Car accident severity

PRESENTED BY

VIVEK KUMAR SONI

Introduction

What is an accident ?

An unexpected, unplanned occurrence that may cause injury and property damage.

Data

- What is data ?

Data are characteristics or information, usually numerical, that are collected through observation.

Data are available in many online storage sites.

Data preprocessing

Data cleaning:

- Removing Nan values
- Removing unusual dataset

Data wrangling:

- Converting data into desired data types

Exploratory analysis

1. Descriptive analysis

```
d1=data['ADDRTYPE'].value_counts()
d1
```

```
1.0      123315
2.0       63447
0.0         742
Name: ADDRTYPE, dtype: int64
```

```
d2=data['COLLISIONTYPE'].value_counts()
d2
```

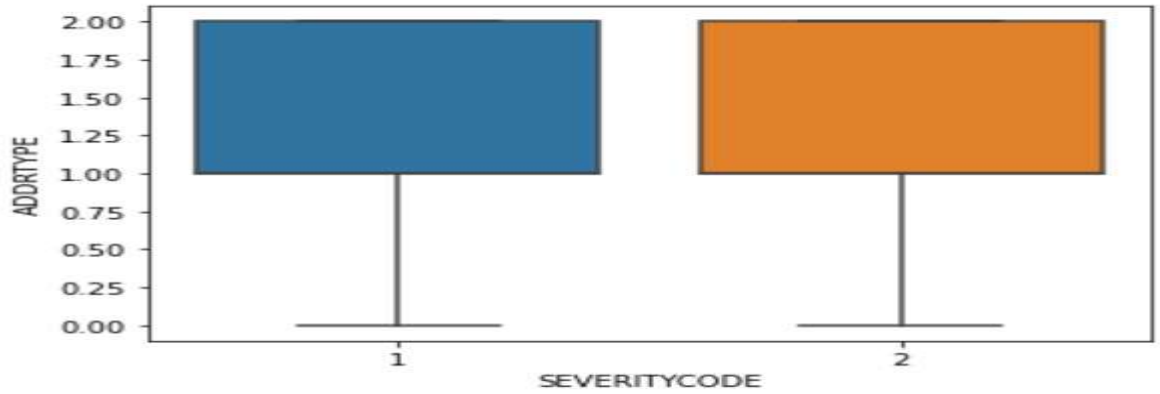
```
5.0      46679
0.0      34555
7.0      33794
4.0      23440
9.0      18442
3.0      13659
6.0       6589
1.0       5399
```

2. Box Plot Analysis

```
import seaborn as sns
```

```
#B1
sns.boxplot(x=data['SEVERITYCODE'],y=data['ADDRTYPE'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x27401e268b0>
```



3. GroupBy Analysis

```
g1=data.groupby(['ADDRTYPE'])['SEVERITYCODE'].value_counts(normalize=True)
g1
```

```
ADDRTYPE  SEVERITYCODE
0.0       1               0.892183
          2               0.107817
1.0       1               0.761367
          2               0.238633
2.0       1               0.568727
          2               0.431273
Name: SEVERITYCODE, dtype: float64
```

4. Pearson correlation analysis

```
from scipy import stats
```

```
from scipy.stats import pearsonr
```

```
#p1
pearson_coef,p_value=stats.pearsonr(data['ADDRTYPE'],data['SEVERITYCODE'])
pearson_coef,p_value
```

```
(0.19971784115718683, 0.0)
```

```
#p2
pearson_coef,p_value=stats.pearsonr(data['COLLISIONTYPE'],data['SEVERITYCODE'])
pearson_coef,p_value
```

```
(-0.12834127033207823, 0.0)
```

```
#p3
pearson_coef,p_value=stats.pearsonr(data['PERSONCOUNT'],data['SEVERITYCODE'])
pearson_coef,p_value
```

```
(0.1283681223505565 6, 0.0)
```

Modeling

Decision tree

A decision tree is a flowchart-like structure in which each internal node represents a test on a feature (e.g. whether a coin flip comes up heads or tails) , each leaf node represents a class label (decision taken after computing all features) and branches represent conjunctions of features that lead to those class labels. The paths from root to leaf represent classification rules.

Logistic regression:

Logistic regression is a classification algorithm, used when the value of the target variable is categorical in nature. Logistic regression is most commonly used when the data in question has binary output, so when it belongs to one class or another, or is either a 0 or 1.

KNN:

K-nearest neighbors (KNN) algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression predictive problems. However, it is mainly used for classification predictive problems in industry. KNN is a lazy learning algorithm because it does not have a specialized training phase and uses all the data for training while classification. KNN is also a non-parametric learning algorithm because it doesn't assume anything about the underlying data.

Result

METHOD	ACCURACY
Decision tree	75%
Logistic regression	72%
KNN	75.5%

Conclusion

In this project I have made three model namely Decision tree, Logistic regression and KNN. On basis of different analysis and result I can conclude that KNN is best classifier for this model. This is because KNN gives best result if number of classes are two. But accuracy of Decision tree was also good.

THANK YOU