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

# Load dataset
# Replace 'accidents.csv' with your actual file path
df = pd.read_csv('accidents.csv')

# Display basic info
print("Dataset head:")
print(df.head())

# Drop rows with missing values for simplicity
df.dropna(inplace=True)

# Select relevant features (example features; adjust based on your dataset)
features = ['Severity', 'Temperature(F)', 'Humidity(%)', 'Pressure(in)',
            'Visibility(mi)', 'Wind_Speed(mph)', 'Weather_Condition']

df = df[features]

# Convert categorical data to numeric
df = pd.get_dummies(df, columns=['Weather_Condition'], drop_first=True)

# Split dataset into X and y
X = df.drop('Severity', axis=1)
y = df['Severity']

# Train/Test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Model training
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Prediction
y_pred = model.predict(X_test)

# Evaluation
print("\nClassification Report:")
print(classification_report(y_test, y_pred))

# Confusion Matrix
conf_matrix = confusion_matrix(y_test, y_pred)
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues')
```

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