CAR ACCIDENT SEVERITY PREDICTION

-GANAPATHI BALAJI

CONTENT

- Problem Definition
- Exploratory Data Analysis
- Feature Engineering
- Model Process
- Model Report
- Summary Conclusion

PROBLEM DEFINITION - BACKGROUND

- World Health Organization (WHO) reported that more than **1.25 million** people die each year while **50 million** are injured as a result of road accidents worldwide

- Road accidents are the **10th** leading cause of death globally

- On current trends, road traffic accidents are to become the **7th** leading cause of death by **2030** making it a major public health concern

- Between the years 2005 and 2016, there were roughly **2 million** road accidents reported in the United Kingdom (UK) alone of which **16,000** were fatal

World Health Organization

PROBLEM DEFINITION - OBJECTIVE

- Used the Traffic Accidents dataset obtained from *Kaggle's* data repository to:
  - **Identify factors** responsible for most of the reported accidents
  - Build a **predictive model** capable of accurately determining the severity of an accident based on a set of input factors
  - Provide recommendations to **Department of Transport** - to improve road safety policies and prevent accident recurrences where possible

EXPLORATORY DATA ANALYSIS

	Unique_count
PEDCYLCOUNT	3
PEDCOUNT	7
VEHCOUNT	13
SDOT_COLCODE	39
PERSONCOUNT	47
SEGLANEKEY	1955
CROSSWALKKEY	2198
INTKEY	7614
X	23563
Y	23839
SDOTCOLNUM	114932
OBJECTID	194673
INCKEY	194673
COLDETKEY	194673

Remove keys as it won't add value

	count	unique	top	freq
SPEEDING	9333	1	Y	9333
EXCEPTRSNDESC	5638	1	Not Enough Information, or Insufficient Locati...	5638
PEDROWNOTGRNT	4667	1	Y	4667
INATTENTIONIND	29805	1	Y	29805
HITPARKEDCAR	194673	2	N	187457
STATUS	194673	2	Matched	189796
EXCEPTRSNCODE	84811	2		79173
SEVERITYDESC	194673	2	Property Damage Only Collision	136485
ADDRTYPE	192747	3	Block	126926
UNDERINFL	189789	4	N	100274
JUNCTIONTYPE	188344	7	Mid-Block (not related to intersection)	89800
ROADCOND	189661	9	Dry	124510
LIGHTCOND	189503	9	Daylight	116137
COLLISIONTYPE	189769	10	Parked Car	47987
WEATHER	189592	11	Clear	111135
SDOT_COLDESC	194673	39	MOTOR VEHICLE STRUCK MOTOR VEHICLE, FRONT END ...	85209
ST_COLDESC	189769	62	One parked--one moving	44421
ST_COLCODE	194655	115	32	27612
INCDATE	194673	5985	2006/11/02 00:00:00+00	96
LOCATION	191996	24102	BATTERY ST TUNNEL NB BETWEEN ALASKAN WY VI NB ...	276
INCDTTM	194673	162058	11/2/2006	96
REPORTNO	194673	194670	1776526	2

Removing these columns as these are constant through out the data and and unkown for most of the observations

Removing description columns as we also have codes

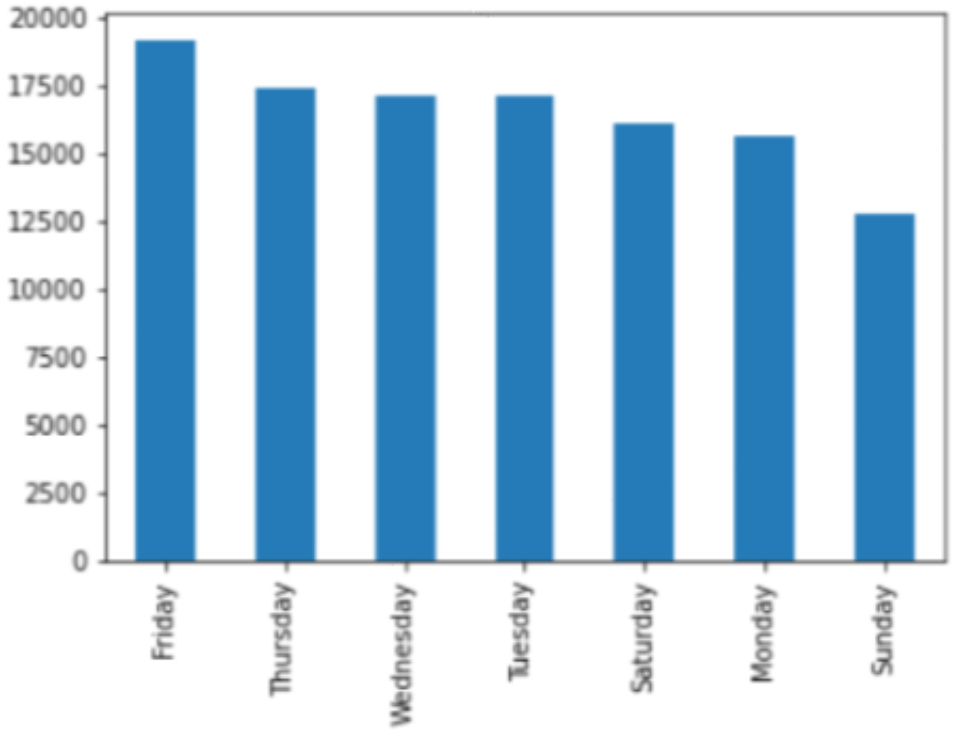
Removing time column as we already have date and LIGHTCOND fields

Removing Key value as it not adding any value

EXPLORATORY DATA ANALYSIS

```
In [180]: df_under_sample['day_of_week'].value_counts().plot.bar()

Out[180]: <matplotlib.axes._subplots.AxesSubplot at 0x2983744c940>
```

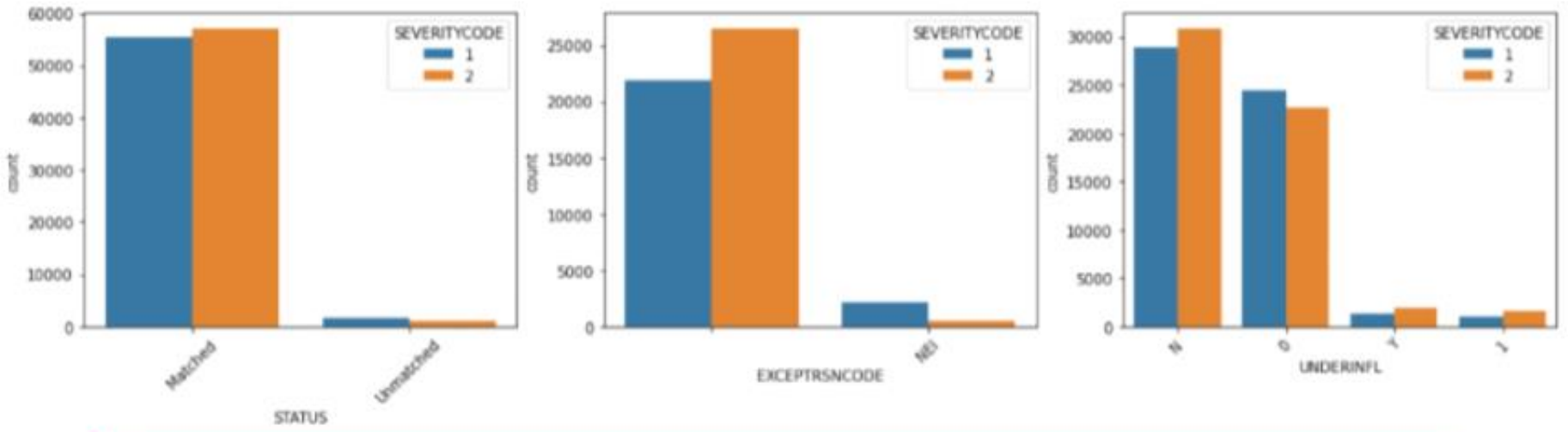


All most all the days are having similar frequency Friday&Sunday has little variation can be ignored as it is not having much impact on verall

EXPLORATORY DATA ANALYSIS

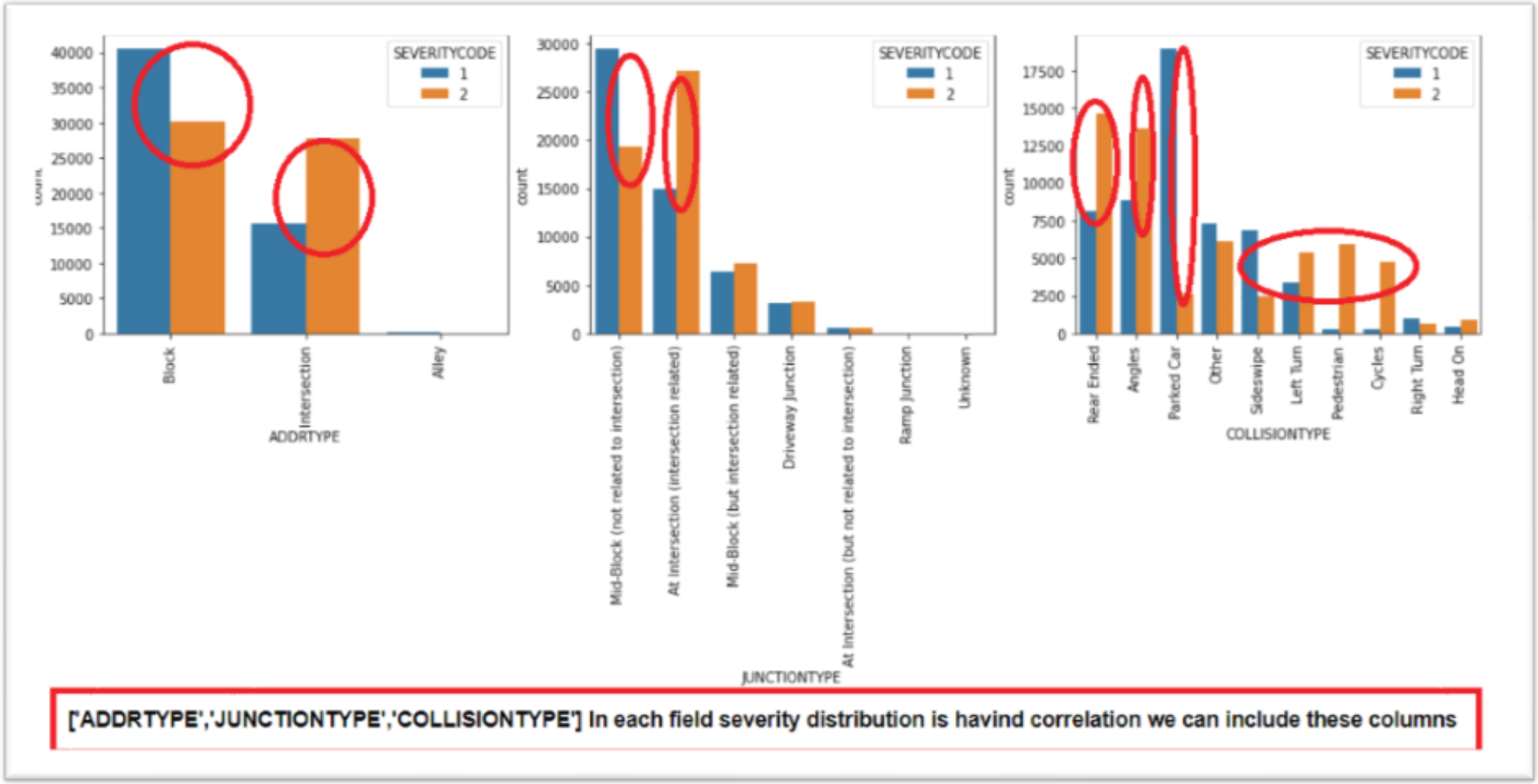
['STATUS','EXCEPTRSNCODE','UNDERINFL']

```
f, ax = plt.subplots(1,3,figsize=(18, 4))
sns.countplot(x='STATUS', hue='SEVERITYCODE',data=df_under_sample,order=df_under_sample['STATUS'].value_counts().index,ax=ax[
sns.countplot(x='EXCEPTRSNCODE', hue='SEVERITYCODE',data=df_under_sample,order=df_under_sample['EXCEPTRSNCODE'].value_counts(
sns.countplot(x='UNDERINFL', hue='SEVERITYCODE',data=df_under_sample,order=df_under_sample['UNDERINFL'].value_counts().index,
ax[0].set_xticklabels(ax[0].get_xticklabels(), rotation=45)
ax[1].set_xticklabels(ax[1].get_xticklabels(), rotation=45)
ax[2].set_xticklabels(ax[2].get_xticklabels(), rotation=45)
f.show()
```



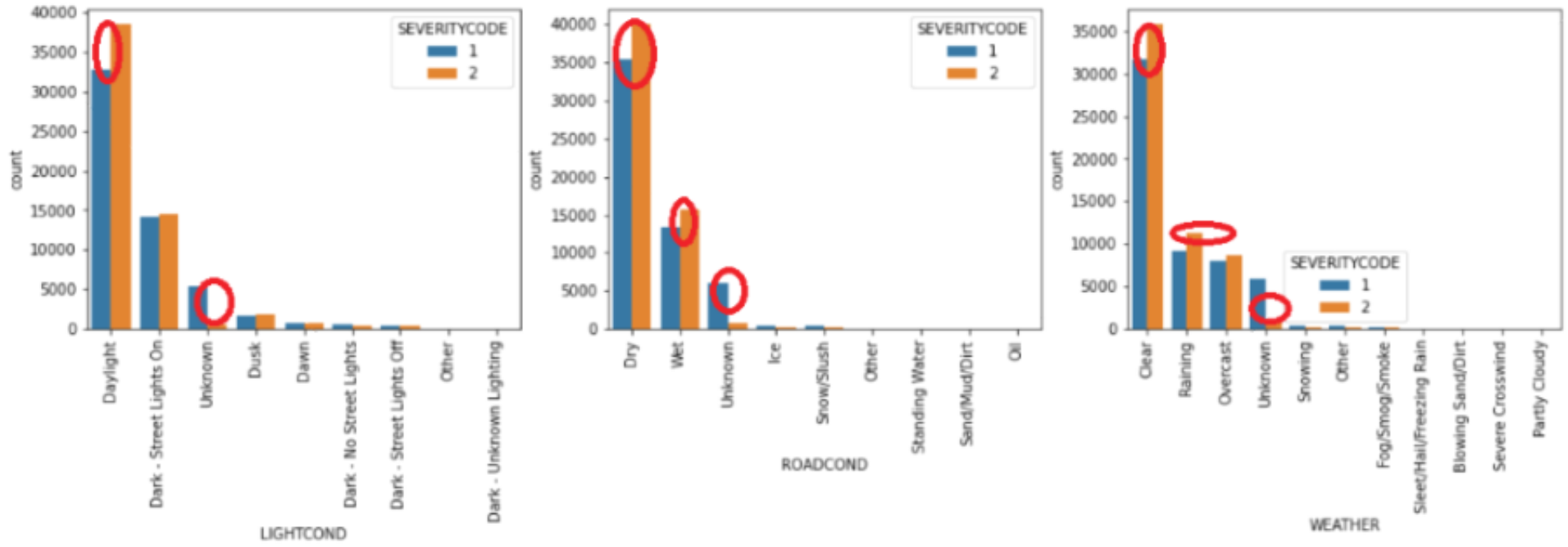
['STATUS','EXCEPTRSNCODE','UNDERINFL'] In each field severity distribution is similar we can ignore these columns

EXPLORATORY DATA ANALYSIS



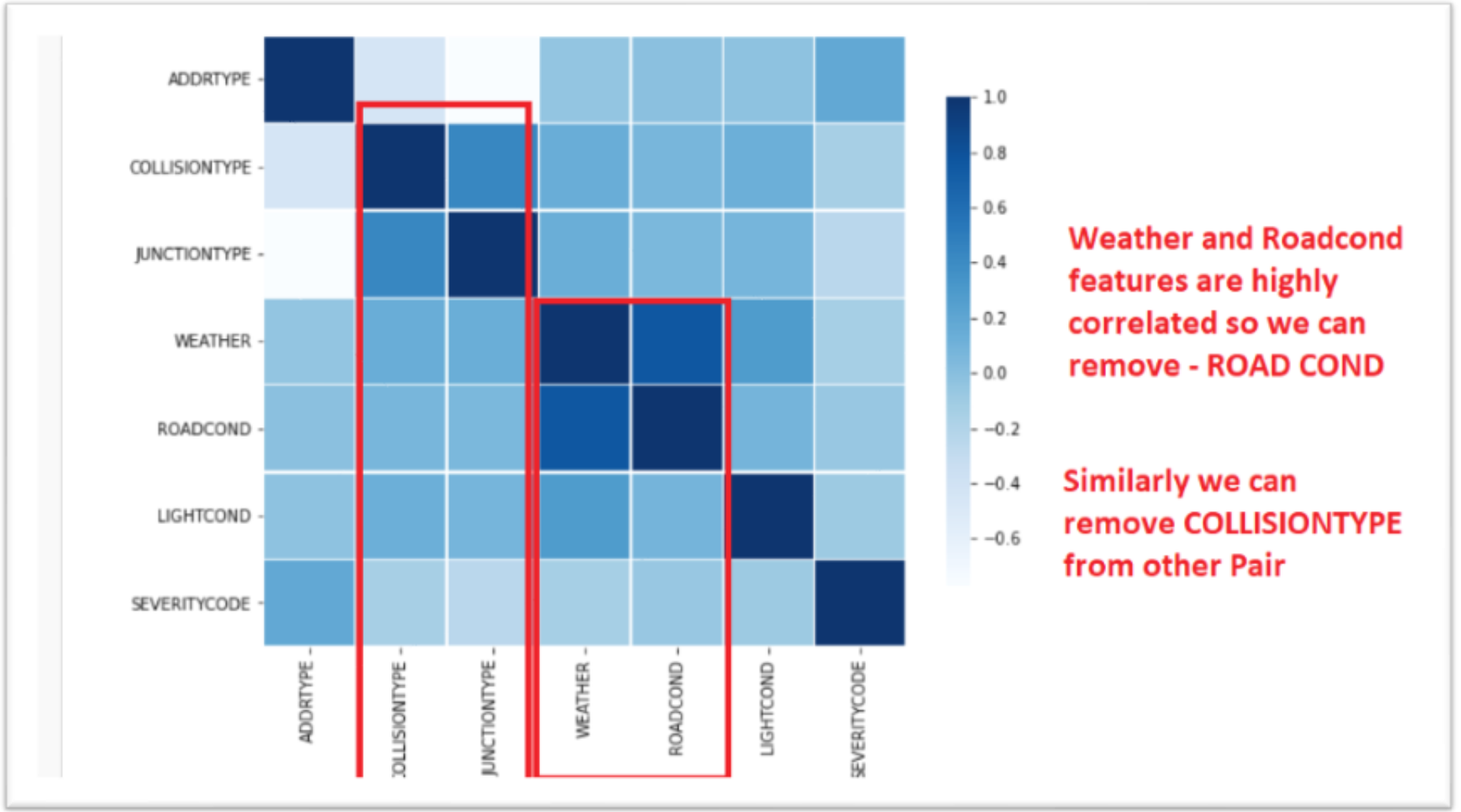
['ADDRTYPE','JUNCTIONTYPE','COLLISIONTYPE'] In each field severity distribution is havind correlation we can include these columns

EXPLORATORY DATA ANALYSIS

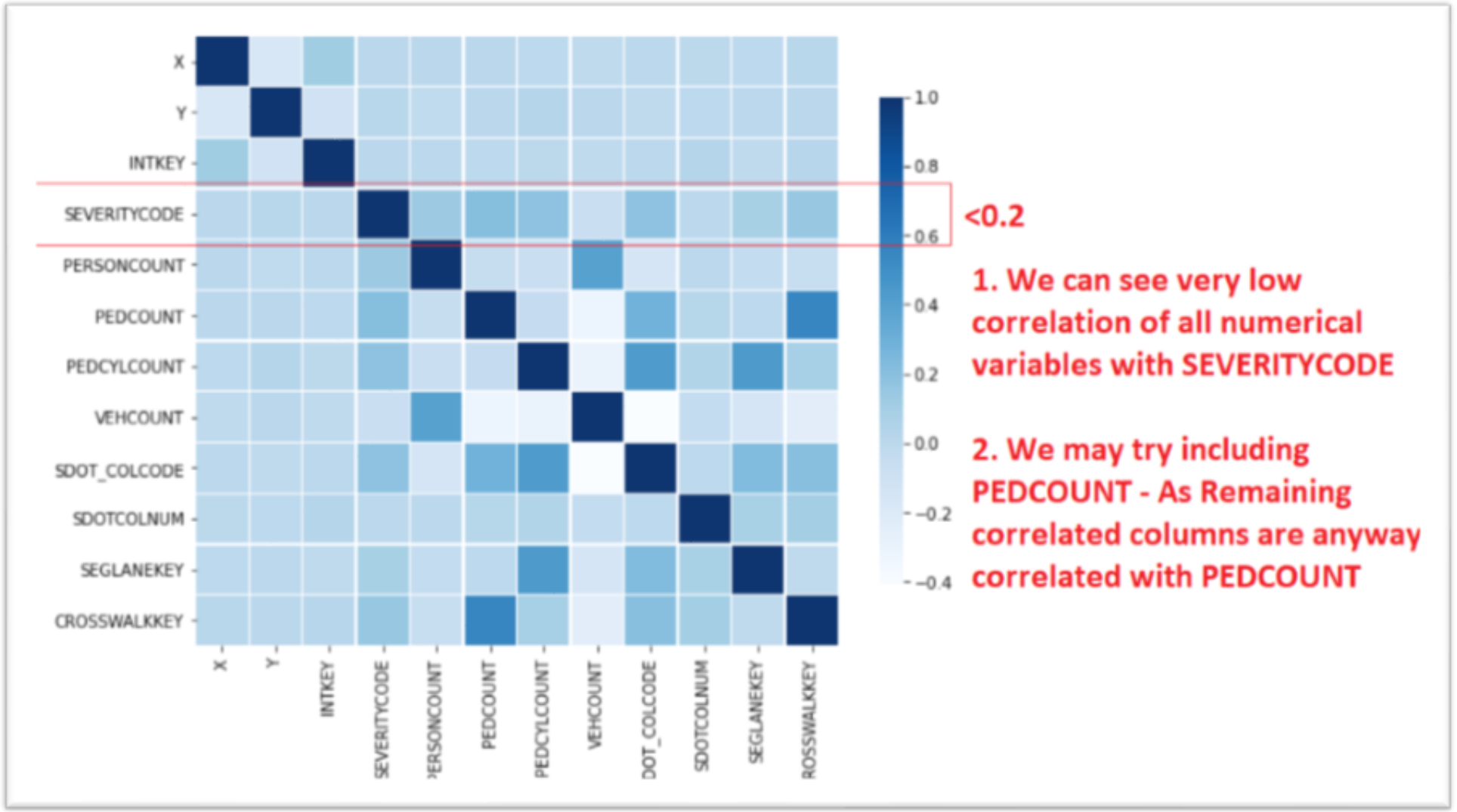


['LIGHTCOND','ROADCOND','WEATHER'] Severity distribution is having some correlation with fields so we can include these columns and test the performance

EXPLORATORY DATA ANALYSIS



EXPLORATORY DATA ANALYSIS



FEATURE ENGINEERING

- After all the EDA I can say the final list of features are as below
  - From Numerical features (#PEDCOUNT)
  - From Categorical features (ADDRTYPE, COLLISIONTYPE, JUNCTIONTYPE, WEATHER, ROADCOND, LIGHTCOND)

```
print('label:',label)
print('features:')
final_features
```

```
label: SEVERITYCODE
features: ['ADDRTYPE',
           'COLLISIONTYPE',
           'JUNCTIONTYPE',
           'WEATHER',
           'ROADCOND',
           'LIGHTCOND',
           'PEDCOUNT']
```

MODEL PROCESS

- Label Encoding for Categorical features
- Creating Training, Test set by balancing severity ratio
- Training by applying different Classification Techniques
- Model Evaluation and Hyper parameters tuning
- Model Score Report

MODEL SCORES REPORT

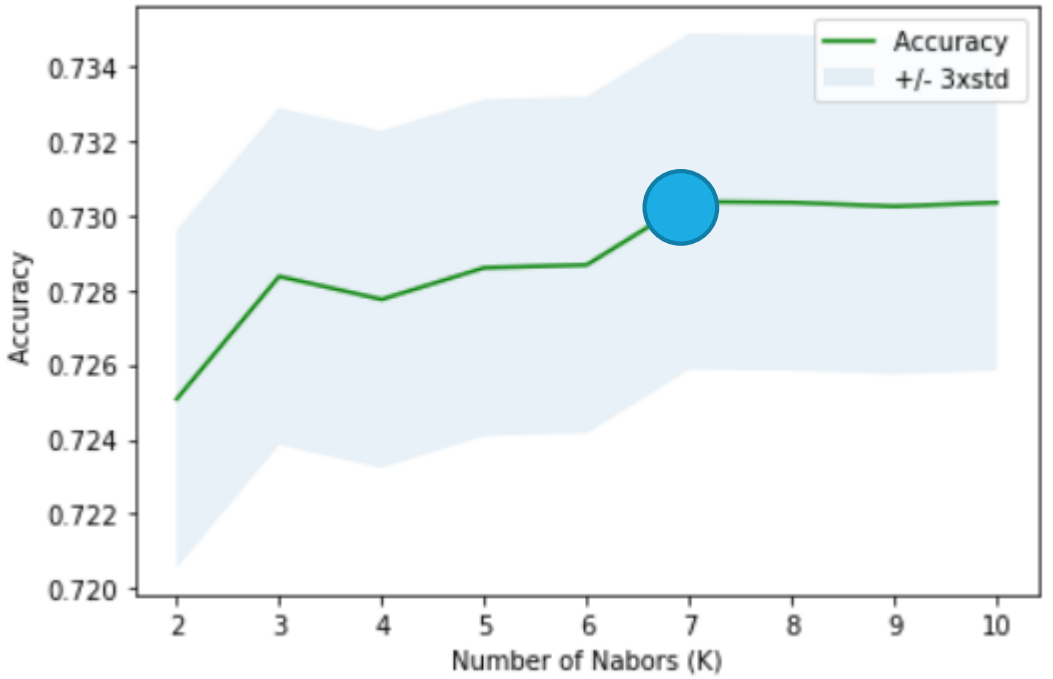
- I have used trained the data on 3 different algorithms
- I did Hyper parameter tuning by changing the weights and getting almost same model scores
- I was Training on SVM also it is taking lot of time so not putting that here

	Algorithm	F1-score	Jaccard	LogLoss
0	KNN	0.837984	0.730371	NA
1	Decision Tree	0.838124	0.730551	NA
2	LogisticRegression	0.837691	0.730037	0.556894

CLASSIFICATION EVALUATION

- At k value 7 is where we are getting better accuracy of 0.73

	precision	recall	f1-score	support
1	0.72	0.99	0.84	27297
2	0.90	0.11	0.20	11638
micro avg	0.73	0.73	0.73	38935
macro avg	0.81	0.55	0.52	38935
weighted avg	0.78	0.73	0.65	38935



DECISION TREE EVALUATION

- At k value 4 is where we are getting better F1-score of 0.73

Decision Trees is giving best results at Dept = 4 using entropy

[49]:
```
yhat=drugTree.predict(X_test)
yhat_prob=drugTree.predict_proba(X_test)

report=report.append({"Algorithm": "Decision Tree" , "Jaccard": jacc

print (classification_report(y_test, yhat))
```

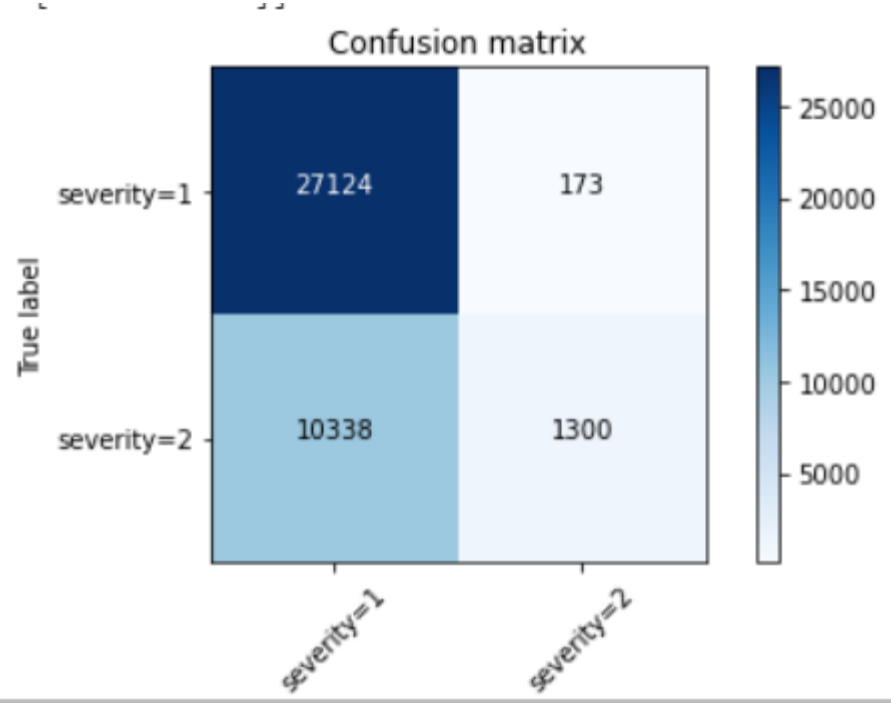
```
              precision    recall  f1-score   support

           1       0.72      0.99      0.84     27297
           2       0.90      0.11      0.20     11638

   micro avg       0.73      0.73      0.73     38935
   macro avg       0.81      0.55      0.52     38935
weighted avg       0.78      0.73      0.65     38935
```

DECISION TREE EVALUATION

- At k value 4 is where we are getting better F1-score of 0.84



Logistic Regression is performing best with regularization val = 0.03

[54]:
```
yhat=LR.predict(X_test)
yhat_prob=LR.predict_proba(X_test)

report=report.append({"Algorithm": "LogisticRegression" , "Jaccard": jaccar
print (classification_report(y_test, yhat))
```

	precision	recall	f1-score	support
1	0.72	0.99	0.84	27297
2	0.88	0.11	0.20	11638
micro avg	0.73	0.73	0.73	38935
macro avg	0.80	0.55	0.52	38935
weighted avg	0.77	0.73	0.65	38935

SUMMARY CONCLUSION

- Useful and informative models built to predict accident severity
- Value in guiding public traffic polices to focus on important factors to prevent accident injuries
- Accuracy of model has room for improvement, more insights could be gained
  - Collision type be further processed and used in model
  - Accident address be grouped based on injury occurrence ratio and used in model
  - Accident trend by dates

Thank You