```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, accuracy_score
import matplotlib.pyplot as plt
import seaborn as sns

# Create a sample DataFrame
# Replace with your desired features and data types
data = {
    'speed_control': np.random.randint(0, 10, size=100),
    'vehicle_count': np.random.rand(100),
    'hours': np.random.choice(['A', 'B', 'C'], size=100),
    'Accident_severity': np.random.randint(0, 3, size=100)  # Example
target variable
}
df = pd.DataFrame(data)

# Encode categorical variables
label_encoders = {}
for column in df.columns:
    if df[column].dtype == 'object':
        le = LabelEncoder()
        df[column] = le.fit_transform(df[column].astype(str))
        label_encoders[column] = le

# Define features and target
features = df.drop(columns=['Accident_severity'])
target = df['Accident_severity']

# Split data
X_train, X_test, y_train, y_test = train_test_split(features, target,
test_size=0.3, random_state=42)

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

# Predict
y_pred = model.predict(X_test)

# Evaluate
print("Classification Report:\n", classification_report(y_test,
y_pred))
print("Accuracy:", accuracy_score(y_test, y_pred))
```

```
# Feature importance
importances = model.feature_importances_
indices = np.argsort(importances)[::-1]

plt.figure(figsize=(12, 8))
sns.barplot(x=importances[indices], y=features.columns[indices])
plt.title("Feature Importances")
plt.tight_layout()
plt.show()
```