IBM Data Science Project

To predict traffic accident severity by machine learning

ZHANG HAIBIN

18-OCT-2020

1. PROJECT INTRODUCTION

1) Background

Millions of people were involved traffic accident every year. And to reduce traffic accidents and reduce the severity of car accidents is an very important public concern and safety challenge.

2) Problem

There are different major factors for different accidents, to reveal the accidents commonality and factors relationship, and develop good model to predict severity of traffic accident is the focus.

3) Interest

Transportation department of government, traffic designers, police and respective drivers of vehicles should be interested at this kind of topic.

2. DATA ACQUISITION AND CLEANING

2.1 Data Source

Data provided by the Seattle Department of Transportation (SDOT) , can be downloaded as below link: https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Data-Collisions.csv

This dataset has details about 194k accident details. Each accident is defined by 38 different attributes (features).

The metadata of the features are given as below link:

https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Metadata.pdf

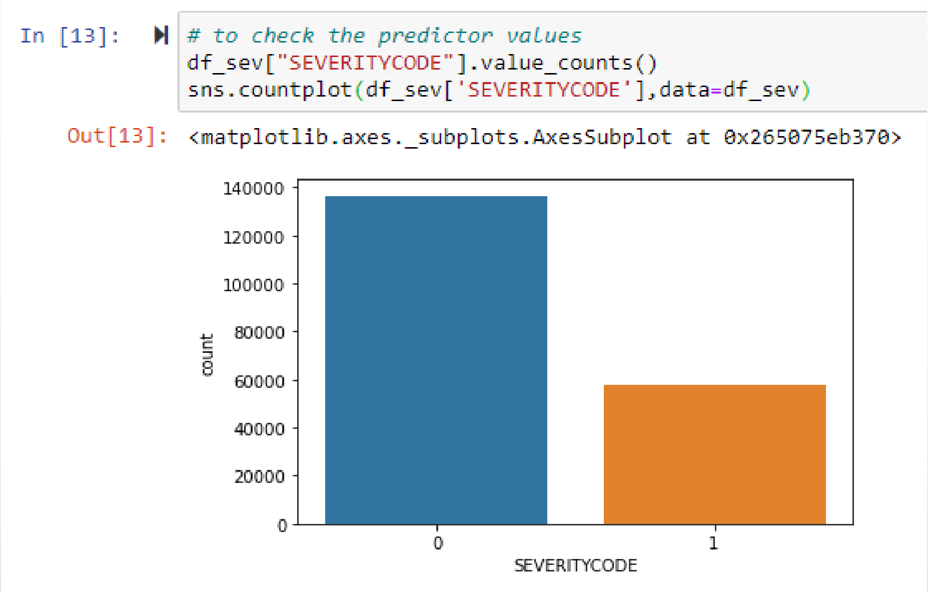
2.2 Feature Selection and drop

- ❖ SEVERITYCODE is the predictor
- ❖ ‘ADDRTYPE’, JUNCTIONTYPE’, ‘WEATHER’, ‘ROADCOND’ and ‘LIGHTCOND’. And for exploratory data analysis

2.3 Data handling for the selected features

3. EXPLORATORY DATA ANALYSIS(EDA)

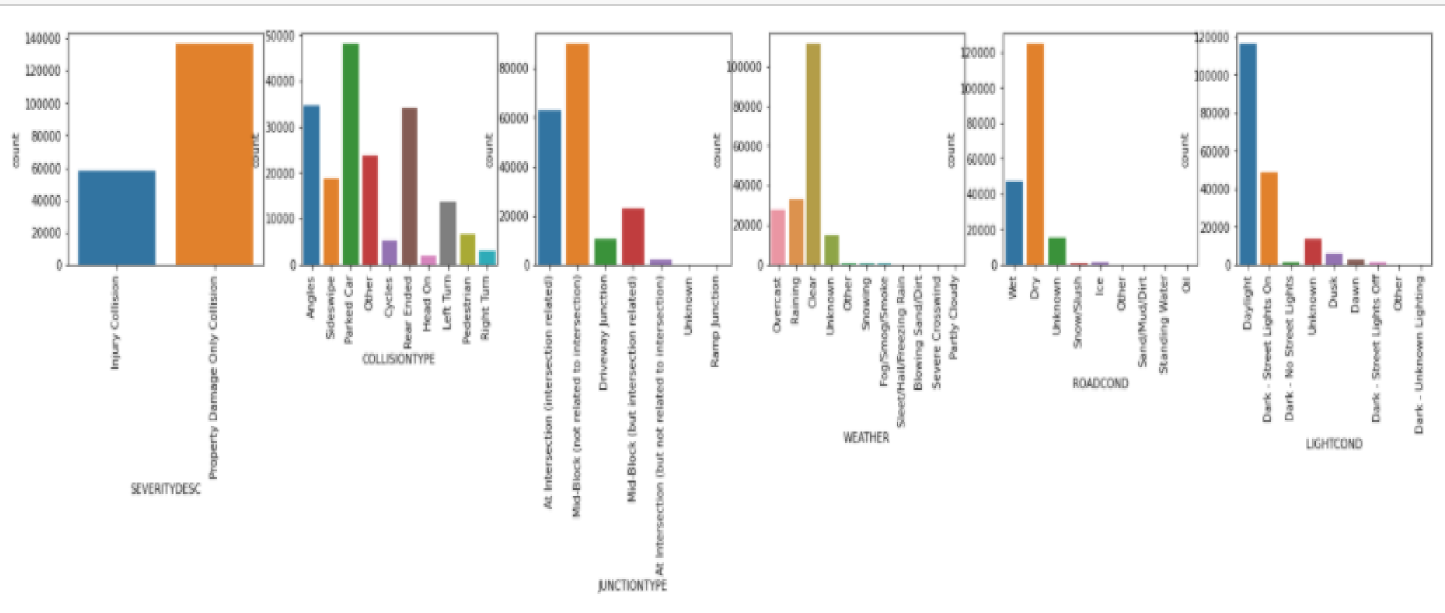
3.1 Target variable



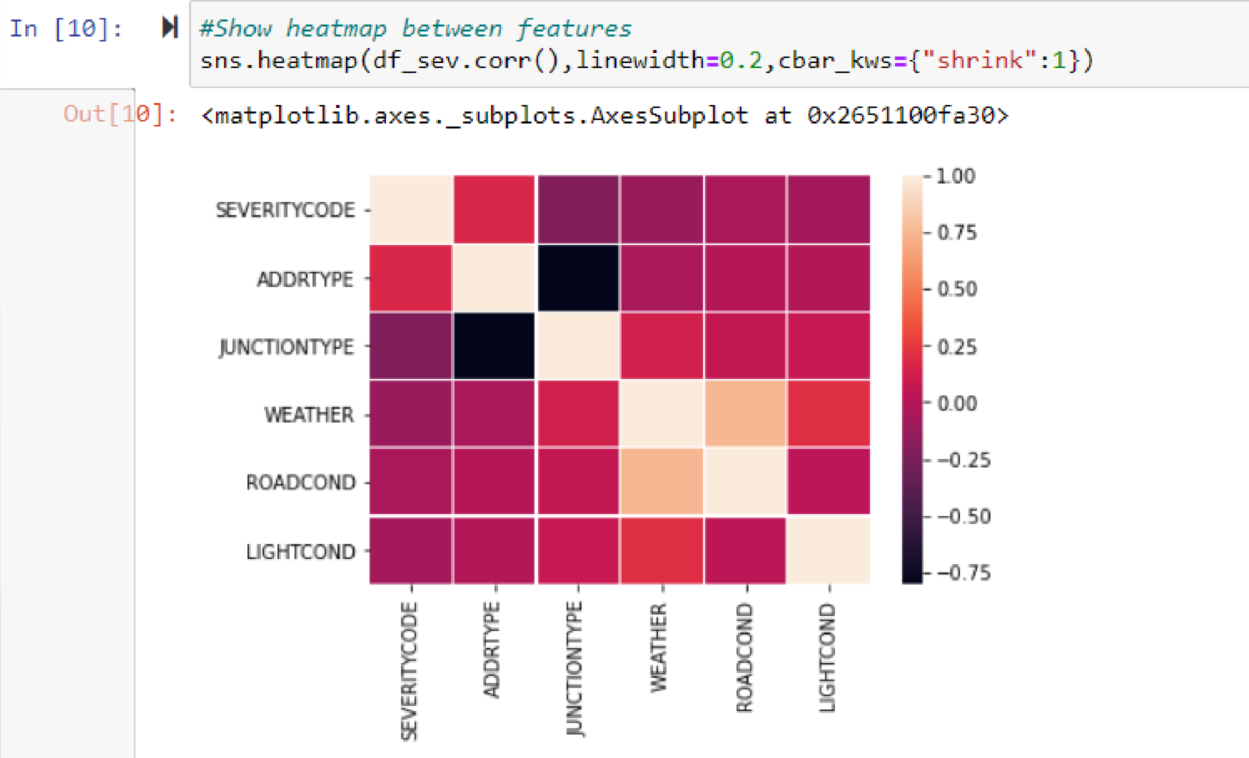
```
In [13]:  # to check the predictor values
          df_sev["SEVERITYCODE"].value_counts()
          sns.countplot(df_sev['SEVERITYCODE'],data=df_sev)

Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x265075eb370>
```

Bar chart to check and show values of target variable

3.2 Categorical variables analysis



3.3 Relationship between “SEVERITYCODE” and selected features

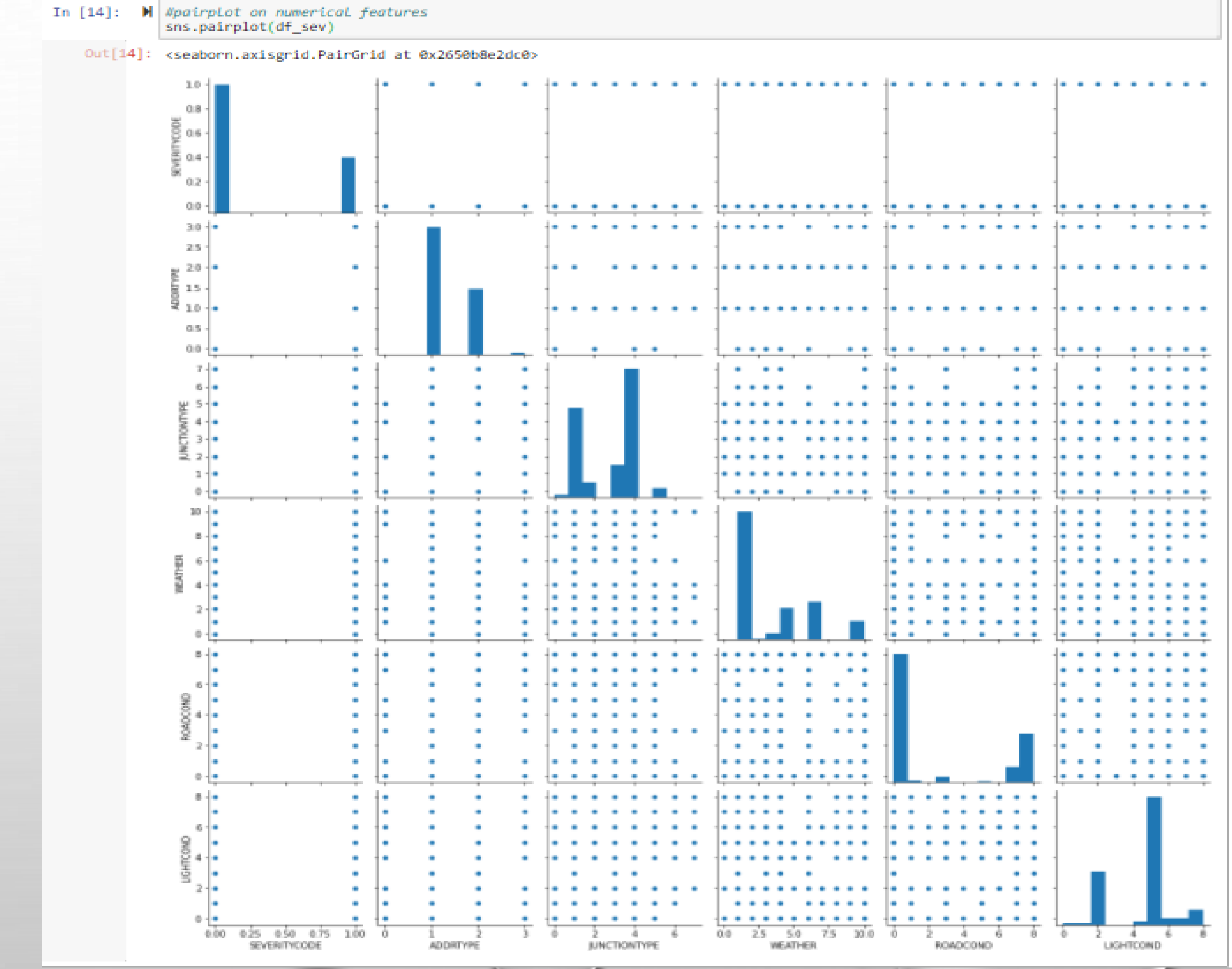


```
In [10]:  #Show heatmap between features
          sns.heatmap(df_sev.corr(),linewidth=0.2,cbar_kws={"shrink":1})
```

```
Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x2651100fa30>
```

To get the overall picture for better understanding the relationship among target variable and the selected features

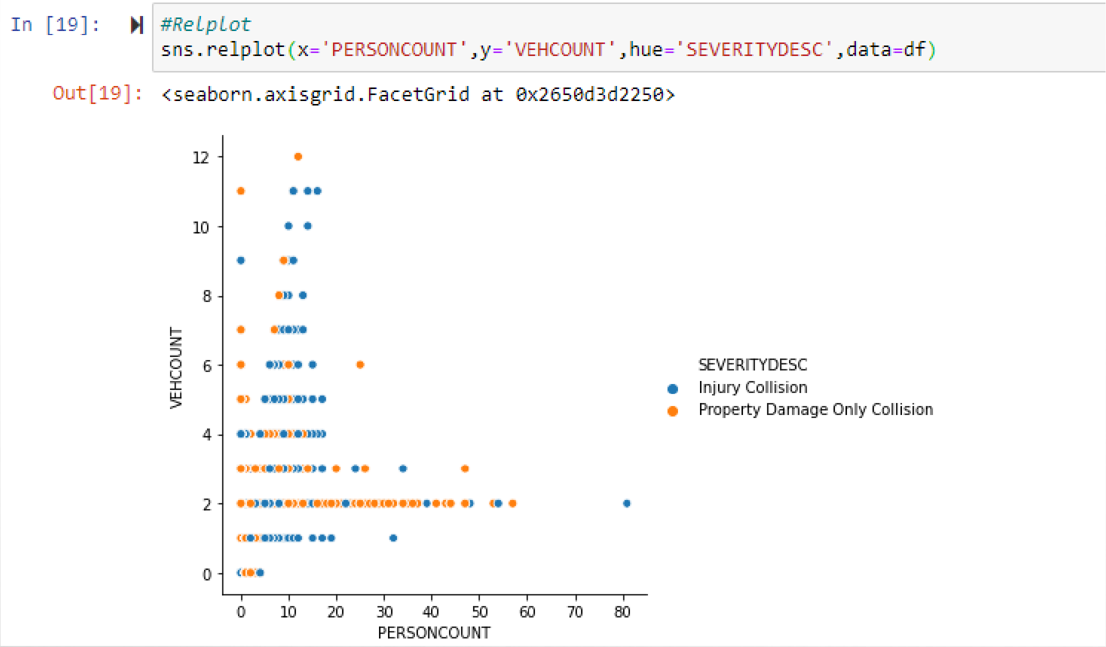
3.4 PairPlot on numerical variables



```
In [14]:  #pairplot on numerical features
          sns.pairplot(df_sev)
```

```
Out[14]: <seaborn.axisgrid.PairGrid at 0x2650b8e2dc0>
```

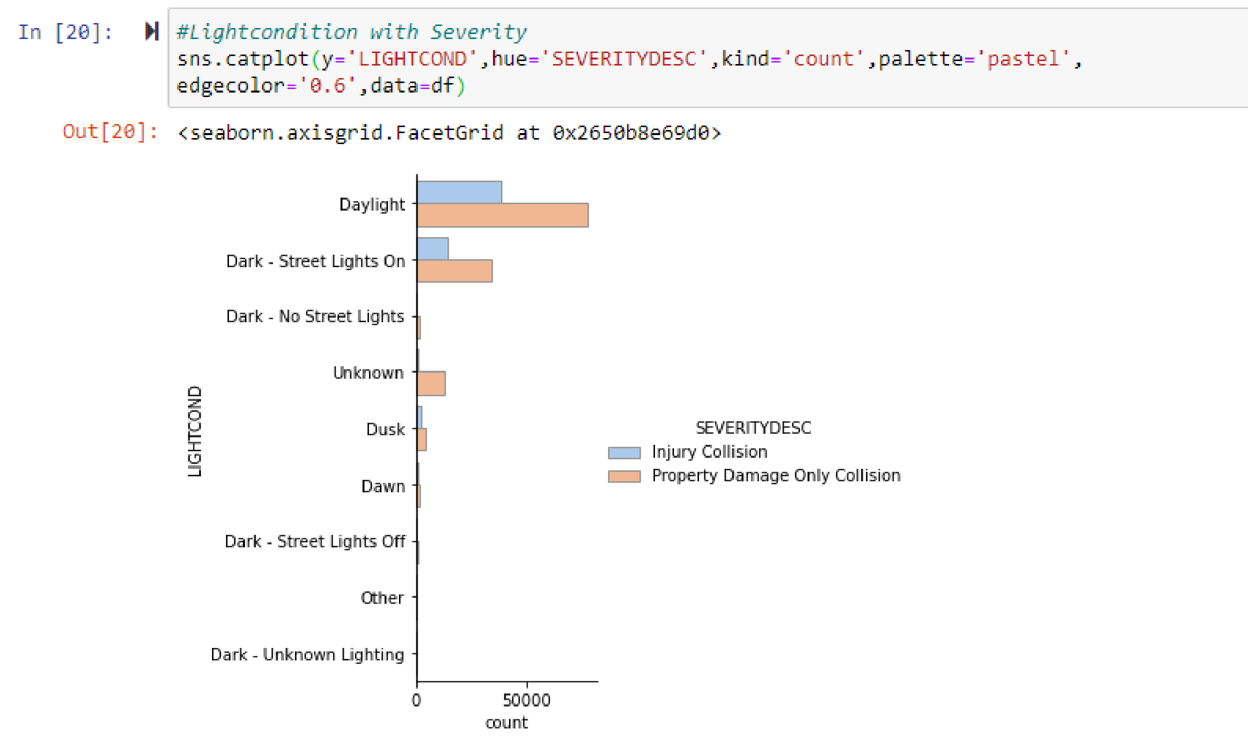
3.4.1 Relationship between VEHCOUNT, PERSONCOUNT based on SEVERITY



```
In [19]:  #Relplot
          sns.relplot(x='PERSONCOUNT',y='VEHCOUNT',hue='SEVERITYDESC',data=df)

Out[19]: <seaborn.axisgrid.FacetGrid at 0x2650d3d2250>
```

Strong correlation between vehicle count and person count

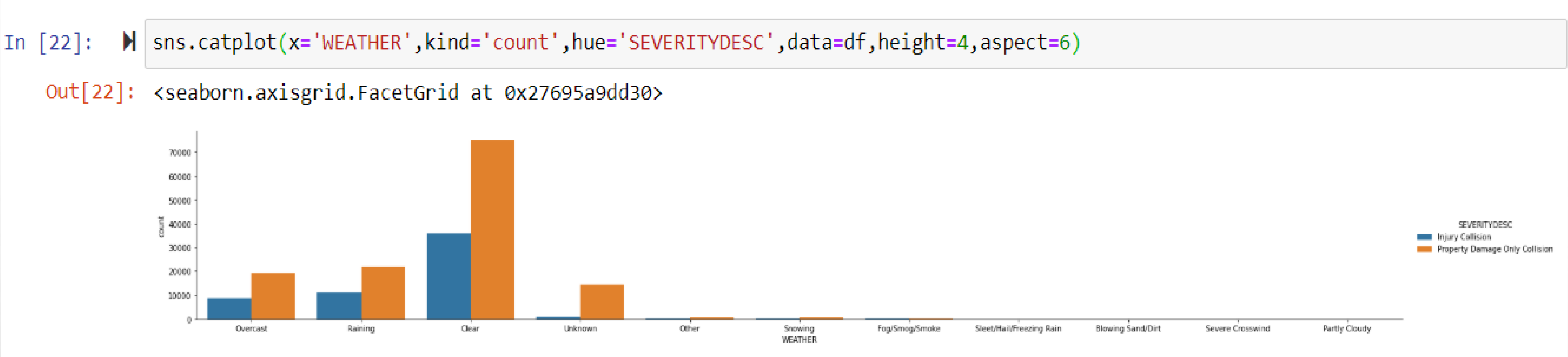
3.4.2 Relationship between LIGHTCOND by SEVERITY of Accidents



```
In [20]:  #Lightcondition with Severity
          sns.catplot(y='LIGHTCOND',hue='SEVERITYDESC',kind='count',palette='pastel',
          edgecolor='0.6',data=df)

Out[20]: <seaborn.axisgrid.FacetGrid at 0x2650b8e69d0>
```

More accidents happened happened at daylight

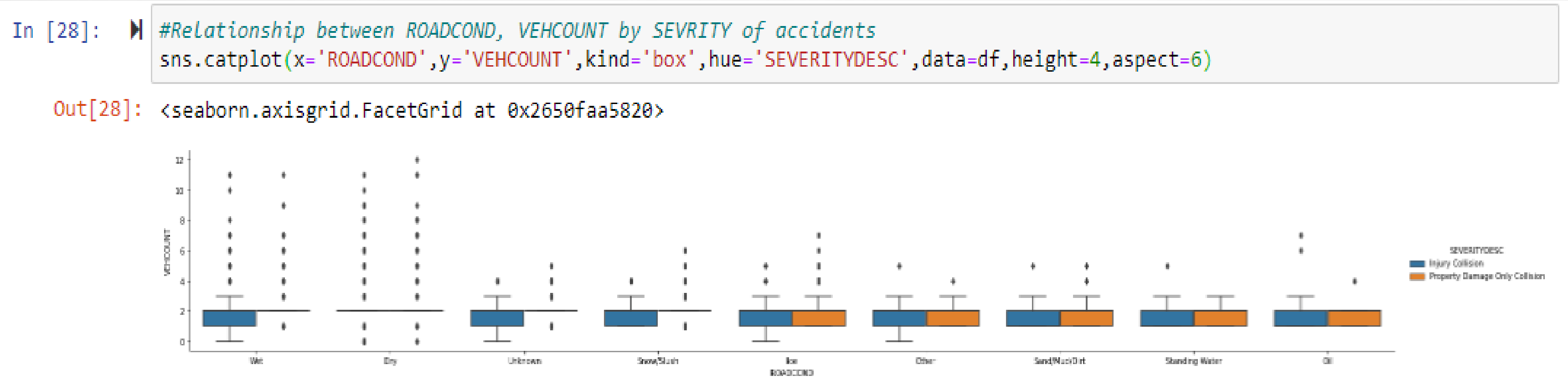
3.4.3 Relationship between WEATHER by SEVERITY of accidents



```
In [22]:  sns.catplot(x='WEATHER',kind='count',hue='SEVERITYDESC',data=df,height=4,aspect=6)

  Out[22]: <seaborn.axisgrid.FacetGrid at 0x27695a9dd30>
```

A lot of accidents happened at clear weather

3.4.4 Relationship between ROADCOND, VEHCOUNT by SEVRITY of accidents

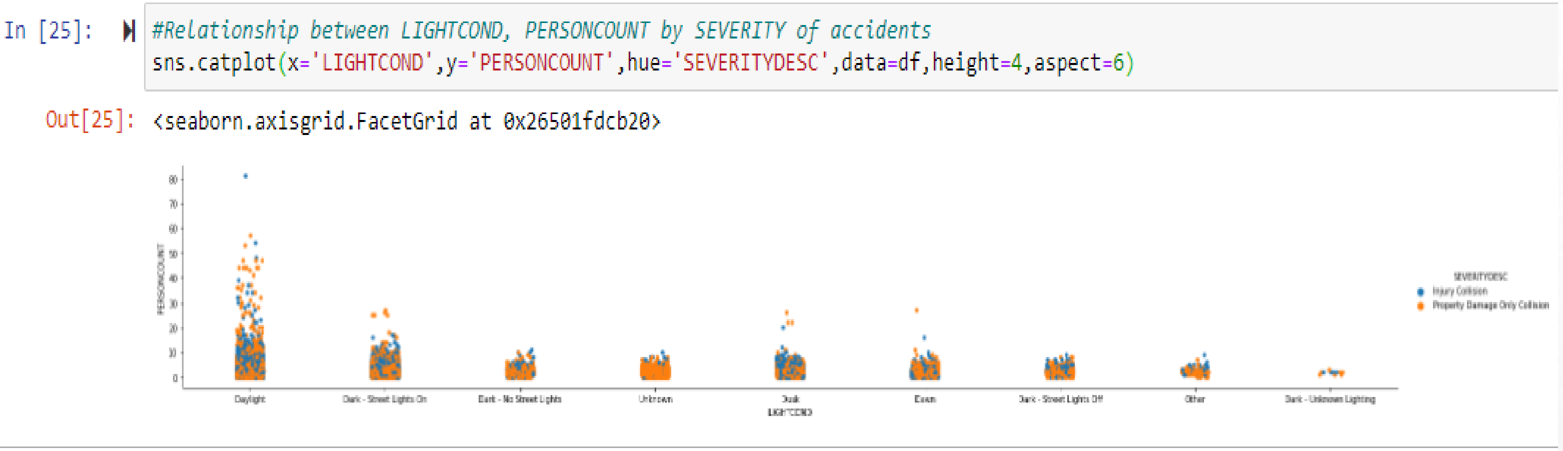


```
In [28]:  #Relationship between ROADCOND, VEHCOUNT by SEVRITY of accidents
          sns.catplot(x='ROADCOND',y='VEHCOUNT',kind='box',hue='SEVERITYDESC',data=df,height=4,aspect=6)

  Out[28]: <seaborn.axisgrid.FacetGrid at 0x2650faa5820>
```

The abnormal road condition may

cause sever accidents.

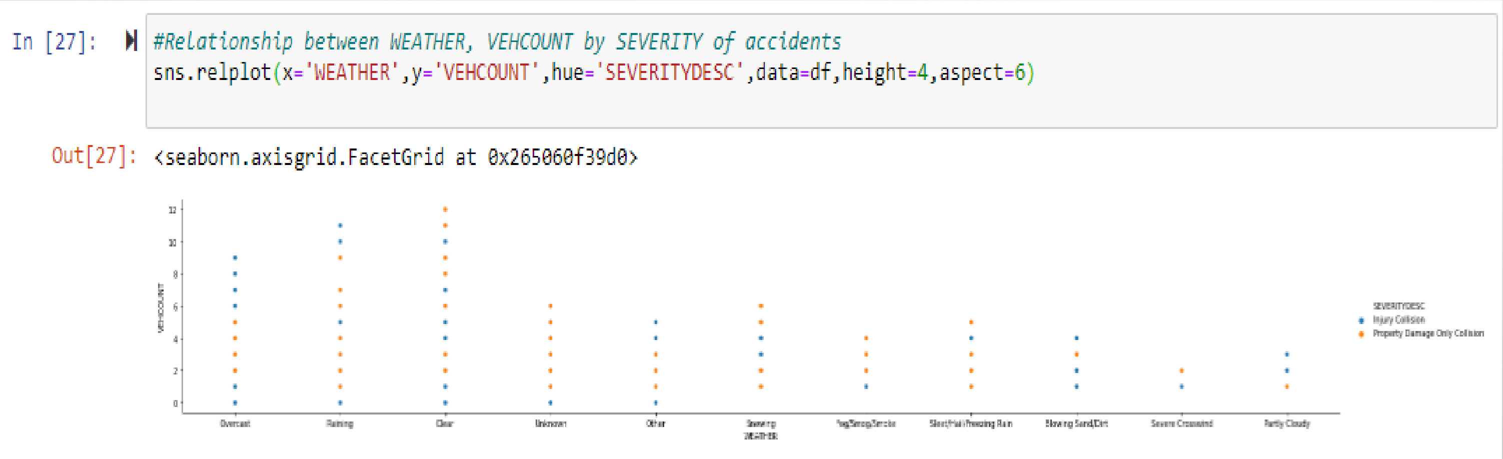
3.4.5 Relationship between LIGHTCOND, PERSONCOUNT by SEVERITY of accidents



```
In [25]:  #Relationship between LIGHTCOND, PERSONCOUNT by SEVERITY of accidents
          sns.catplot(x='LIGHTCOND',y='PERSONCOUNT',hue='SEVERITYDESC',data=df,height=4,aspect=6)

Out[25]: <seaborn.axisgrid.FacetGrid at 0x26501fdcb20>
```

More people involved in daylight accident

3.4.6 Relationship between WEATHER, VEHCOUNT by SEVERITY of accidents



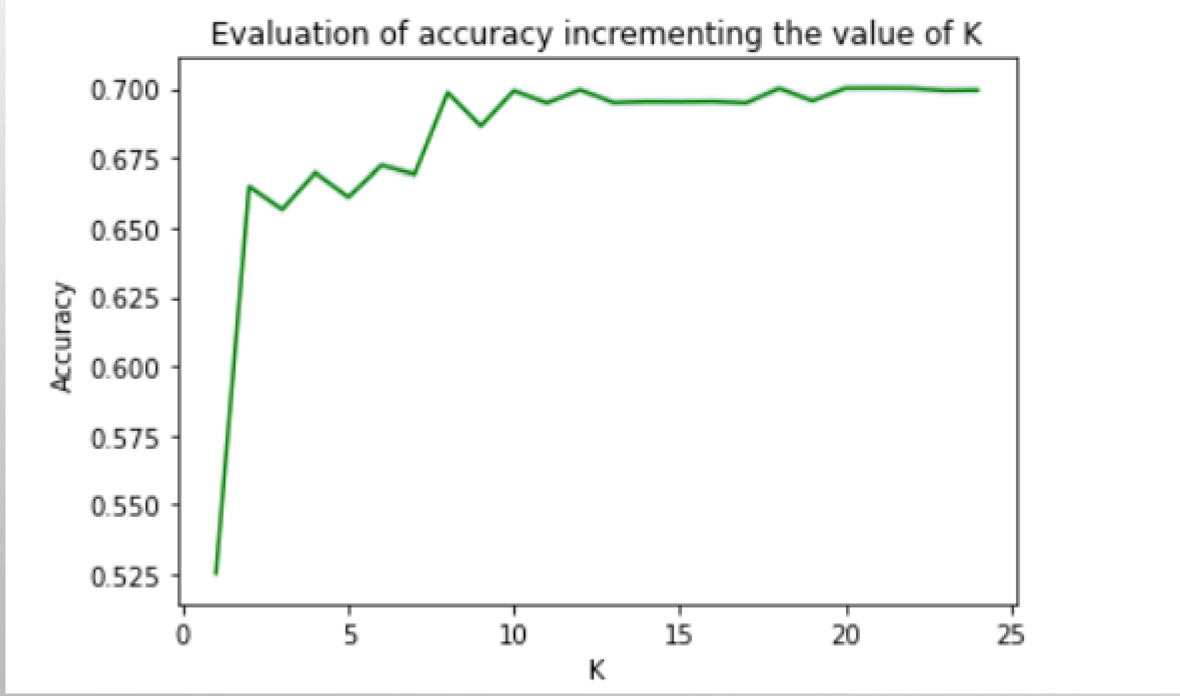
```
In [27]:  #Relationship between WEATHER, VEHCOUNT by SEVERITY of accidents
          sns.relplot(x='WEATHER',y='VEHCOUNT',hue='SEVERITYDESC',data=df,height=4,aspect=6)

Out[27]: <seaborn.axisgrid.FacetGrid at 0x265060f39d0>
```

Clear weather and raining days has high risk to lead traffic accident

4.Predictive Modeling

Three machining learning :K Nearest Neighbor, Decision tree, Logistic Regression to be utilized for prediction

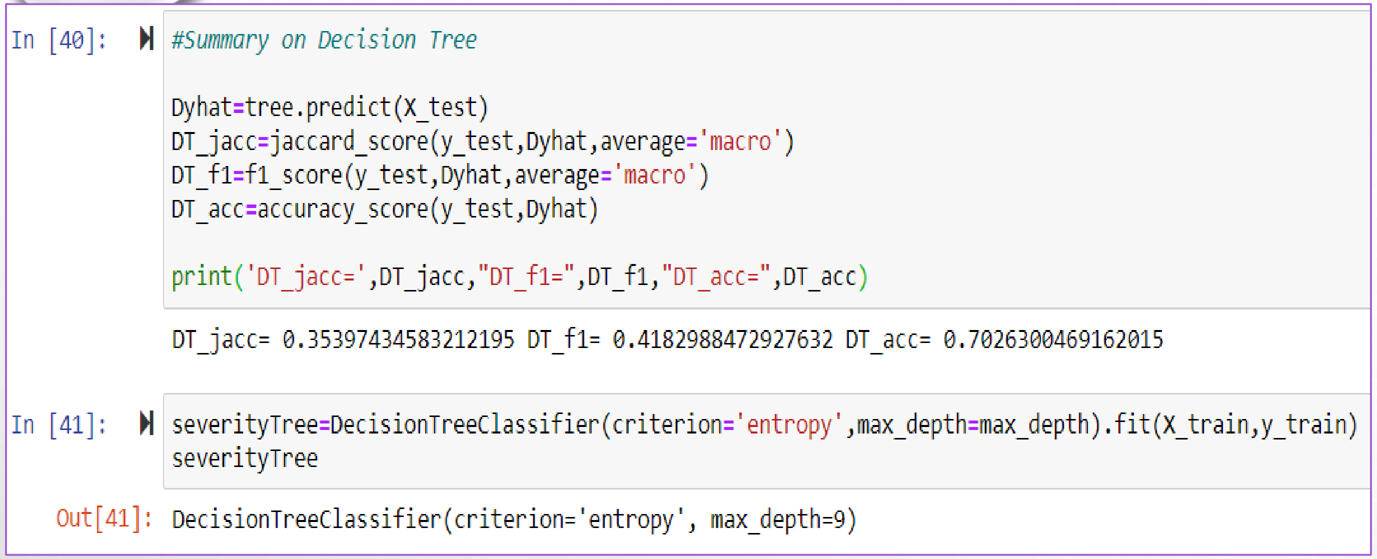
4.1 KNN (K Nearest Neighbor)



KNN_jacc=
0.37191246702069936
KNN_f1=
0.45670261566621373
KNN_acc=
0.6995137152837232

The best accuracy when K=21

4.2 Decision Tree



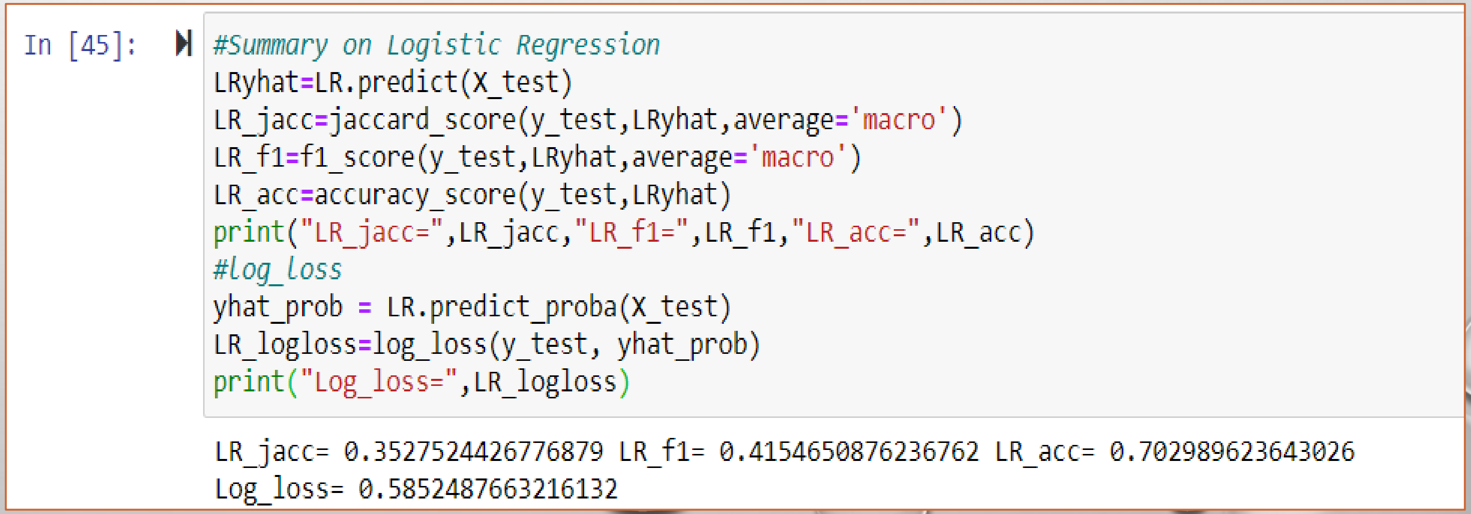
In [40]:
```
#Summary on Decision Tree

Dyhat=tree.predict(X_test)
DT_jacc=jaccard_score(y_test,Dyhat,average='macro')
DT_f1=f1_score(y_test,Dyhat,average='macro')
DT_acc=accuracy_score(y_test,Dyhat)

print('DT_jacc=',DT_jacc,"DT_f1=",DT_f1,"DT_acc=",DT_acc)
```

```
DT_jacc= 0.35397434583212195 DT_f1= 0.4182988472927632 DT_acc= 0.7026300469162015
```

In [41]:
```
severityTree=DecisionTreeClassifier(criterion='entropy',max_depth=max_depth).fit(X_train,y_train)
severityTree
```

Out[41]:
```
DecisionTreeClassifier(criterion='entropy', max_depth=9)
```

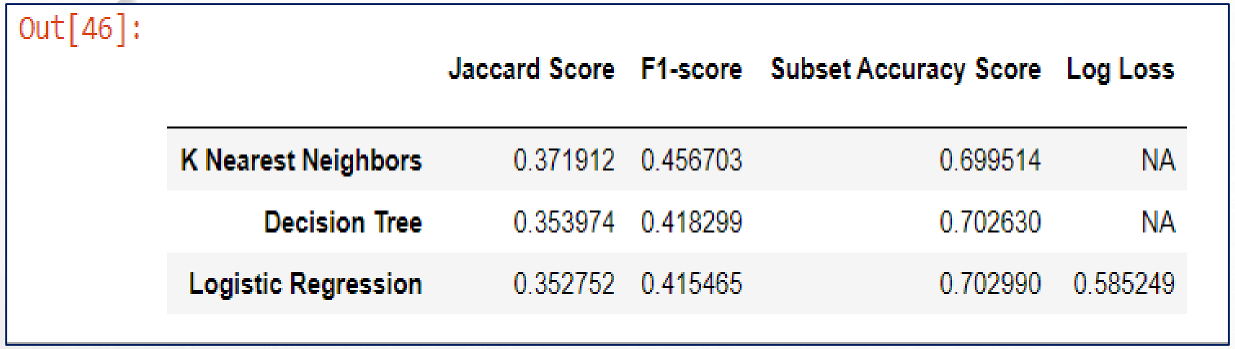
4.3 Logistic Regression



In [45]:
```
#Summary on Logistic Regression
LRyhat=LR.predict(X_test)
LR_jacc=jaccard_score(y_test,LRyhat,average='macro')
LR_f1=f1_score(y_test,LRyhat,average='macro')
LR_acc=accuracy_score(y_test,LRyhat)
print("LR_jacc=",LR_jacc,"LR_f1=",LR_f1,"LR_acc=",LR_acc)
#log_loss
yhat_prob = LR.predict_proba(X_test)
LR_logloss=log_loss(y_test, yhat_prob)
print("Log_loss=",LR_logloss)
```

```
LR_jacc= 0.3527524426776879 LR_f1= 0.4154650876236762 LR_acc= 0.702989623643026
Log_loss= 0.5852487663216132
```

5.Result and Evaluation



Out[46]:

	Jaccard Score	F1-score	Subset Accuracy Score	Log Loss
K Nearest Neighbors	0.371912	0.456703	0.699514	NA
Decision Tree	0.353974	0.418299	0.702630	NA
Logistic Regression	0.352752	0.415465	0.702990	0.585249

Overall,the three models got similar accuracy score. KNN achieved better Jaccard and F1 scores, however, KNN need spent much more time for the computation.

In addition, logistic regression made the most sense because of its binary nature.

6.Conclusion

1) KNN, Decision Tree and Logistic Regression have consistent accuracy to predict the severity of traffic accident
2) Selected features have great impact on severity of traffic accidents based on machine learning result
3) The prediction is meaningful for the public concerned.