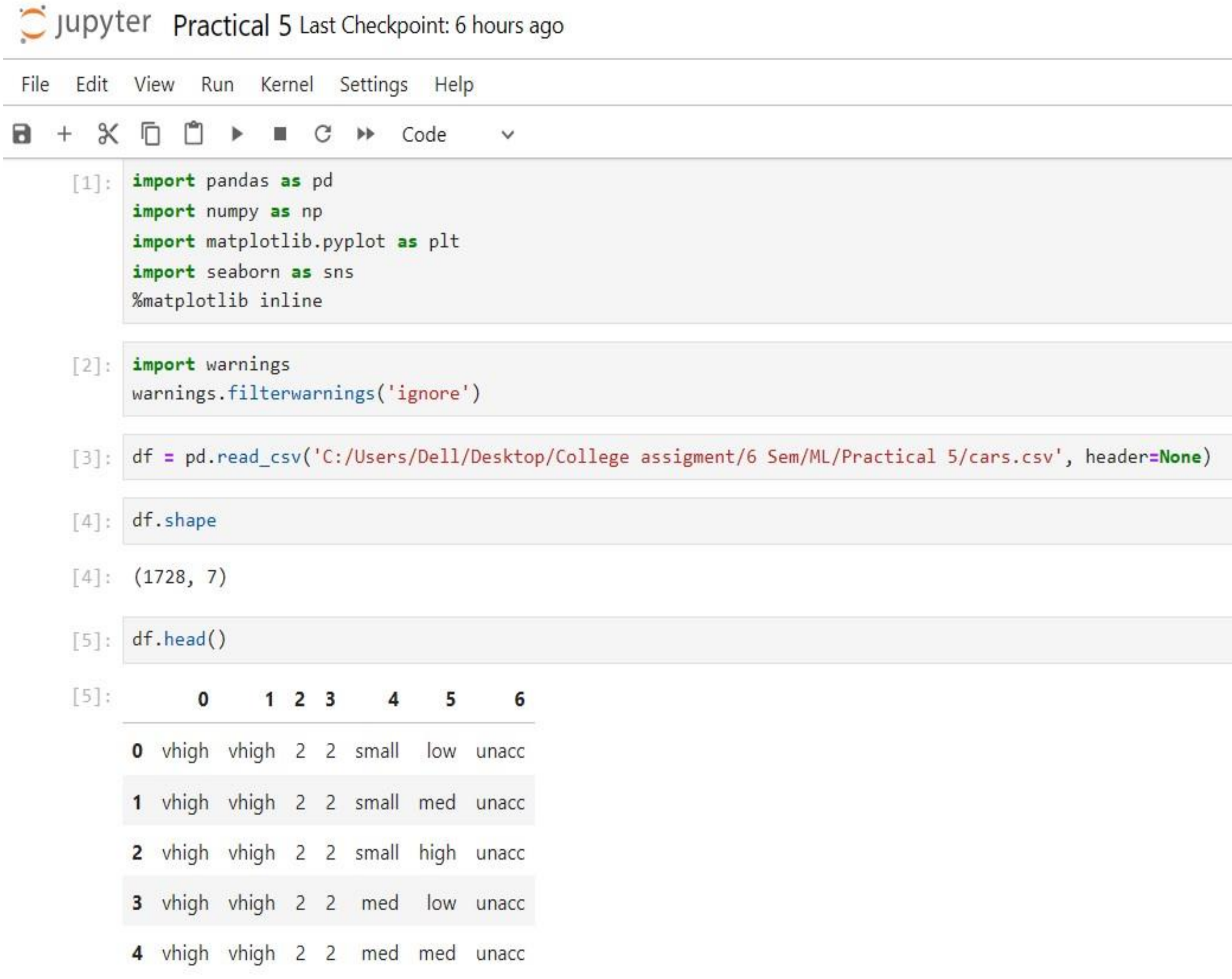


```
[1]: import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as sns
     %matplotlib inline
```

```
[2]: import warnings
     warnings.filterwarnings('ignore')
```

```
[3]: df = pd.read_csv('C:/Users/Dell/Desktop/College assigment/6 Sem/ML/Practical 5/cars.csv', header=None)
```

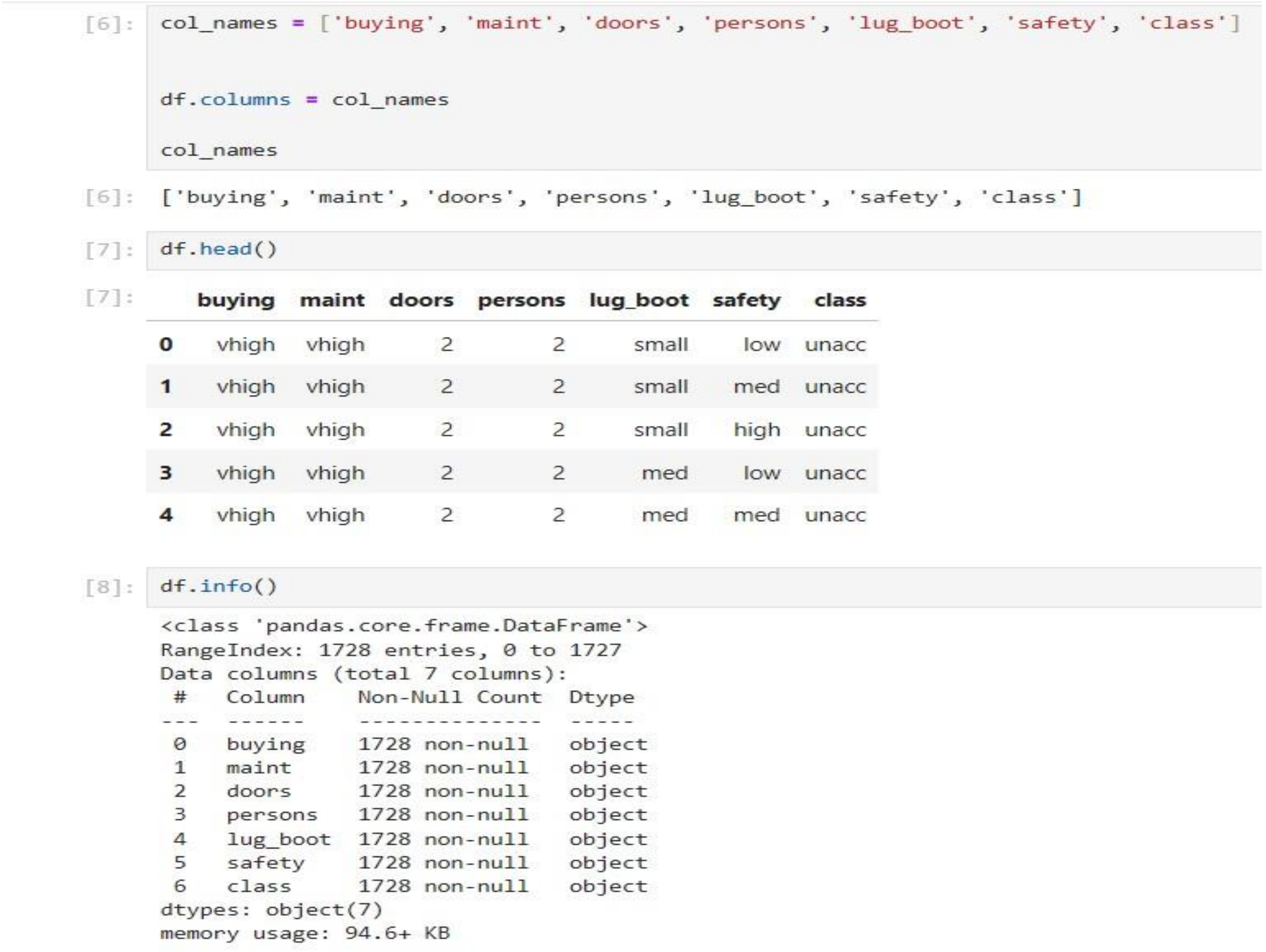
```
[4]: df.shape
```

```
[4]: (1728, 7)
```

```
[5]: df.head()
```

[5]:

	0	1	2	3	4	5	6
0	vhigh	vhigh	2	2	small	low	unacc
1	vhigh	vhigh	2	2	small	med	unacc
2	vhigh	vhigh	2	2	small	high	unacc
3	vhigh	vhigh	2	2	med	low	unacc
4	vhigh	vhigh	2	2	med	med	unacc



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


     df.columns = col_names

     col_names
```

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

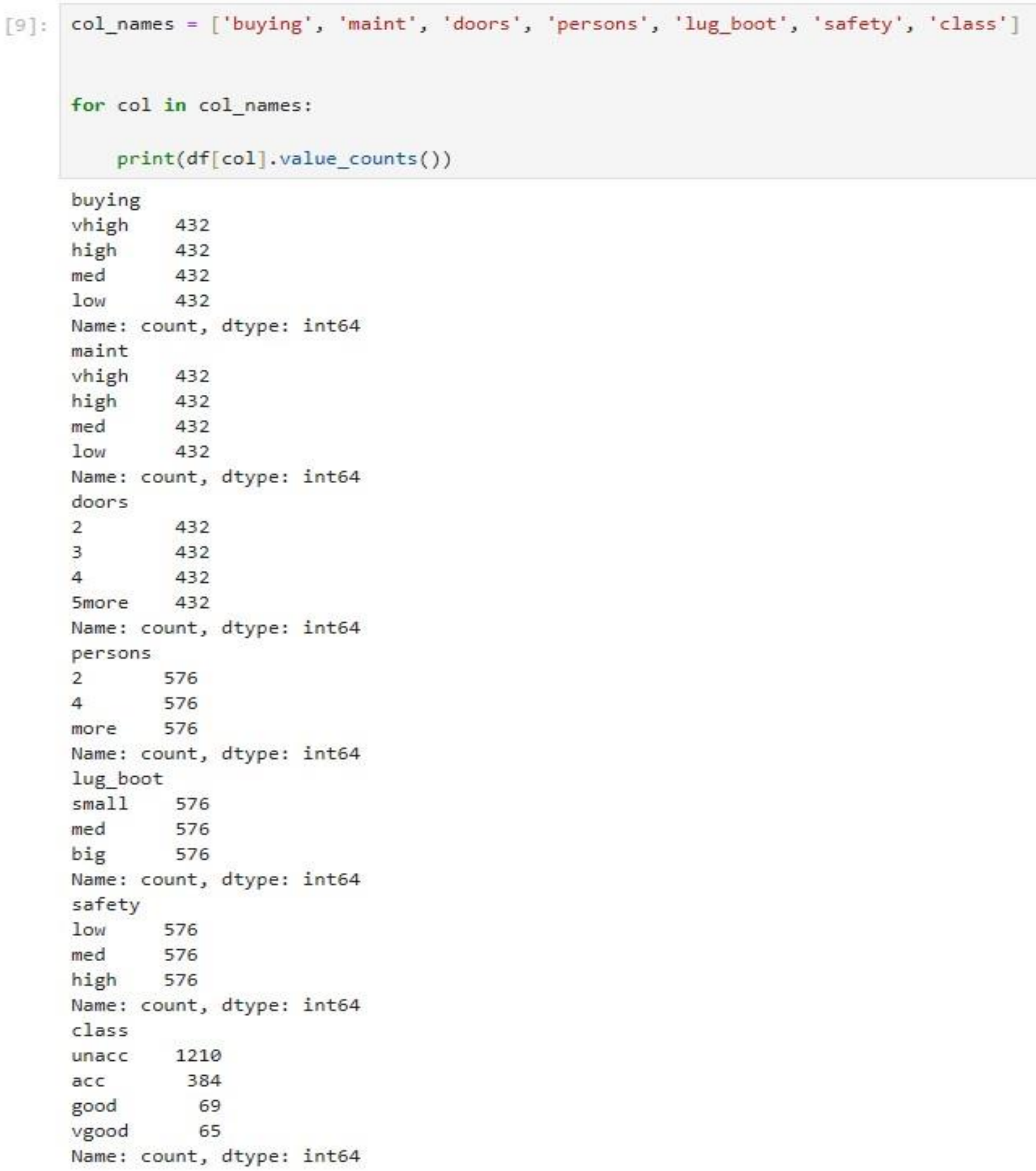
```
[7]: df.head()
```

[7]:

	buying	maint	doors	persons	lug_boot	safety	class
0	vhigh	vhigh	2	2	small	low	unacc
1	vhigh	vhigh	2	2	small	med	unacc
2	vhigh	vhigh	2	2	small	high	unacc
3	vhigh	vhigh	2	2	med	low	unacc
4	vhigh	vhigh	2	2	med	med	unacc

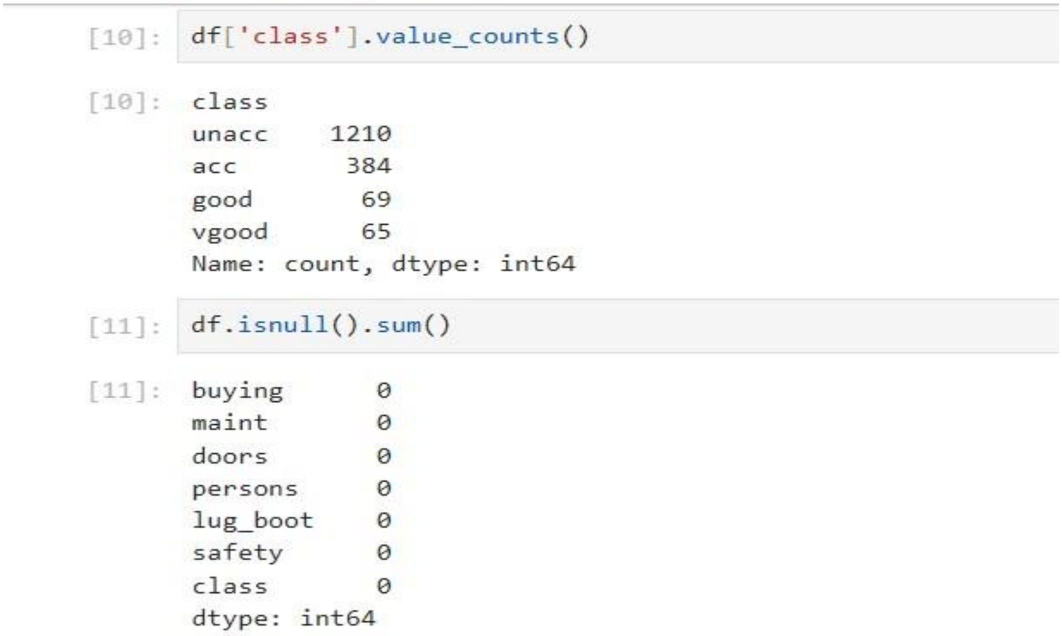
```
[8]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1728 entries, 0 to 1727
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   buying    1728 non-null   object
 1   maint     1728 non-null   object
 2   doors     1728 non-null   object
 3   persons   1728 non-null   object
 4   lug_boot  1728 non-null   object
 5   safety    1728 non-null   object
 6   class     1728 non-null   object
dtypes: object(7)
memory usage: 94.6+ KB
```



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


     for col in col_names:

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

```
buying
vhigh    432
high     432
med      432
low      432
Name: count, dtype: int64
maint
vhigh    432
high     432
med      432
low      432
Name: count, dtype: int64
doors
2        432
3        432
4        432
5more    432
Name: count, dtype: int64
persons
2       576
4       576
more    576
Name: count, dtype: int64
lug_boot
small    576
med      576
big      576
Name: count, dtype: int64
safety
low     576
med     576
high    576
Name: count, dtype: int64
class
unacc    1210
acc       384
good       69
vgood      65
Name: count, dtype: int64
```

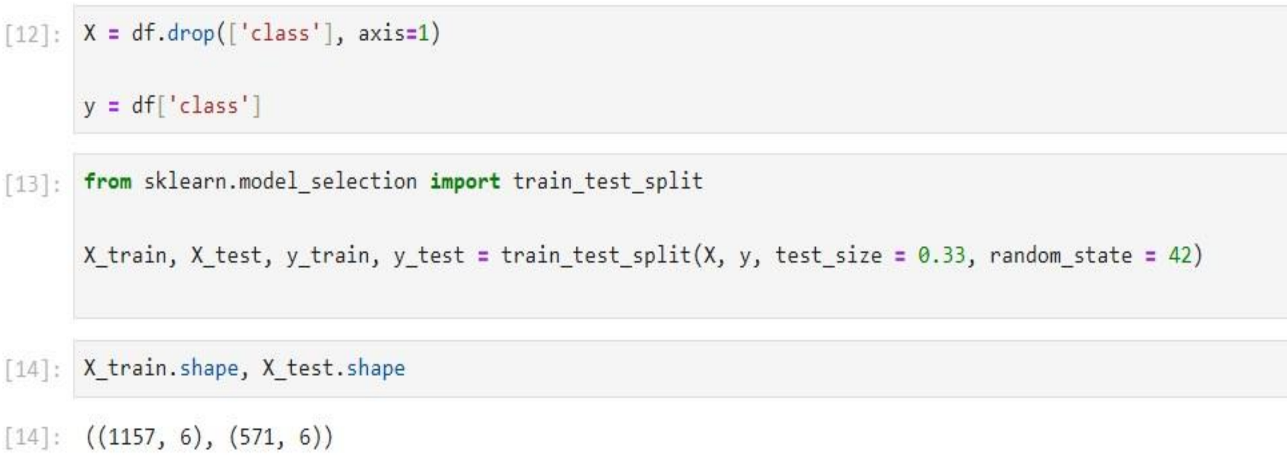


```
[10]: df['class'].value_counts()
```

```
[10]: class
      unacc    1210
      acc       384
      good       69
      vgood      65
      Name: count, dtype: int64
```

```
[11]: df.isnull().sum()
```

```
[11]: buying      0
      maint       0
      doors       0
      persons     0
      lug_boot    0
      safety      0
      class       0
      dtype: int64
```

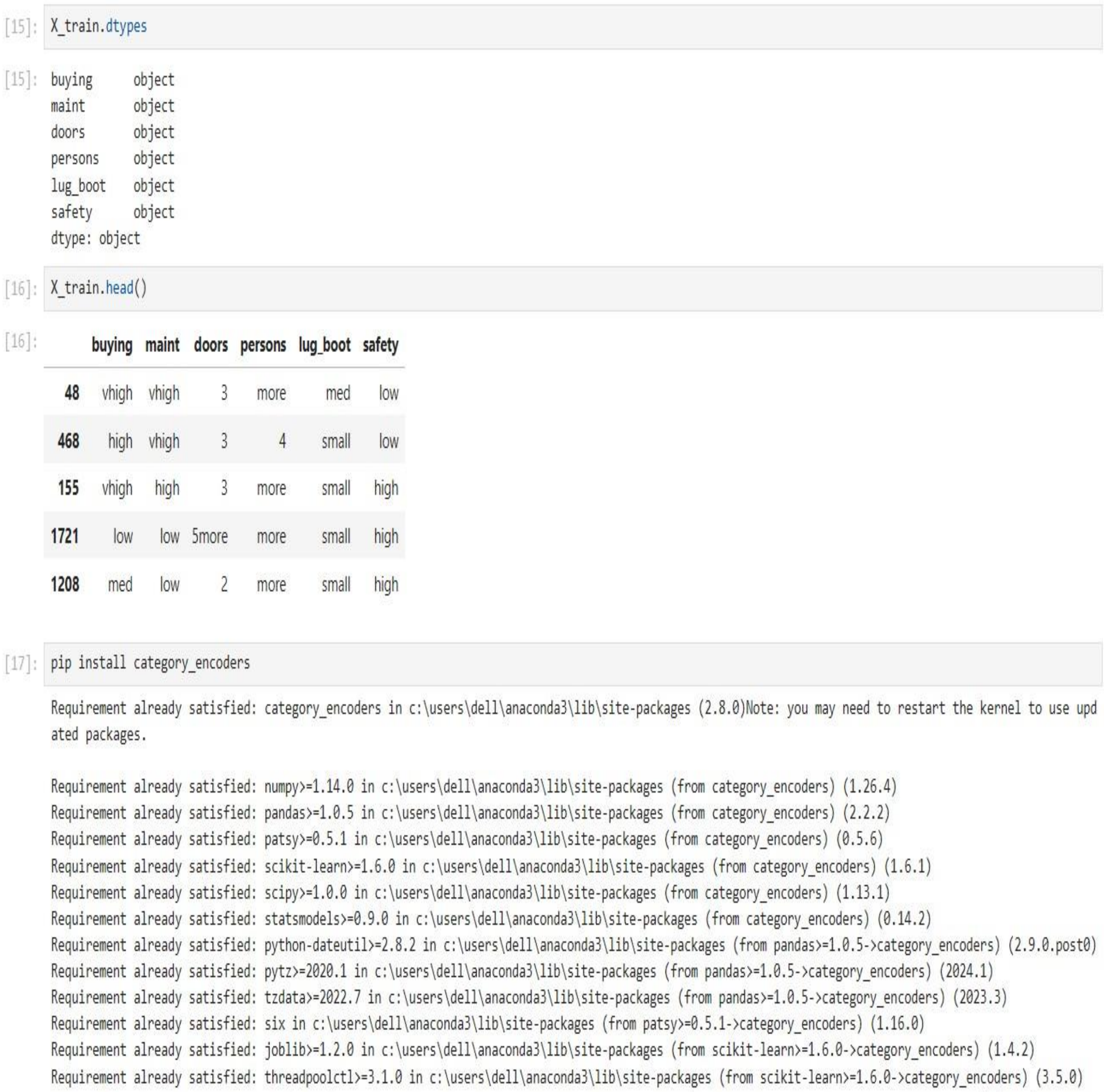


```
[12]: X = df.drop(['class'], axis=1)

      y = df['class']
```

```
[13]: from sklearn.model_selection import train_test_split

      X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33, random_state = 42)
```

```
[14]: X_train.shape, X_test.shape
```

```
[14]: ((1157, 6), (571, 6))
```



```
[15]: X_train.dtypes
```

```
[15]: buying      object
      maint       object
      doors       object
      persons     object
      lug_boot    object
      safety      object
      dtype: object
```

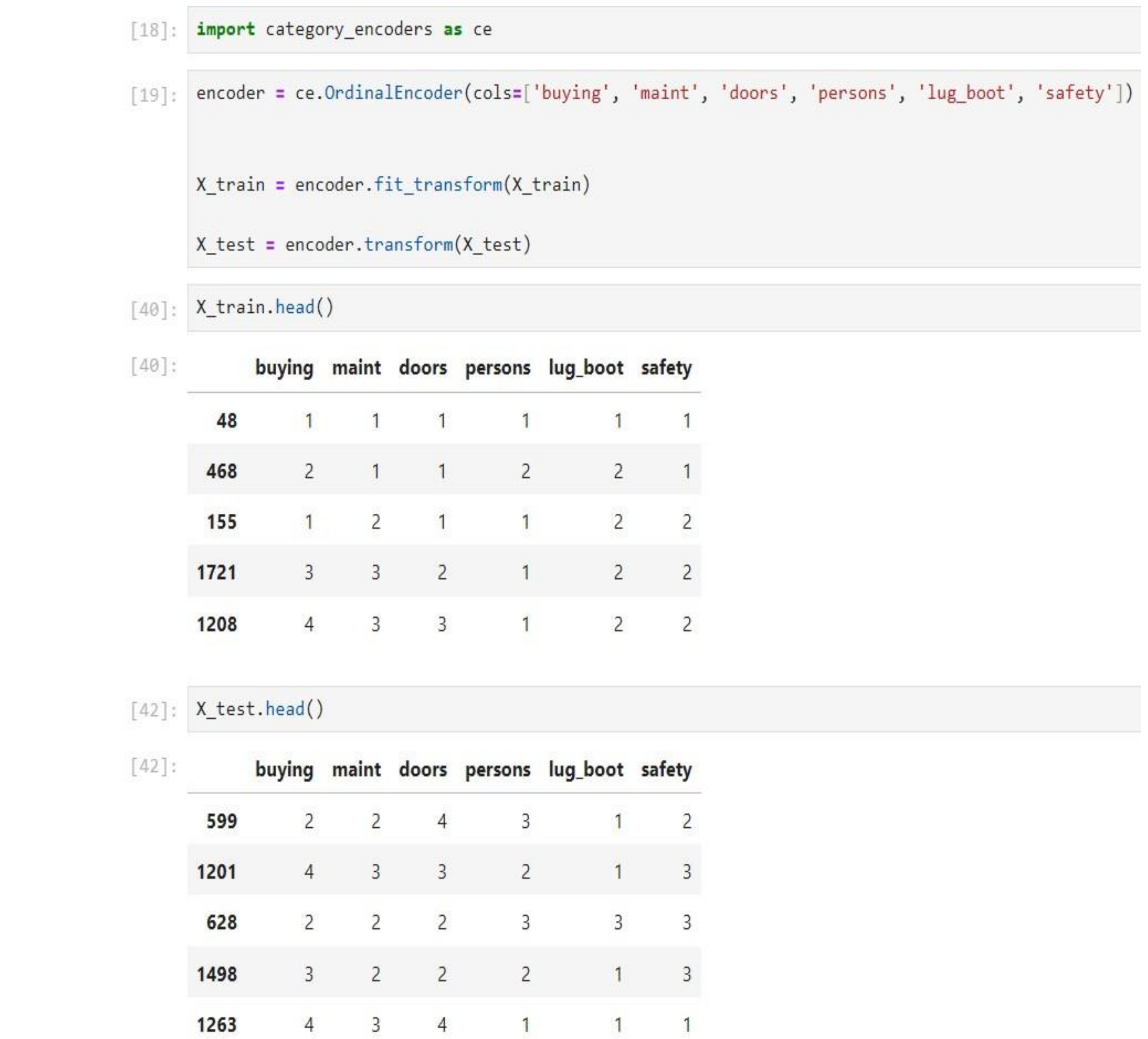
```
[16]: X_train.head()
```

[16]:

	buying	maint	doors	persons	lug_boot	safety
48	vhigh	vhigh	3	more	med	low
468	high	vhigh	3	4	small	low
155	vhigh	high	3	more	small	high
1721	low	low	5more	more	small	high
1208	med	low	2	more	small	high

```
[17]: pip install category_encoders
```

```
Requirement already satisfied: category_encoders in c:\users\dell\anaconda3\lib\site-packages (2.8.0)Note: you may need to restart the kernel to use upd
ated packages.

Requirement already satisfied: numpy>=1.14.0 in c:\users\dell\anaconda3\lib\site-packages (from category_encoders) (1.26.4)
Requirement already satisfied: pandas>=1.0.5 in c:\users\dell\anaconda3\lib\site-packages (from category_encoders) (2.2.2)
Requirement already satisfied: patsy>=0.5.1 in c:\users\dell\anaconda3\lib\site-packages (from category_encoders) (0.5.6)
Requirement already satisfied: scikit-learn>=1.6.0 in c:\users\dell\anaconda3\lib\site-packages (from category_encoders) (1.6.1)
Requirement already satisfied: scipy>=1.0.0 in c:\users\dell\anaconda3\lib\site-packages (from category_encoders) (1.13.1)
Requirement already satisfied: statsmodels>=0.9.0 in c:\users\dell\anaconda3\lib\site-packages (from category_encoders) (0.14.2)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\dell\anaconda3\lib\site-packages (from pandas>=1.0.5->category_encoders) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\dell\anaconda3\lib\site-packages (from pandas>=1.0.5->category_encoders) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\dell\anaconda3\lib\site-packages (from pandas>=1.0.5->category_encoders) (2023.3)
Requirement already satisfied: six in c:\users\dell\anaconda3\lib\site-packages (from patsy>=0.5.1->category_encoders) (1.16.0)
Requirement already satisfied: joblib>=1.2.0 in c:\users\dell\anaconda3\lib\site-packages (from scikit-learn>=1.6.0->category_encoders) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\dell\anaconda3\lib\site-packages (from scikit-learn>=1.6.0->category_encoders) (3.5.0)
```



```
[18]: import category_encoders as ce
```

```
[19]: encoder = ce.OrdinalEncoder(cols=['buying', 'maint', 'doors', 'persons', 'lug_boot', 'safety'])


      X_train = encoder.fit_transform(X_train)

      X_test = encoder.transform(X_test)
```

```
[40]: X_train.head()
```

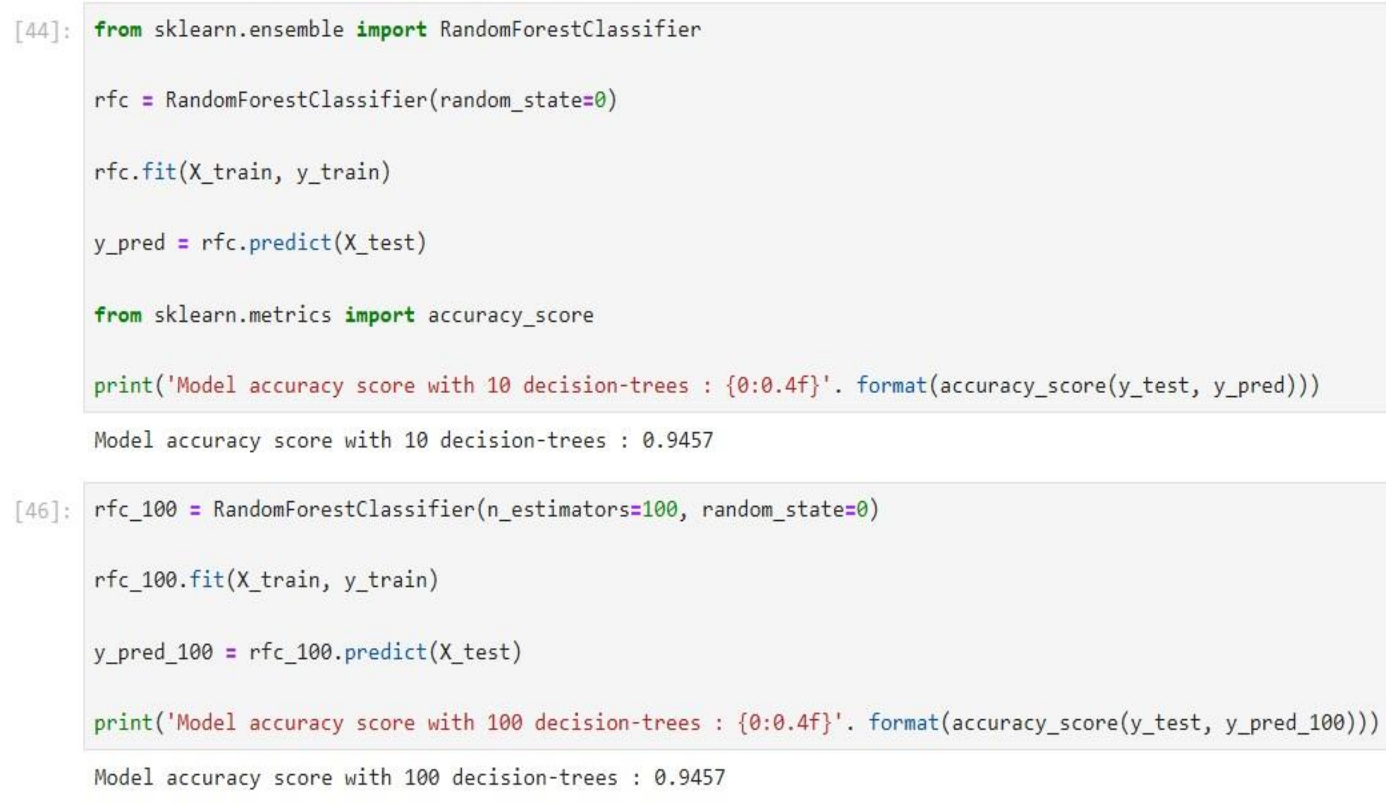
[40]:

	buying	maint	doors	persons	lug_boot	safety
48	1	1	1	1	1	1
468	2	1	1	2	2	1
155	1	2	1	1	2	2
1721	3	3	2	1	2	2
1208	4	3	3	1	2	2

```
[42]: X_test.head()
```

[42]:

	buying	maint	doors	persons	lug_boot	safety
599	2	2	4	3	1	2
1201	4	3	3	2	1	3
628	2	2	2	3	3	3
1498	3	2	2	2	1	3
1263	4	3	4	1	1	1



```
[44]: from sklearn.ensemble import RandomForestClassifier

      rfc = RandomForestClassifier(random_state=0)

      rfc.fit(X_train, y_train)

      y_pred = rfc.predict(X_test)

      from sklearn.metrics import accuracy_score

      print('Model accuracy score with 10 decision-trees : {0:0.4f}'. format(accuracy_score(y_test, y_pred)))
```

```
Model accuracy score with 10 decision-trees : 0.9457
```

```
[46]: rfc_100 = RandomForestClassifier(n_estimators=100, random_state=0)

      rfc_100.fit(X_train, y_train)

      y_pred_100 = rfc_100.predict(X_test)

      print('Model accuracy score with 100 decision-trees : {0:0.4f}'. format(accuracy_score(y_test, y_pred_100)))
```

```
Model accuracy score with 100 decision-trees : 0.9457
```