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

```
df = pd.read_csv("Accident Prediction.csv")
df.shape
print(df.head(5))
```

```
       State Name  City Name  Year    Month Day of Week Time of Day  \
0   Uttar Pradesh    Lucknow  2020     June    Saturday    00:31:00
1       Rajasthan    Jodhpur  2021     July    Saturday    08:26:00
2  Andhra Pradesh   Tirupati  2019  October      Sunday    12:47:00
3       Karnataka  Bangalore  2023  October    Thursday    18:59:00
4   Uttar Pradesh   Varanasi  2022     July      Friday    18:51:00

  Driver Gender  Driver Age Driver License Status Accident Severity  ...  \
0        Female          34                Valid             Minor  ...
1          Male          53                Valid             Fatal  ...
2        Female          51                Valid             Minor  ...
3        Female          51              Expired             Fatal  ...
4        Female          23              Expired             Fatal  ...

   Number of Casualties Number of Fatalities  Weather Conditions  \
0                    10                    5               Rainy
1                     7                    1               Clear
2                     7                    4               Rainy
3                    10                    2                Hazy
4                     9                    4               Clear

          Road Type Road Condition Lighting Conditions  \
0     State Highway            Dry               Dark
1        Urban Road        Damaged               Dawn
2      Village Road        Damaged               Dark
3      Village Road            Wet               Dark
4  National Highway            Wet               Dawn

  Traffic Control Presence Speed Limit (km/h) Alcohol Involvement  \
0                  Signals                76                 Yes
1                    Signs                35                  No
2                    Signs                90                  No
3                    Signs               114                  No
4                  Signals                34                 Yes

   Accident Location Details
0              Straight Road
1              Straight Road
2                      Curve
3               Intersection
4                     Bridge

[5 rows x 22 columns]
```

```
print(df.describe())
```

```
              Year  Driver Age  Number of Vehicles Involved  \
count   149.000000  149.000000                   149.000000
mean   2020.731544   43.993289                     2.932886
std       1.654767   15.793215                     1.403010
min    2018.000000   18.000000                     1.000000
25%    2020.000000   29.000000                     2.000000
50%    2021.000000   45.000000                     3.000000
75%    2022.000000   57.000000                     4.000000
max    2023.000000   69.000000                     5.000000

       Number of Casualties  Number of Fatalities  Speed Limit (km/h)
count            149.000000            149.000000          149.000000
mean               5.268456              2.442953           75.053691
std                3.268670              1.653753           26.452095
min                0.000000              0.000000           30.000000
25%                2.000000              1.000000           51.000000
50%                6.000000              2.000000           77.000000
75%                8.000000              4.000000           99.000000
max               10.000000              5.000000          120.000000
```

```
df = df.drop_duplicates()
print(df.isnull().sum())
```

```
State Name                  0
City Name                   0
Year                        0
Month                       0
Day of Week                 0
Time of Day                 0
```

```
Driver Gender                0
Driver Age                   0
Driver License Status        0
Accident Severity            0
Number of Vehicles Involved  0
Vehicle Type Involved        0
Number of Casualties         0
Number of Fatalities         0
Weather Conditions           0
Road Type                    0
Road Condition               0
Lighting Conditions          0
Traffic Control Presence     0
Speed Limit (km/h)           0
Alcohol Involvement          0
Accident Location Details    0
dtype: int64
```

```
print(df['Accident Severity'].value_counts())
```

```
Accident Severity
Fatal      56
Serious    48
Minor      45
Name: count, dtype: int64
```

```
from sklearn.preprocessing import LabelEncoder
```

```
for col in df.columns:
    if df[col].dtype == 'object':
        encoder = LabelEncoder()
        df[col] = encoder.fit_transform(df[col])
```

```
x=df.drop('Accident Severity',axis=1)
y=df['Accident Severity']
```

```
from sklearn.model_selection import train_test_split
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report
```

```
model_lr = LogisticRegression()
model_lr.fit(x_train, y_train)
y_pred = model_lr.predict(x_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
print("Classification Report:\n", classification_report(y_test, y_pred))
```

```
Accuracy: 0.4666666666666667
Classification Report:
              precision    recall  f1-score   support

           0       0.44      0.70      0.54        10
           1       0.67      0.60      0.63        10
           2       0.20      0.10      0.13        10

    accuracy                           0.47        30
   macro avg       0.43      0.47      0.43        30
weighted avg       0.43      0.47      0.43        30

/usr/local/lib/python3.12/dist-packages/sklearn/linear_model/_logistic.py:465: ConvergenceWarning: lbfgs failed to converge
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

```
from sklearn.neighbors import KNeighborsClassifier
```

```
model_knn=KNeighborsClassifier()
model_knn.fit(x_train,y_train)
y_pred=model_knn.predict(x_test)
accuracy=accuracy_score(y_test,y_pred)
print("Accuracy:",accuracy)
```

```
Accuracy: 0.3333333333333333
```

```
from sklearn.tree import DecisionTreeClassifier
```

```
model_dt=DecisionTreeClassifier()
model_dt.fit(x_train,y_train)
y_pred=model_dt.predict(x_test)
accuracy=accuracy_score(y_test,y_pred)
print("Accuracy:",accuracy)
```

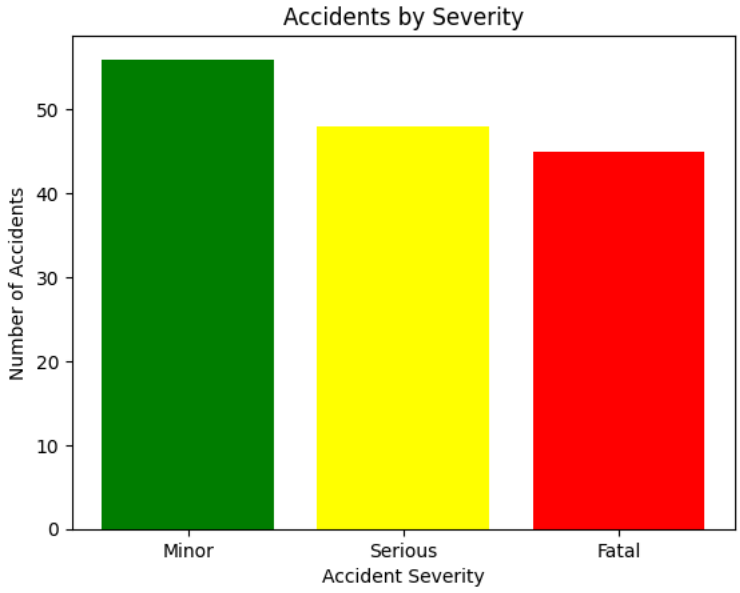
```
Accuracy: 0.3333333333333333
```

```
print("Logistic Regression Accuracy:", accuracy_score(y_test, y_pred))
print("KNN Accuracy:", accuracy_score(y_test, y_pred))
print("Decision Tree Accuracy:", accuracy_score(y_test, y_pred))
```

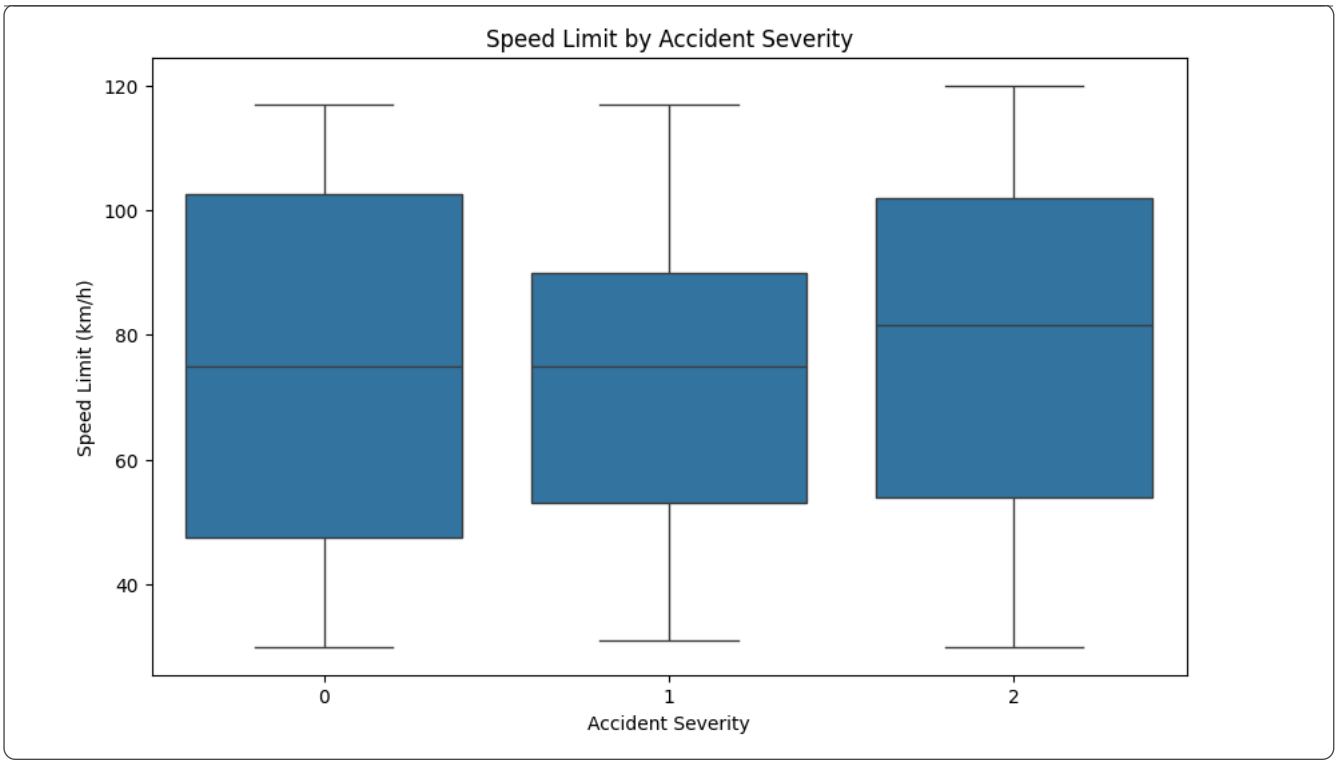
```
Logistic Regression Accuracy: 0.3333333333333333
KNN Accuracy: 0.3333333333333333
Decision Tree Accuracy: 0.3333333333333333
```

```
x = ["Minor", "Serious", "Fatal"]
y = [56, 48, 45]
colors = ["green", "yellow", "red"]

plt.bar(x, y, color=colors, width=0.8)
plt.xlabel("Accident Severity")
plt.ylabel("Number of Accidents")
plt.title("Accidents by Severity")
plt.show()
```




```
if 'Speed Limit (km/h)' in df.columns:
    plt.figure(figsize=(10,6))
    sns.boxplot(data=df, x='Accident Severity', y='Speed Limit (km/h)')
    plt.title("Speed Limit by Accident Severity")
    plt.show()
```


Speed Limit by Accident Severity
Speed Limit (km/h)
120
100
80
60
40
0
1
2
Accident Severity