Car accident severity solution

Introduction

- Collisions in junctions can have two consequences: property damage or/and injury Collisions in junctions cause long delay.
- Increase traffic congestion.
- Incurs costs that can be avoided.
- Simple measures can reduce collision risks.
- Exploratory data analysis and predictive modelling can help provide insight to involved parties who can implement safety measures to reduce risk of a collision

Data Wrangling

- Use given data by the course on road collisions for Seattle City
- Read data into correct format and clean accordingly
- Remove duplicate or columns with same elements but which are either in categorical or numerical variables
- Remove the Null values
- Use get dummies for conversion.

```
#Checking The Null values
df.isnull().sum()
```

```
WEATHER          5081
ROADCOND         5012
JUNCTIONTYPE     6329
VEHCOUNT            0
PERSONCOUNT         0
SEVERITYDESC        0
ADDRTYPE         1926
SDOT_COLDESC        0
LIGHTCOND        5170
dtype: int64
```

```
#verifying  agian The Null values
df.isnull().sum()
```

```
WEATHER          0
ROADCOND         0
JUNCTIONTYPE     0
VEHCOUNT         0
PERSONCOUNT      0
SEVERITYDESC     0
ADDRTYPE         0
SDOT_COLDESC     0
LIGHTCOND        0
dtype: int64
```

Encoding categorical values

```
Feature=pd.get_dummies(df[['WEATHER','ROADCOND','JUNCTIONTYPE','VEHCOUNT','PERSONCOUNT','ADDRTYPE', 'SU

Feature.head()
```

	VEHCOUNT	PERSONCOUNT	WEATHER_Blowing Sand/Dirt	WEATHER_Clear	WEATHER_Fog/Smog/Smoke	WEATHER_Other	WEATHE
0	2	2	0	0	0	0	
1	2	2	0	0	0	0	
2	3	4	0	0	0	0	
3	3	3	0	1	0	0	
4	2	2	0	0	0	0	

5 rows × 80 columns

Here, The categorical values are encoded to numericakl using get dummies.

Model Development and Evaluation

- Import libraries and modules to perform
- Train_test_split
- Random Forest Classifier
- Metrics

Random Forest Model

```
from sklearn.ensemble import RandomForestClassifier

#Modelling
model=RandomForestClassifier(n_estimators=100)
model.fit(x_train,y_train)
```

```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                       max_depth=None, max_features='auto', max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=100,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

```
#Prediction
y_pred=model.predict(x_test)

print(y_test[0:5])
print(y_pred[0:5])
```

```
['Property Damage Only Collision' 'Property Damage Only Collision'
 'Property Damage Only Collision' 'Property Damage Only Collision'
 'Property Damage Only Collision']
['Property Damage Only Collision' 'Injury Collision'
 'Property Damage Only Collision' 'Property Damage Only Collision'
 'Property Damage Only Collision']
```

Evaluation Scores

```
scores= {"accuracy score": ac,"jaccard similarity score": jc,"f1 score": fs}
df = pd.DataFrame(scores,index=['Random Forest'])
df
```

	accuracy score	jaccard similarity score	f1 score
Random Forest	74.583823	74.583823	70.812256

Conclusion

- Drop missing values if the range is between 5-10% of the total rows.
- Work with categorical attributes using get dummiesto solve the issue
- Use Random Forest technique for classification model
- This project aimed at exploring the data to provide insight in severity levels for road collisions at junctions. The predictive model would be useful to help local authorities decide on whether to implement new safety measures in certain areas