

```
import pandas as pd
from google.colab import files
Data=files.upload()
```



Choose Files

No file chosen

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving accident_prediction_india.csv to accident_prediction_india.csv

```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
```

```
df = pd.read_csv('accident_prediction_india.csv')
```

```
print(df.head())
print(df.info())
df.dropna(inplace=True)
label_cols = ['State Name', 'City Name', 'Month', 'Day of Week', 'Time of Day', 'Vehicle Ty
le = LabelEncoder()
for col in label_cols:
    df[col] = le.fit_transform(df[col])
```



	State Name	City Name	Year	Month	Day of Week	Time of Day	\
0	Jammu and Kashmir	Unknown	2021	May	Monday	1:46	
1	Uttar Pradesh	Lucknow	2018	January	Wednesday	21:30	
2	Chhattisgarh	Unknown	2023	May	Wednesday	5:37	
3	Uttar Pradesh	Lucknow	2020	June	Saturday	0:31	
4	Sikkim	Unknown	2021	August	Thursday	11:21	

	Accident Severity	Number of Vehicles Involved	Vehicle Type Involved	\
0	Serious	5	Cycle	
1	Minor	5	Truck	
2	Minor	5	Pedestrian	
3	Minor	3	Bus	
4	Minor	5	Cycle	

	Number of Casualties	...	Road Type	Road Condition	\
0	0	...	National Highway	Wet	
1	5	...	Urban Road	Dry	
2	6	...	National Highway	Under Construction	
3	10	...	State Highway	Dry	
4	7	...	Urban Road	Wet	

	Lighting Conditions	Traffic Control Presence	Speed Limit (km/h)	Driver Age	\
--	---------------------	--------------------------	--------------------	------------	---

0	Dark	Signs	61	66
1	Dusk	Signs	92	60
2	Dawn	Signs	120	26
3	Dark	Signals	76	34
4	Dusk	Signs	115	30

	Driver Gender	Driver License Status	Alcohol Involvement	\
0	Male	NaN	Yes	
1	Male	NaN	Yes	
2	Female	NaN	No	
3	Female	Valid	Yes	
4	Male	NaN	No	

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

[5 rows x 22 columns]

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 3000 entries, 0 to 2999

Data columns (total 22 columns):

#	Column	Non-Null Count	Dtype
---	-----	-----	-----
0	State Name	3000 non-null	object
1	City Name	3000 non-null	object
2	Year	3000 non-null	int64
3	Month	3000 non-null	object
4	Day of Week	3000 non-null	object
5	Time of Day	3000 non-null	object
6	Accident Severity	3000 non-null	object
7	Number of Vehicles Involved	3000 non-null	int64
8	Vehicle Type Involved	3000 non-null	object
9	Number of Casualties	3000 non-null	int64

```
severity_map = {'Minor': 0, 'Serious': 1, 'Fatal': 2}
```

```
df['Accident Severity'] = df['Accident Severity'].map(severity_map)
```

```
features = ['State Name', 'City Name', 'Month', 'Day of Week', 'Time of Day',
            'Number of Vehicles Involved', 'Vehicle Type Involved']
```

```
X = df[features]
```

```
y = df['Accident Severity']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
model = RandomForestClassifier(n_estimators=100, random_state=42)
```

```
model.fit(X_train, y_train)
```

```
y_pred = model.predict(X_test)
```

```
print(classification_report(y_test, y_pred))
```

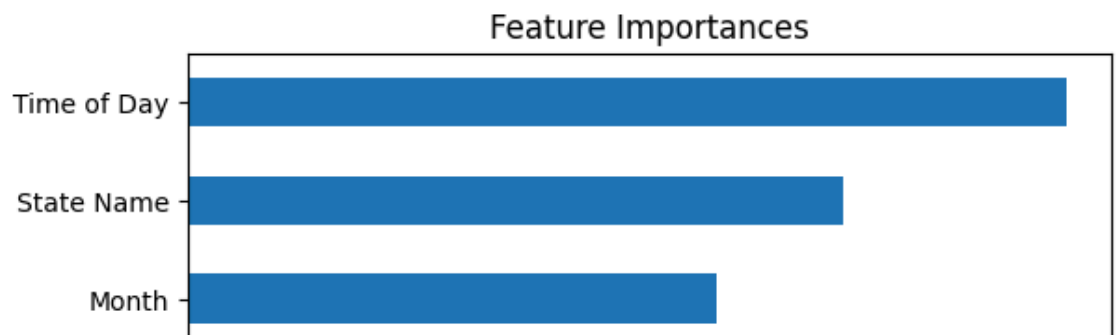
```
print(confusion_matrix(y_test, y_pred))
```



	precision	recall	f1-score	support
0	0.31	0.43	0.36	93
1	0.30	0.28	0.29	100
2	0.42	0.33	0.37	120
accuracy			0.34	313
macro avg	0.35	0.35	0.34	313
weighted avg	0.35	0.34	0.34	313


```
[[40 28 25]
 [44 28 28]
 [43 38 39]]
```

```
importances = pd.Series(model.feature_importances_, index=features)
importances.sort_values().plot(kind='barh', title='Feature Importances')
plt.show()
```



```
print("Initial Shape:", df.shape)
print("\nMissing Values:\n", df.isnull().sum())
print("Shape after dropping nulls:", df.shape)
print("Final Processed Data:\n", df.head())
```



```
Initial Shape: (1565, 22)
```

```
Missing Values:
State Name      0
City Name      0
Year           0
Month          0
Day of Week    0
Time of Day    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
Driver Age              0
Driver Gender           0
Driver License Status   0
Alcohol Involvement     0
Accident Location Details 0
```

```
dtype: int64
```

```
Shape after dropping nulls: (1565, 22)
```

```
Final Processed Data:
```

	State Name	City Name	Year	Month	Day of Week	Time of Day	\
3	29	10	2020	6	2	15	
5	18	23	2020	9	5	904	
7	10	23	2022	2	6	105	
8	24	7	2021	5	2	892	
13	3	23	2020	6	6	830	

	Accident Severity	Number of Vehicles Involved	Vehicle Type Involved	\
3	0	3	1	
5	1	1	5	
7	0	3	5	
8	2	3	5	
13	1	5	2	

	Number of Casualties	...	Road Type	Road Condition	\
3	10	...	State Highway	Dry	
5	6	...	Village Road	Dry	
7	1	...	State Highway	Damaged	
8	7	...	Urban Road	Damaged	
13	0	...	State Highway	Wet	

	Lighting Conditions	Traffic Control Presence	Speed Limit (km/h)	Driver Age	\
3	Dark	Signals	76	34	
5	Daylight	Police Checkpost	48	44	
7	Dark	Police Checkpost	62	42	
8	Dawn	Signs	35	53	
13	Dusk	Signals	98	44	

	Driver Gender	Driver License Status	Alcohol Involvement	\
3	Female	Valid	Yes	

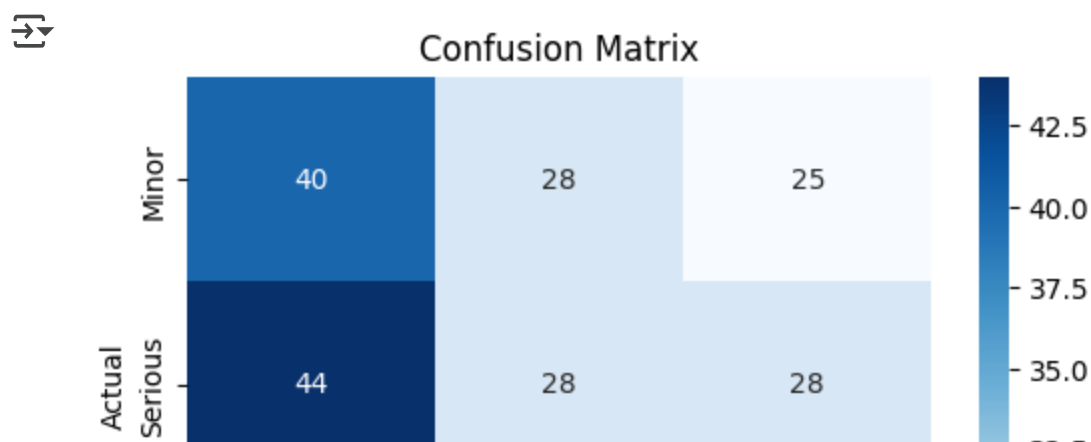
```
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```



```
RandomForestClassifier
RandomForestClassifier(random_state=42)
```

```
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=['Minor', 'Serious', 'Fatal'],
            yticklabels=['Minor', 'Serious', 'Fatal'])
```

```
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```



```
from sklearn.metrics import accuracy_score
```

```
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

```
Accuracy: 0.34185303514376997
```

Classification Report:

	precision	recall	f1-score	support
0	0.31	0.43	0.36	93
1	0.30	0.28	0.29	100
2	0.42	0.33	0.37	120
accuracy			0.34	313
macro avg	0.35	0.35	0.34	313
weighted avg	0.35	0.34	0.34	313

```
from google.colab import drive
drive.mount('/content/drive')
```

