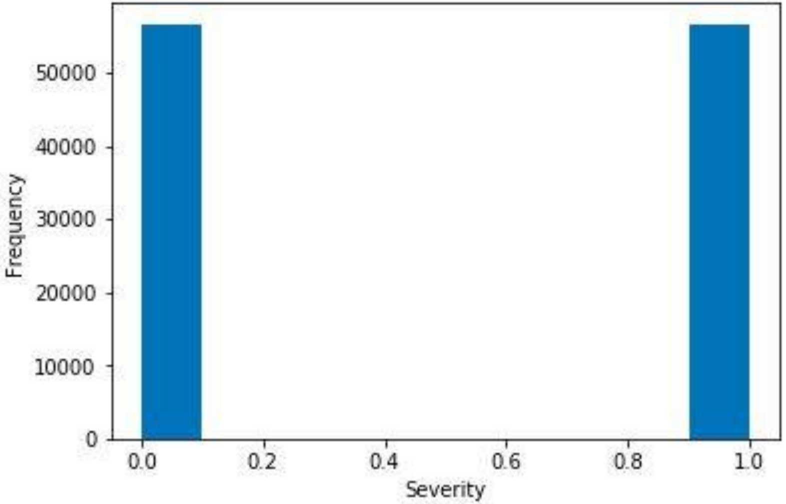
- Frequency Count Encoding – On the below attributes
    - WEATHER
    - COLLISIONTYPE
    - JUNCTIONTYPE
    - ROADCOND
    - LIGHTCOND.
- One Hot Encoding –
    - ADDRTYPE

Okay so we see imbalanced predictor data , so we down-sampled our SEVERITYCODE data to balance the dataset.

```
import matplotlib.pyplot as plt
%matplotlib inline
cdf.plot(kind='hist')
plt.xlabel('Severity')
plt.ylabel('Frequency')
plt.show()
```

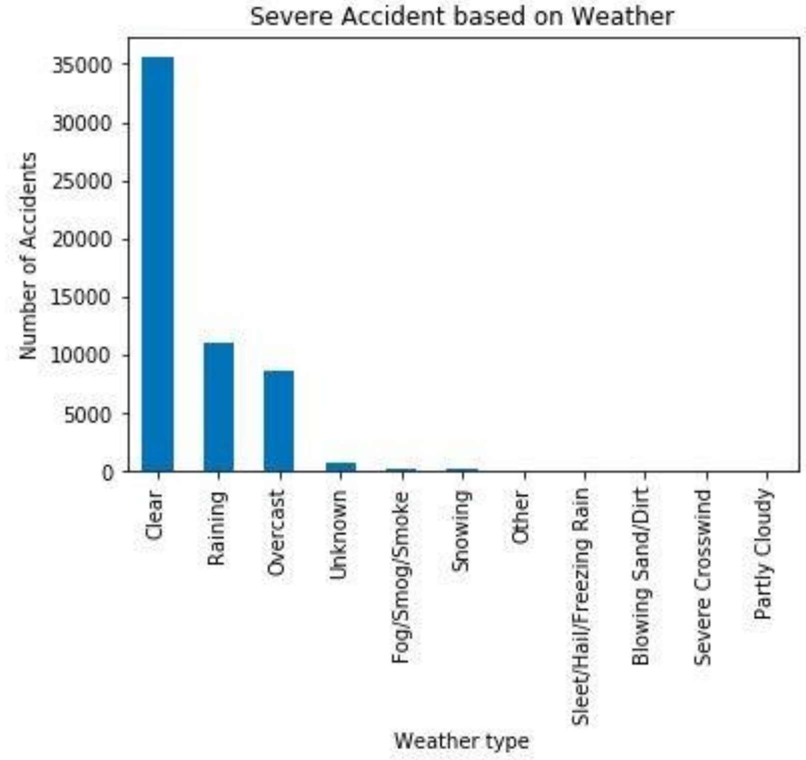


```
df_cust.SEVERITYCODE.plot(kind='hist')
plt.xlabel('Severity')
plt.ylabel('Frequency')
plt.show()
```



After down-sampling

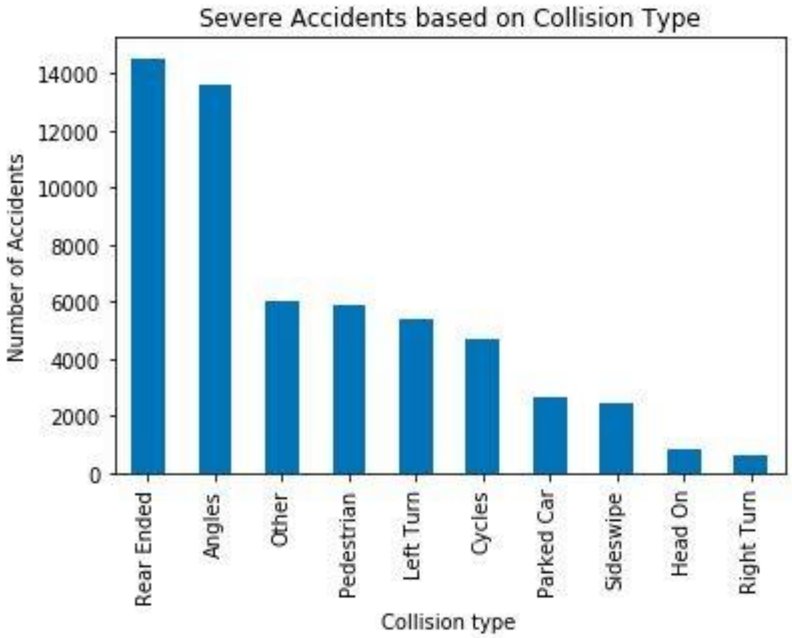
22]:
```
df_cust[df_cust.SEVERITYCODE==1].WEATHER.value_counts().plot(kind=
plt.title('Severe Accident based on Weather')
plt.ylabel('Number of Accidents')
plt.xlabel('Weather type')
plt.show()
```



Severity based on weather

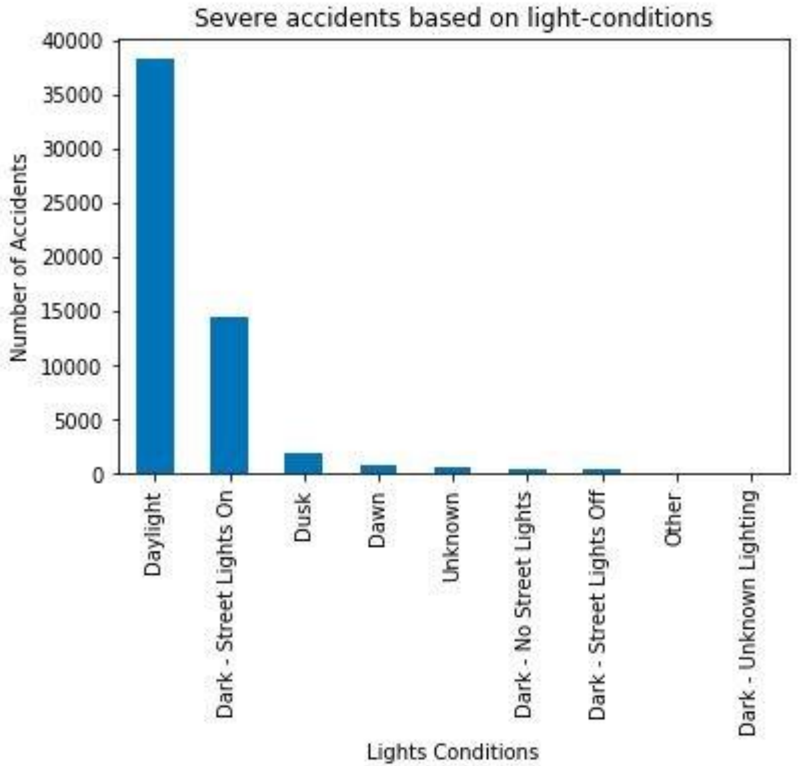
Severity based on Collision Type

```
df_cust[df_cust.SEVERITYCODE==1].COLLISIONTYPE.value_counts().pl
plt.title('Severe Accidents based on Collision Type')
plt.ylabel('Number of Accidents')
plt.xlabel('Collision type')
plt.show()
```



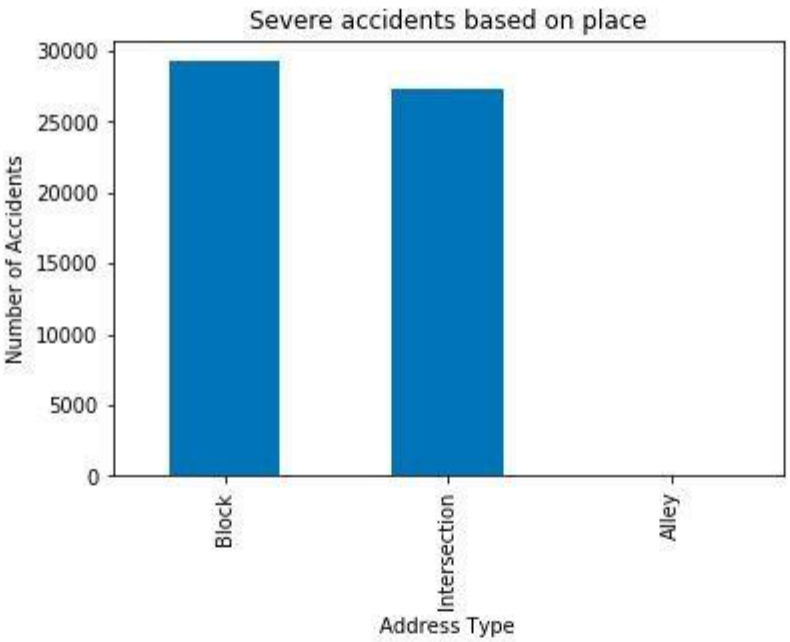
Severity based on Light conditions

```
plt.title('Severe accidents based on light-conditions')
plt.ylabel('Number of Accidents')
plt.xlabel('Lights Conditions')
plt.show()
```

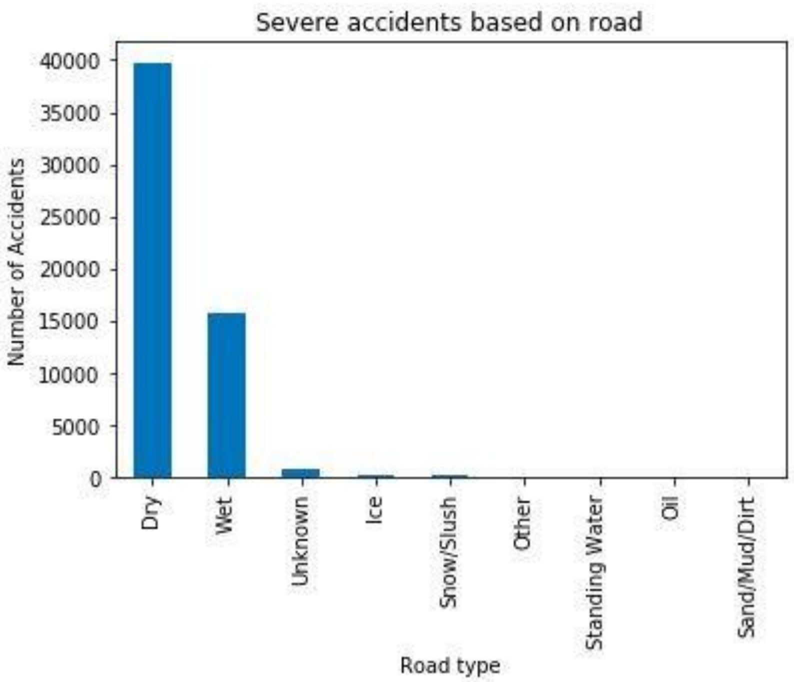


Severity based on Address type

```
plt.title('Severe accidents based on place')
plt.ylabel('Number of Accidents')
plt.xlabel('Address Type')
plt.show()
```



Severity based on road type



Model Development

Results

- Once we selected our Dependent and Independent features, we split our data set for Training and Testing phase.
- 80% of the dataset was used for training the model, while the rest 20% would be used for model evaluation phase.
- As, classification problem, we developed 4 types of model to test the accuracy that which one gives the best output.
- Once training and testing data was complete, we normalized the training set using StandardScaler() function.

Decision Tree – After fitting the decision tree classifier model with our training test, we predict the outcome and matched it with our test data set and got an accuracy of:

- F1 Score: 0.691
- Accuracy-Score: 0.692

K-Nearest Neighbors –

After fitting the K-Nearest Neighbor classifier model with our training test, we predict the outcome and matched it with our test data set and got an accuracy of:

- Accuracy-Score: 0.668

Logistic Regression

So after training the Logistic Regression Classifier model with our training dataset we predicted the outcome of severity with an overall accuracy of:

- f1 score: 0.576
- Accuracy score: 0.597

Support Vector Machine(SVM) –

So after training the SVM Classifier model with our training dataset we predicted the outcome of severity with an overall accuracy of:

- f1 score: 0.708
- Accuracy_score: 0.654

Classification Report

```
from sklearn.metrics import classification_report,confusion_matrix
import itertools
cnf=confusion_matrix(ytest,yhat_svm)
np.set_printoptions(precision=2)
print('Classification Report:\n',classification_report(ytest,yhat_svm))
```

```
Classification Report:
              precision    recall  f1-score   support

           0       0.74      0.47      0.57     11243
           1       0.62      0.84      0.71     11414

   micro avg       0.65      0.65      0.65     22657
   macro avg       0.68      0.65      0.64     22657
weighted avg       0.68      0.65      0.64     22657
```

Conclusion

We were able to conclude that the best classification model for dataset is Decision Tree Classifier model. Also, other observations:

- People might be careless while driving in daytime than night
- Most of the severe accidents happens in clear weather.
- It is also noticed that most of severe accidents happens on dry roads.
- Most of the severe accidents happens at Intersection and Blocks