Car Accident Severity Report

(IBM Coursera Applied Data Science Capstone Project)

Introduction

- According to World Health Organization road traffic injuries
  - were the tenth leading cause of death globally in 2010
  - were the eighth leading cause of death globally in 2016
  - are the main cause of death among those aged 15–29 years

- How can we minimize the severity of road traffic accidents?
- Based on various factors, are we able to predict how severe injuries will be?

Data

- The data used in this project was collected by the Seattle Police Department, recorded by Traffic Records, and provided by Coursera via this link.
- The time period for this data is from 2004 - 2020 and
- the dataset contains information on 194,673 car related accidents in the state of Seattle.
- 37 different attributes including:
  - severity
  - location of the collision
  - collision type
  - date and time of the accident
  - the type of junction where the collision took place
  - weather conditions, road conditions, and light conditions

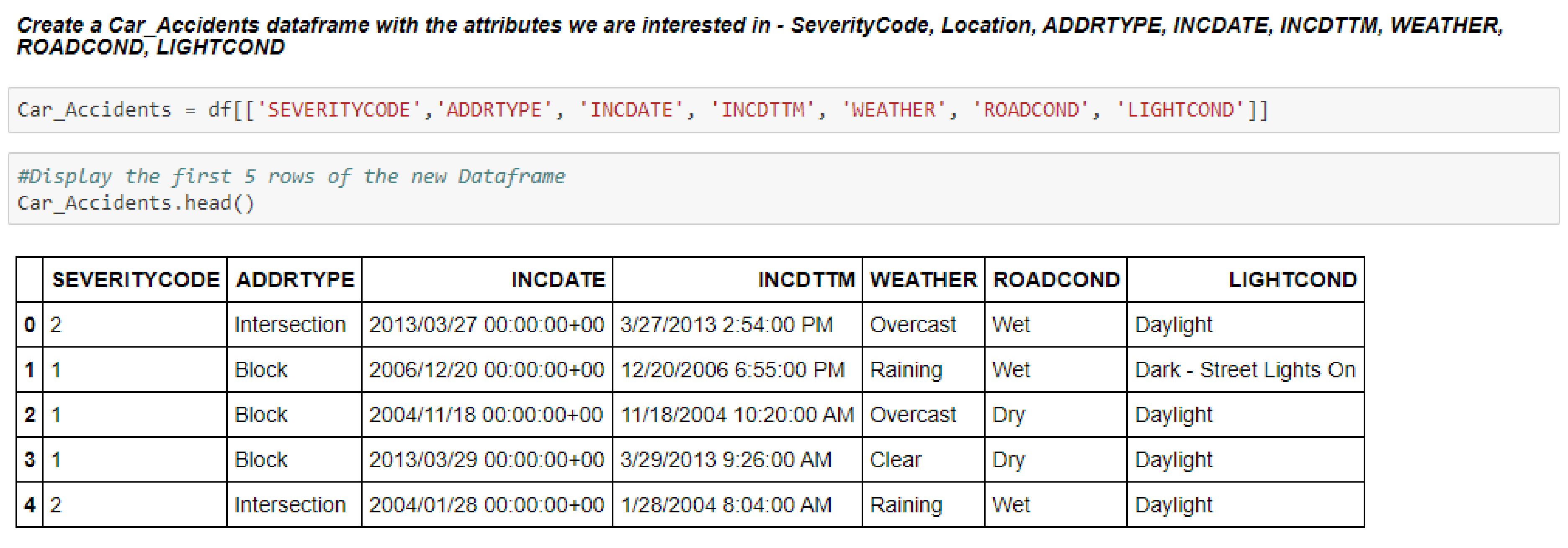
Data (cont’d)

- The target attribute is severity and the Seattle Police Department records accident severity according to the following schema:
  - 0: Unknown/no data
  - 1: Property damage only
  - 2: Minor injury collision
  - 2b: Major injury collision
  - 3: Fatality collision

Data (cont'd)

- This dataset only had entries for two levels – 1 (property damage only) and 2 (minor collision only).
- The below attributes were selected as the independent variables:
  - Collision Address Type (whether alley, block or intersection)
  - Day of the Incident
  - Time of the Incident
  - Weather Conditions
  - Road Conditions
  - Light Conditions
- More about the metadata can be found here.

Data Preprocessing



Create a Car_Accidents dataframe with the attributes we are interested in - SeverityCode, Location, ADDRTYPE, INCDATE, INCDTTM, WEATHER, ROADCOND, LIGHTCOND

```
Car_Accidents = df[['SEVERITYCODE','ADDRTYPE', 'INCDATE', 'INCDTTM', 'WEATHER', 'ROADCOND', 'LIGHTCOND']]
```

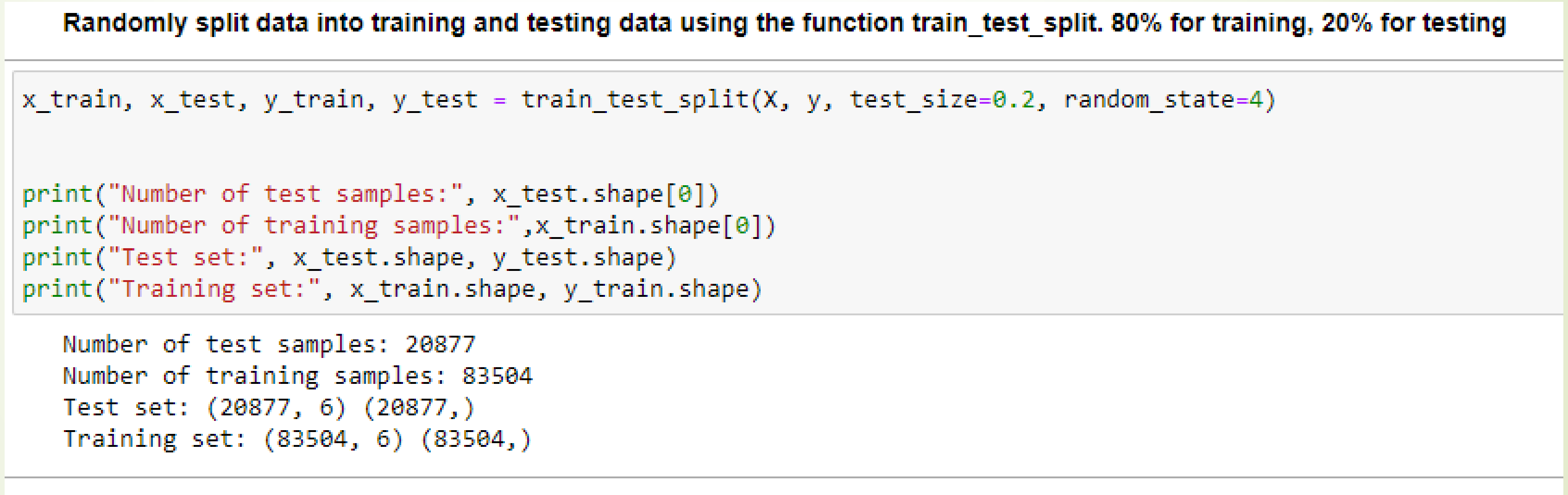
```
#Display the first 5 rows of the new Dataframe
Car_Accidents.head()
```

	SEVERITYCODE	ADDRTYPE	INCDATE	INCDTTM	WEATHER	ROADCOND	LIGHTCOND
0	2	Intersection	2013/03/27 00:00:00+00	3/27/2013 2:54:00 PM	Overcast	Wet	Daylight
1	1	Block	2006/12/20 00:00:00+00	12/20/2006 6:55:00 PM	Raining	Wet	Dark - Street Lights On
2	1	Block	2004/11/18 00:00:00+00	11/18/2004 10:20:00 AM	Overcast	Dry	Daylight
3	1	Block	2013/03/29 00:00:00+00	3/29/2013 9:26:00 AM	Clear	Dry	Daylight
4	2	Intersection	2004/01/28 00:00:00+00	1/28/2004 8:04:00 AM	Raining	Wet	Daylight

Data Preprocessing (cont'd)

- Rows with missing values dropped
- Incident date converted to Day of the Week
- Incident time converted to Hour of Day
- All attributes converted to int data type

Methodology



Randomly split data into training and testing data using the function train_test_split. 80% for training, 20% for testing

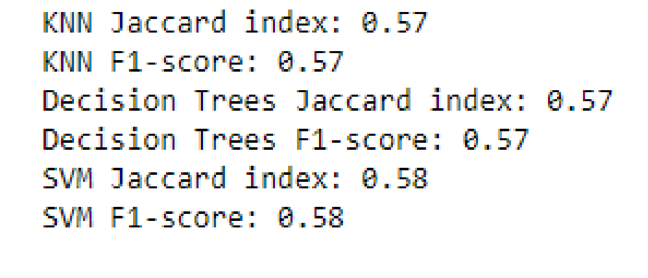
```
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=4)


print("Number of test samples:", x_test.shape[0])
print("Number of training samples:",x_train.shape[0])
print("Test set:", x_test.shape, y_test.shape)
print("Training set:", x_train.shape, y_train.shape)
```

```
Number of test samples: 20877
Number of training samples: 83504
Test set: (20877, 6) (20877,)
Training set: (83504, 6) (83504,)
```

Methodology (cont'd)

- Three algorithms were trained on the pre-processed dataset and their accuracies compared:
  - K Nearest Neighbor(KNN), Decision Trees and Support Vector Machine (SVM)

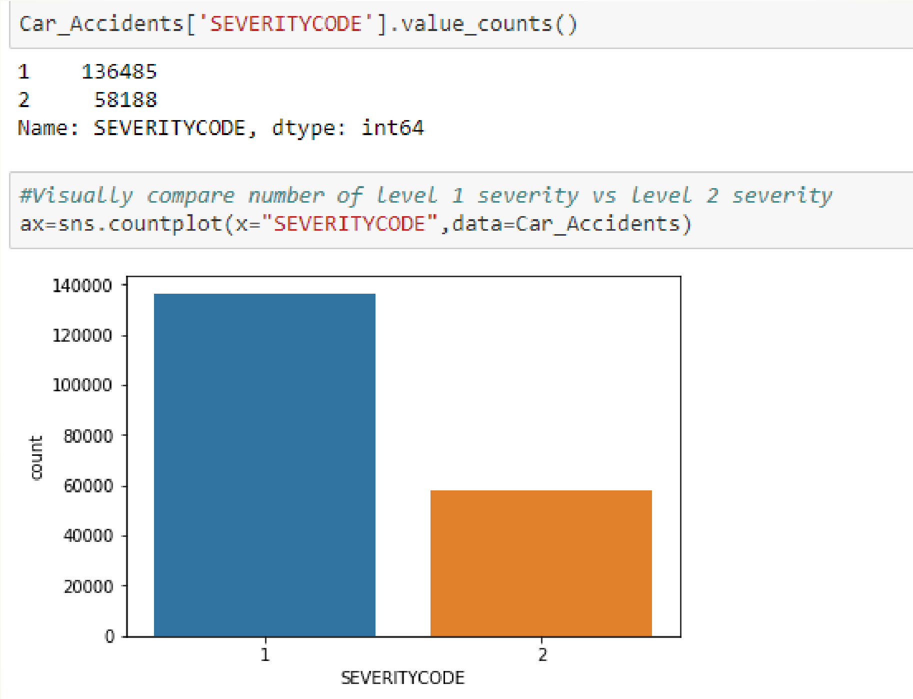
- Jaccard Index and F1-score comparison



```
KNN Jaccard index: 0.57
KNN F1-score: 0.57
Decision Trees Jaccard index: 0.57
Decision Trees F1-score: 0.57
SVM Jaccard index: 0.58
SVM F1-score: 0.58
```

Results

- There were twice as much property damage vs minor injuries

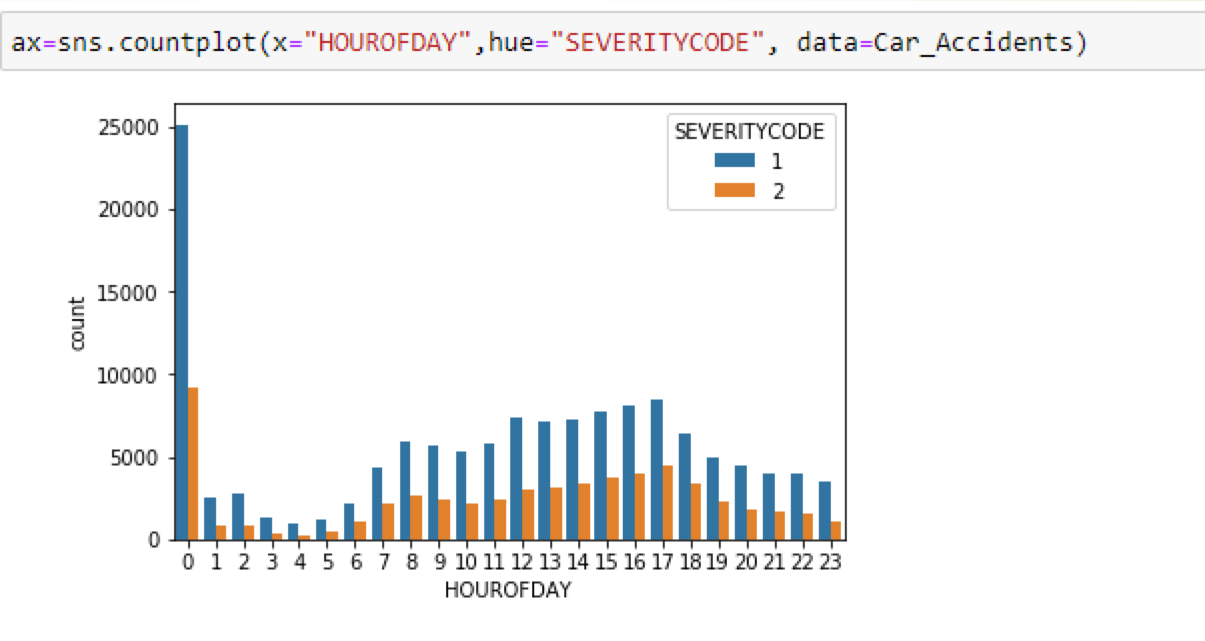


```
Car_Accidents['SEVERITYCODE'].value_counts()
```

```
1    136485
2     58188
Name: SEVERITYCODE, dtype: int64
```

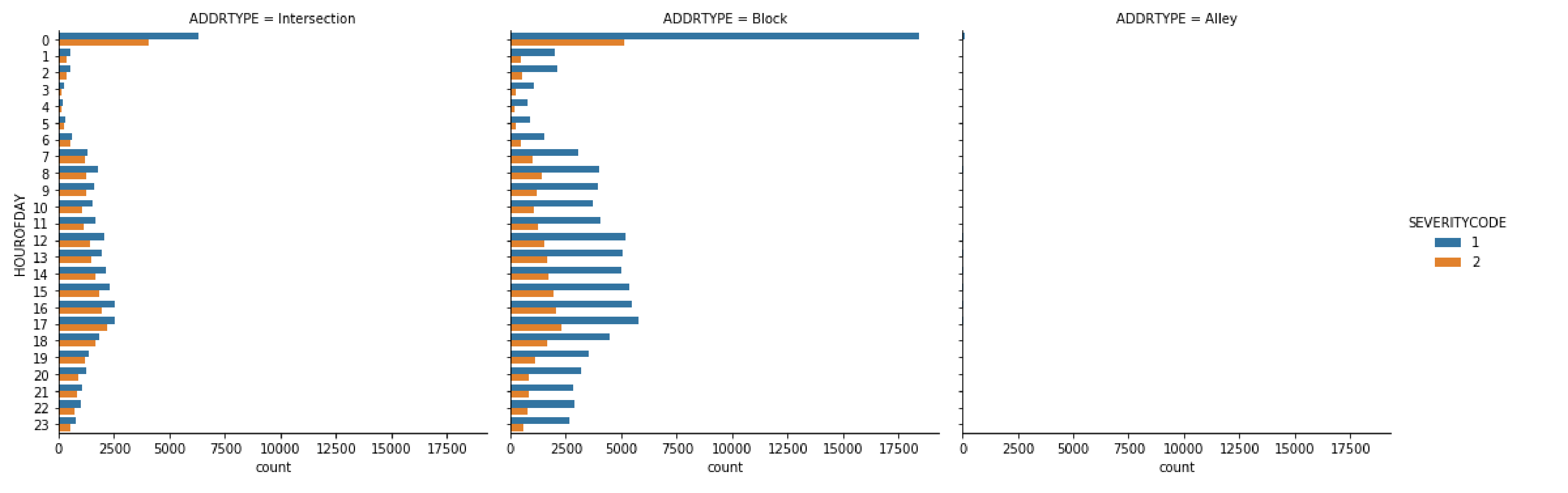
```
#Visually compare number of level 1 severity vs level 2 severity
ax=sns.countplot(x="SEVERITYCODE",data=Car_Accidents)
```

Most accidents occur around midnight. Both level 1 and 2 are at their highest at this time

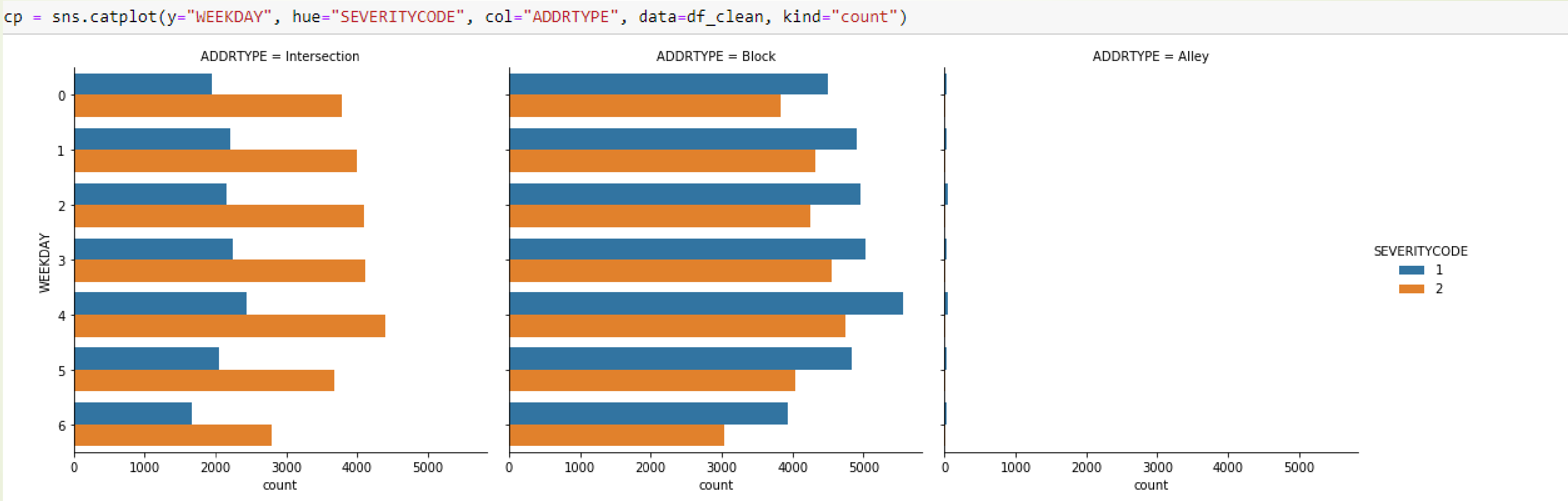


```
ax=sns.countplot(x="HOUROFDAY",hue="SEVERITYCODE", data=Car_Accidents)
```

Most property damage collisions occur at Blocks at around midnight. The number of minor injuries is almost equal at intersections and blocks at midnight.



The day of the week doesn’t seem to impact the number of accidents, but there are more minor injuries at intersections vs property damage at intersections



```
cp = sns.catplot(y="WEEKDAY", hue="SEVERITYCODE", col="ADDRTYPE", data=df_clean, kind="count")
```

Discussion

- The algorithms used above gave accuracy scores of either 0.57 or 0.58, meaning that these models can predict the severity code of an accident with an accuracy of 57% - 58%.

Conclusion

- The accuracy of the classifiers is not great, the highest being 58%. This usually means that the model is under fitted or needs to be trained in more data. Though the dataset used had a variety of attributes, most were not used due to the large amounts of missing data. This may have resulted in important correlation being overlooked.
- With more data this could be a useful tool in predicting car accident severities.