```python
import pandas as pd

df = pd.read_csv("car_evaluation.csv")

df.head()
df.tail()
```

```
     vhigh vhigh.1      2   2.1 small   low  unacc
1722   low     low  5more  more   med   med   good
1723   low     low  5more  more   med  high  vgood
1724   low     low  5more  more   big   low  unacc
1725   low     low  5more  more   big   med   good
1726   low     low  5more  more   big  high  vgood
```

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1727 entries, 0 to 1726
Data columns (total 7 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   vhigh    1727 non-null   object
 1   vhigh.1  1727 non-null   object
 2   2        1727 non-null   object
 3   2.1      1727 non-null   object
 4   small    1727 non-null   object
 5   low      1727 non-null   object
 6   unacc    1727 non-null   object
dtypes: object(7)
memory usage: 94.6+ KB
```

```python
col_names = ['buying', 'maint', 'doors', 'persons', 'lug_boot', 
'safety', 'class']


df.columns = col_names

col_names
```

```
['buying', 'maint', 'doors', 'persons', 'lug_boot', 'safety', 'class']
```

```python
col_names = ['buying', 'maint', 'doors', 'persons', 'lug_boot', 
'safety', 'class']


for col in col_names:

    print(df[col].value_counts())
```

```
buying
high     432
```

```
med      432
low      432
vhigh    431
Name: count, dtype: int64
maint
high     432
med      432
low      432
vhigh    431
Name: count, dtype: int64
doors
3        432
4        432
5more    432
2        431
Name: count, dtype: int64
persons
4       576
more    576
2       575
Name: count, dtype: int64
lug_boot
med      576
big      576
small    575
Name: count, dtype: int64
safety
med     576
high    576
low     575
Name: count, dtype: int64
class
unacc    1209
acc       384
good       69
vgood      65
Name: count, dtype: int64
```

```python
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
for column in df.columns:
    df[column] = le.fit_transform(df[column])

X = df.drop('class', axis=1)
y = df['class']

from sklearn.model_selection import train_test_split
```

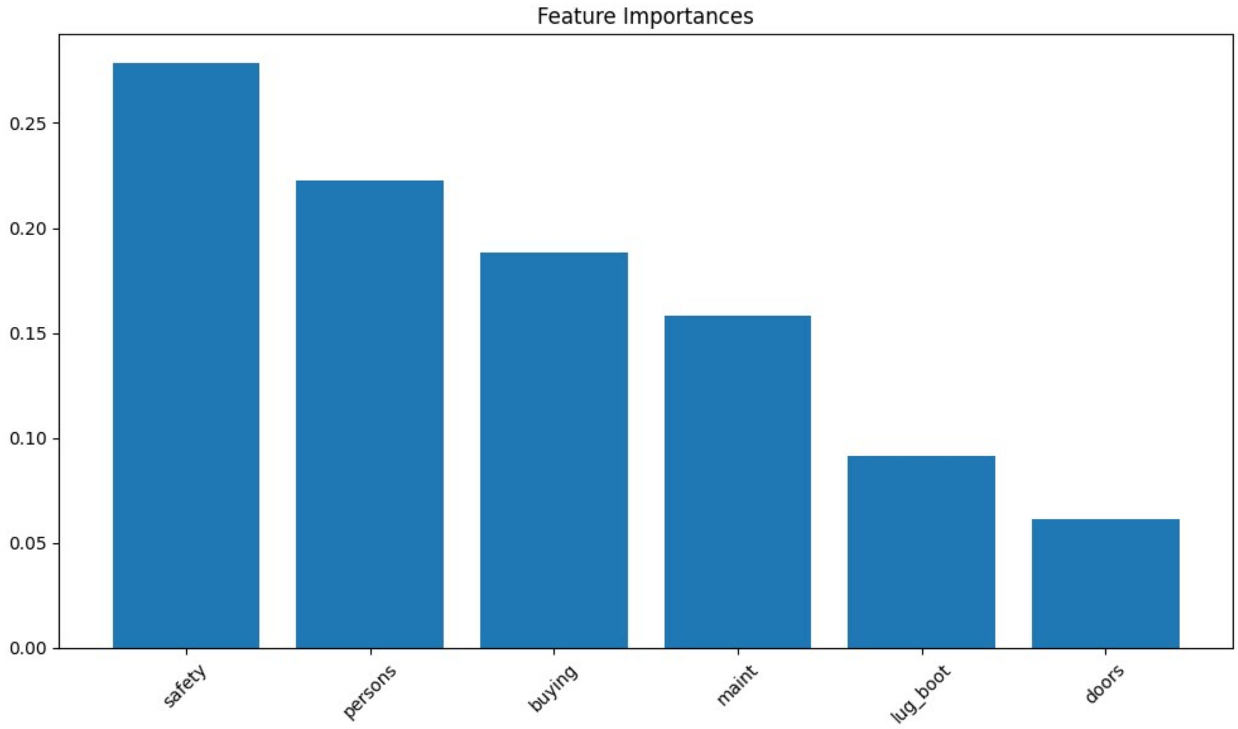
```python
X_train, X_test, y_train, y_test = train_test_split(X, y, 
test_size=0.3, random_state=42)

from sklearn.ensemble import RandomForestClassifier

rf_clf = RandomForestClassifier(n_estimators=100, criterion='entropy', 
random_state=42)

rf_clf.fit(X_train, y_train)
```

```
RandomForestClassifier(criterion='entropy', random_state=42)
```

```python
from sklearn.metrics import accuracy_score

y_pred = rf_clf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Random Forest Accuracy:", accuracy)
```

```
Random Forest Accuracy: 0.9672447013487476
```

```python
import matplotlib.pyplot as plt
import numpy as np

importances = rf_clf.feature_importances_
indices = np.argsort(importances)[::-1]

plt.figure(figsize=(10,6))
plt.title("Feature Importances")
plt.bar(range(X.shape[1]), importances[indices], align="center")
plt.xticks(range(X.shape[1]), X.columns[indices], rotation=45)
plt.tight_layout()
plt.show()
```



```
from sklearn.tree import plot_tree

plt.figure(figsize=(20,10))
plot_tree(rf_clf.estimators_[0], feature_names=X.columns, filled=True,
rounded=True)
plt.show()
```