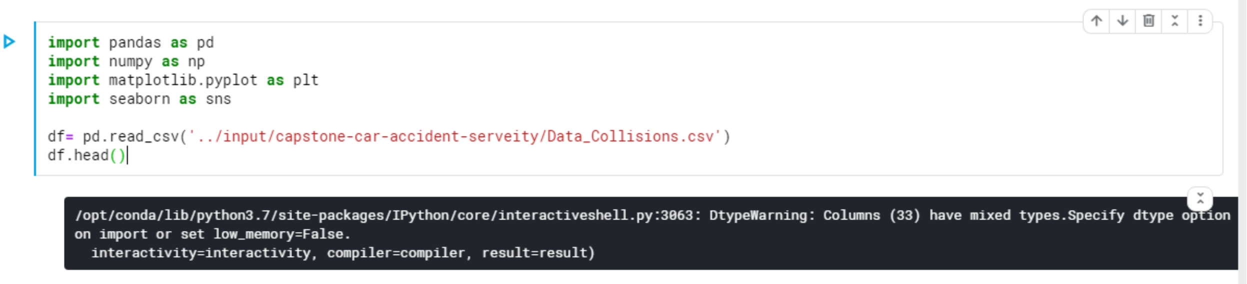
IBM Data Science Capstone:

Introduction:

I will be analyzing traffic accidents and the various circumstances that contribute to them. I will look at the severity and the conditions that cause them. I will use data analysis and machine learning techniques to try to predict the severity of future accidents.

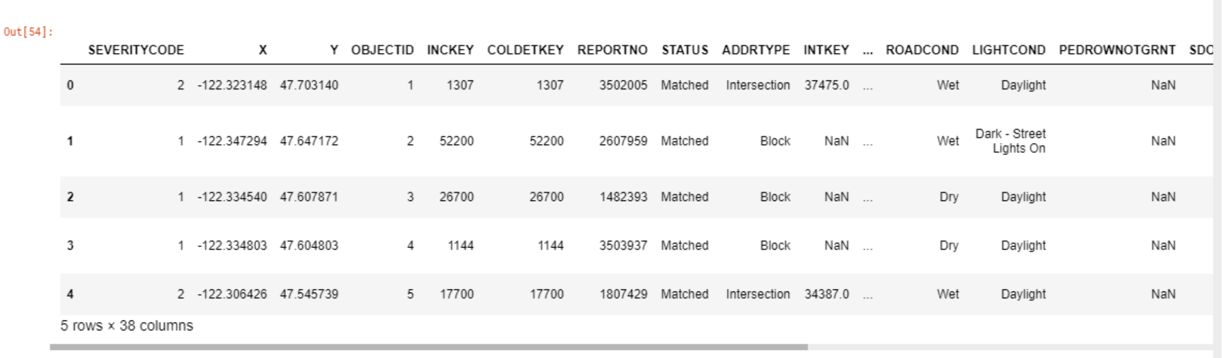
Data:

I will be using a data set provided by IBM that includes extensive data on traffic accidents.



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

df= pd.read_csv('../input/capstone-car-accident-serveity/Data_Collisions.csv')
df.head()
```

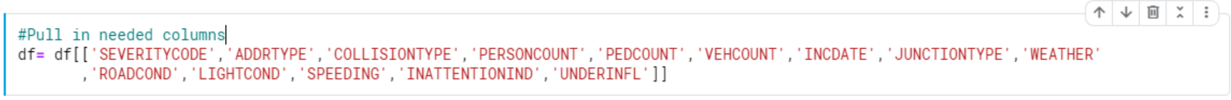
```
/opt/conda/lib/python3.7/site-packages/IPython/core/interactiveshell.py:3063: DtypeWarning: Columns (33) have mixed types.Specify dtype option
on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```



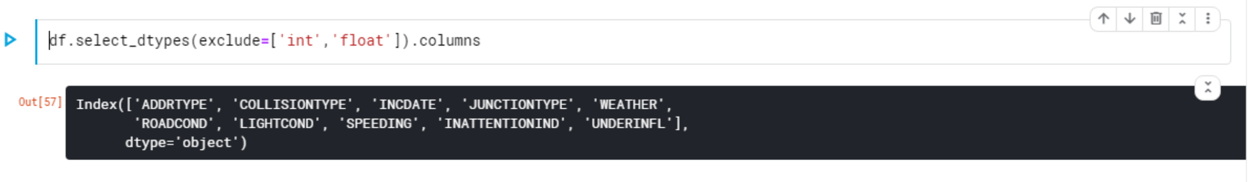
Out[54]:

	SEVERITYCODE	X	Y	OBJECTID	INCKEY	COLDETKEY	REPORTNO	STATUS	ADDRTYPE	INTKEY	...	ROADCOND	LIGHTCOND	PEDROWNOTGRNT	SDC
0	2	-122.323148	47.703140	1	1307	1307	3502005	Matched	Intersection	37475.0	...	Wet	Daylight	NaN	
1	1	-122.347294	47.647172	2	52200	52200	2607959	Matched	Block	NaN	...	Wet	Dark - Street Lights On	NaN	
2	1	-122.334540	47.607871	3	26700	26700	1482393	Matched	Block	NaN	...	Dry	Daylight	NaN	
3	1	-122.334803	47.604803	4	1144	1144	3503937	Matched	Block	NaN	...	Dry	Daylight	NaN	
4	2	-122.306426	47.545739	5	17700	17700	1807429	Matched	Intersection	34387.0	...	Wet	Daylight	NaN	

5 rows × 38 columns



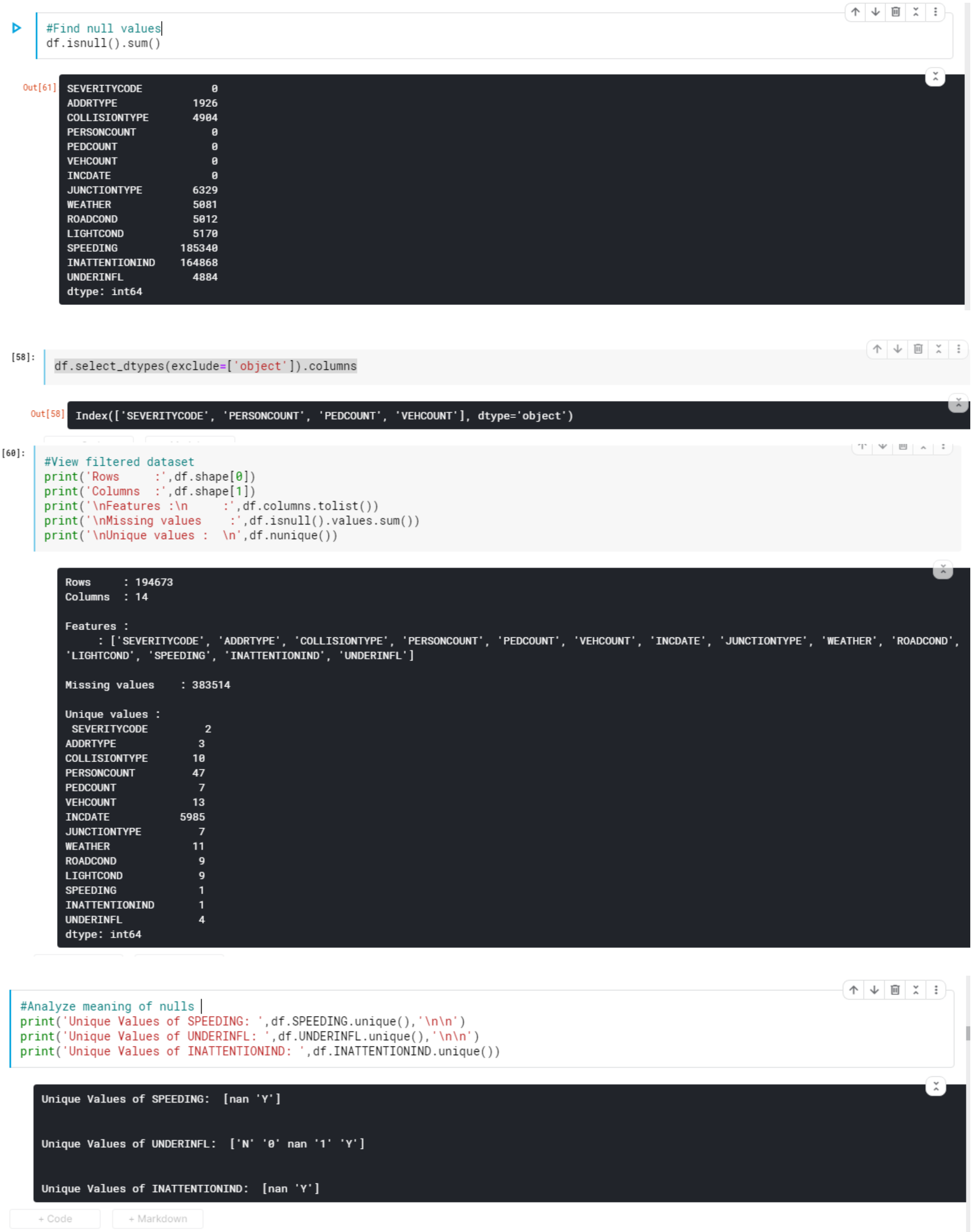
```
#Pull in needed columns
df= df[['SEVERITYCODE','ADDRTYPE','COLLISIONTYPE','PERSONCOUNT','PEDCOUNT','VEHCOUNT','INCDATE','JUNCTIONTYPE','WEATHER'
       ,'ROADCOND','LIGHTCOND','SPEEDING','INATTENTIONIND','UNDERINFL']]
```



```
df.select_dtypes(exclude=['int','float']).columns
```

Out[57]

```
Index(['ADDRTYPE', 'COLLISIONTYPE', 'INCDATE', 'JUNCTIONTYPE', 'WEATHER',
       'ROADCOND', 'LIGHTCOND', 'SPEEDING', 'INATTENTIONIND', 'UNDERINFL'],
      dtype='object')
```



```
#Find null values
df.isnull().sum()
```

Out[61]

```
SEVERITYCODE           0
ADDRTYPE            1926
COLLISIONTYPE       4904
PERSONCOUNT            0
PEDCOUNT               0
VEHCOUNT               0
INCDATE                0
JUNCTIONTYPE        6329
WEATHER             5081
ROADCOND            5012
LIGHTCOND           5170
SPEEDING          185340
INATTENTIONIND    164868
UNDERINFL           4884
dtype: int64
```

[58]:

```
df.select_dtypes(exclude=['object']).columns
```

Out[58]

```
Index(['SEVERITYCODE', 'PERSONCOUNT', 'PEDCOUNT', 'VEHCOUNT'], dtype='object')
```

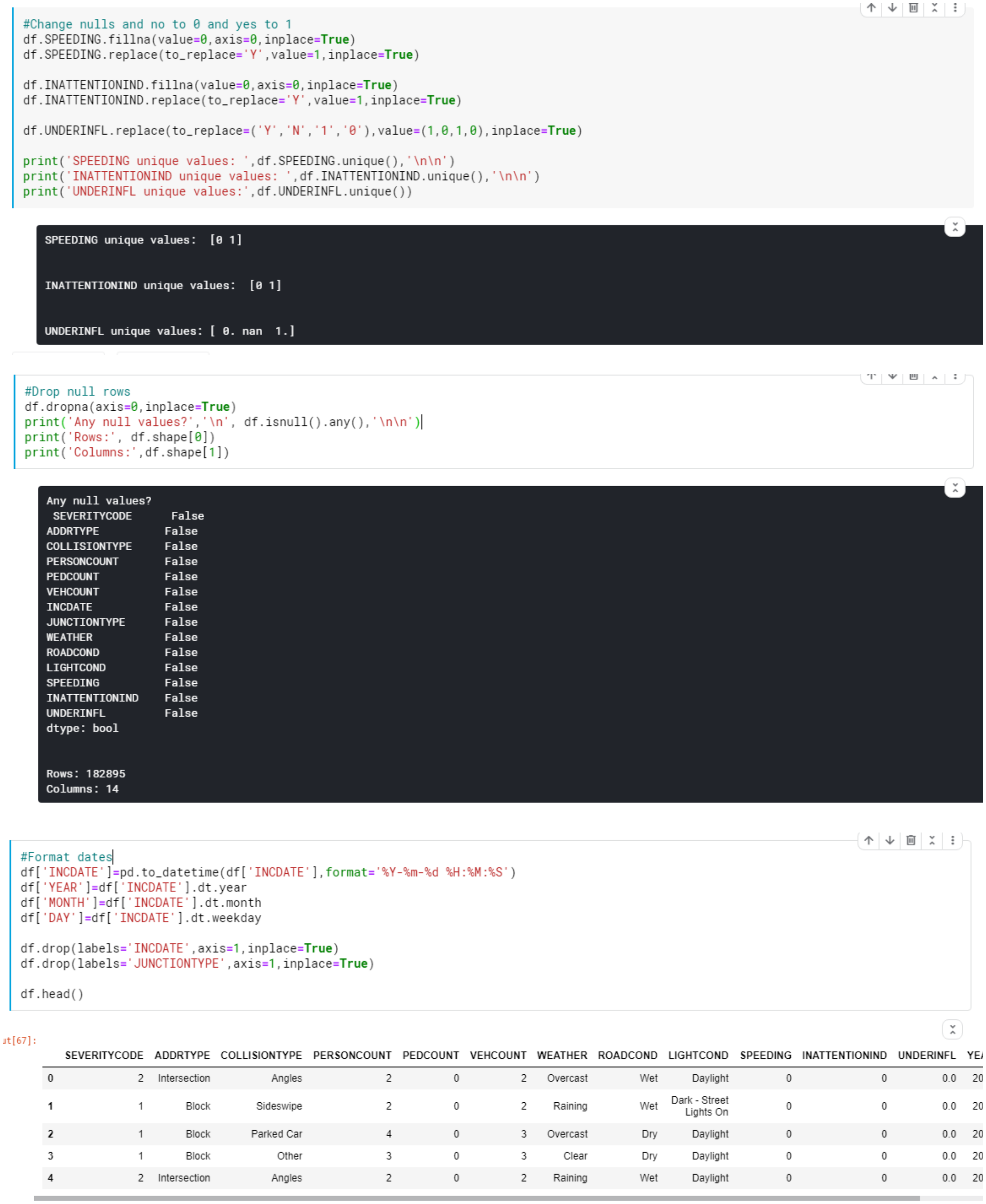
[60]:

```
#View filtered dataset
print('Rows     :',df.shape[0])
print('Columns  :',df.shape[1])
print('\nFeatures :\n     :',df.columns.tolist())
print('\nMissing values    :',df.isnull().values.sum())
print('\nUnique values :  \n',df.nunique())
```

```
Rows     : 194673
Columns  : 14

Features :
     : ['SEVERITYCODE', 'ADDRTYPE', 'COLLISIONTYPE', 'PERSONCOUNT', 'PEDCOUNT', 'VEHCOUNT', 'INCDATE', 'JUNCTIONTYPE', 'WEATHER', 'ROADCOND', 'LIGHTCOND', 'SPEEDING', 'INATTENTIONIND', 'UNDERINFL']

Missing values    : 383514

Unique values :
 SEVERITYCODE         2
ADDRTYPE             3
COLLISIONTYPE       10
PERSONCOUNT         47
PEDCOUNT             7
VEHCOUNT            13
INCDATE           5985
JUNCTIONTYPE         7
WEATHER             11
ROADCOND             9
LIGHTCOND            9
SPEEDING             1
INATTENTIONIND       1
UNDERINFL            4
dtype: int64
```

```
#Analyze meaning of nulls
print('Unique Values of SPEEDING: ',df.SPEEDING.unique(),'\n\n')
print('Unique Values of UNDERINFL: ',df.UNDERINFL.unique(),'\n\n')
print('Unique Values of INATTENTIONIND: ',df.INATTENTIONIND.unique())
```

```
Unique Values of SPEEDING:  [nan 'Y']


Unique Values of UNDERINFL:  ['N' '0' nan '1' 'Y']


Unique Values of INATTENTIONIND:  [nan 'Y']
```



```
#Change nulls and no to 0 and yes to 1
df.SPEEDING.fillna(value=0,axis=0,inplace=True)
df.SPEEDING.replace(to_replace='Y',value=1,inplace=True)

df.INATTENTIONIND.fillna(value=0,axis=0,inplace=True)
df.INATTENTIONIND.replace(to_replace='Y',value=1,inplace=True)

df.UNDERINFL.replace(to_replace=('Y','N','1','0'),value=(1,0,1,0),inplace=True)

print('SPEEDING unique values: ',df.SPEEDING.unique(),'\n\n')
print('INATTENTIONIND unique values: ',df.INATTENTIONIND.unique(),'\n\n')
print('UNDERINFL unique values:',df.UNDERINFL.unique())
```

```
SPEEDING unique values:  [0 1]


INATTENTIONIND unique values:  [0 1]


UNDERINFL unique values: [ 0. nan  1.]
```

```
#Drop null rows
df.dropna(axis=0,inplace=True)
print('Any null values?','\n', df.isnull().any(),'\n\n')
print('Rows:', df.shape[0])
print('Columns:',df.shape[1])
```

```
Any null values?
 SEVERITYCODE      False
ADDRTYPE          False
COLLISIONTYPE     False
PERSONCOUNT       False
PEDCOUNT          False
VEHCOUNT          False
INCDATE           False
JUNCTIONTYPE      False
WEATHER           False
ROADCOND          False
LIGHTCOND         False
SPEEDING          False
INATTENTIONIND    False
UNDERINFL         False
dtype: bool


Rows: 182895
Columns: 14
```

```
#Format dates
df['INCDATE']=pd.to_datetime(df['INCDATE'],format='%Y-%m-%d %H:%M:%S')
df['YEAR']=df['INCDATE'].dt.year
df['MONTH']=df['INCDATE'].dt.month
df['DAY']=df['INCDATE'].dt.weekday

df.drop(labels='INCDATE',axis=1,inplace=True)
df.drop(labels='JUNCTIONTYPE',axis=1,inplace=True)

df.head()
```

ut[67]:

	SEVERITYCODE	ADDRTYPE	COLLISIONTYPE	PERSONCOUNT	PEDCOUNT	VEHCOUNT	WEATHER	ROADCOND	LIGHTCOND	SPEEDING	INATTENTIONIND	UNDERINFL	YEA
0	2	Intersection	Angles	2	0	2	Overcast	Wet	Daylight	0	0	0.0	20
1	1	Block	Sideswipe	2	0	2	Raining	Wet	Dark - Street Lights On	0	0	0.0	20
2	1	Block	Parked Car	4	0	3	Overcast	Dry	Daylight	0	0	0.0	20
3	1	Block	Other	3	0	3	Clear	Dry	Daylight	0	0	0.0	20
4	2	Intersection	Angles	2	0	2	Raining	Wet	Daylight	0	0	0.0	20