campusx-official / 100-days-of-machine-learning Public

Code Issues 3 Pull requests Actions Projects Security Insights

main

100-days-of-machine-learning / day29-sklearn-pipelines / titanic-without-using-pipeline.ipynb

campusx-official Add files via upload History

1 contributor

```
1482 lines (1482 sloc) | 38 KB
```

In [34]:

```
import numpy as np
import pandas as pd

from sklearn.model_selection import train_test_sp
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import MinMaxScaler
from sklearn.tree import DecisionTreeClassifier
```

In [35]:

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

In [36]:

```
df.head()
```

Out[36]:

	PassengerId	Survived	Pclass	Name	Sex	Age	S
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	
4	5	0	3	Allen, Mr. William Henry	male	35.0	

In [37]:

```
df.drop(columns=['PassengerId','Name','Ticket','C
```

In [38]:

```
df.head()
```

Out[38]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Er
0	0	3	male	22.0	1	0	7.2500	
1	1	1	female	38.0	1	0	71.2833	
2	1	3	female	26.0	0	0	7.9250	

3	1	1	female	35.0	1	0	53.1000
4	0	3	male	35.0	0	0	8.0500

In [39]:

```
# Step 1 -> train/test/split
X_train,X_test,y_train,y_test = train_test_split(



                                                r
```

In [40]:

```
X_train.head(2)
```

Out[40]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
331	1	male	45.5	0	0	28.5	S
733	2	male	23.0	0	0	13.0	S

In [41]:

```
y_train.head()
```

Out[41]:

```
331    0
733    0
382    0
704    0
813    0
Name: Survived, dtype: int64
```

In [42]:

```
df.isnull().sum()
```

Out[42]:

```
Survived      0
Pclass        0
Sex           0
Age         177
SibSp         0
Parch         0
Fare          0
Embarked      2
dtype: int64
```

In [43]:

```
# Applying imputation

si_age = SimpleImputer()
si_embarked = SimpleImputer(strategy='most_freque

X_train_age = si_age.fit_transform(X_train[['Age'
X_train_embarked = si_embarked.fit_transform(X_tr

X_test_age = si_age.transform(X_test[['Age']])
X_test_embarked = si_embarked.transform(X_test[['
```

In [45]:

```
X_train_embarked
```

```
Out[45]: array([['S'],
               ['S'],
               ['S'],
               ['S'],
               ['S'],
               ['C'],
               ['S'],
               ['C'],
               ['S'],
               ['S'],
               ['S'],
               ['S'],
               ['S'],
               ['S'],
               ['S'],
               ['S'],
               ['S'],
               ['C'],
               ['S'],
               ['S'],
               ['S'],
               ['S'],
               ['S'],
               ['S'],
               ['S'],
               ['C'],
               ['C'],
               ['S'],
               ['S'],
               ['C'],
               ['S'],
               ['C'],
               ['S'],
               ['Q'],
               ['Q'],
               ['S'],
               ['S'],
               ['S'],
               ['C'],
               ['S'],
               ['S'],
               ['S'],
               ['S'],
               ['S'],
               ['S'],
               ['S'],
               ['S'],
               ['S'],
               ['S'],
               ['S'],
               ['S'],
               ['S'],
               ['C'],
               ['Q'],
               ['C'],
               ['S'],
               ['S'],
               ['S'],
               ['S'],
       ['C'],
       ['Q'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['Q'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['Q'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['Q'],
       ['C'],
       ['C'],
       ['S'],
       ['S'],
       ['Q'],
       ['C'],
       ['C'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['Q'],
       ['S'],
       ['Q'],
       ['C'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['Q'],
       ['S'],
       ['C'],
       ['S'],
       ['C'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['Q']
       ['Q'],
       ['S'],
       ['S'],
       ['S'],
       ['Q'],
       ['S'],
       ['C'],
       ['Q'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['C'],
       ['Q'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['C'],
       ['C'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['Q'],
       ['C'],
       ['C'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['Q'],
       ['C'],
       ['S'],
       ['C'],
       ['Q'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['Q'],
       ['S'],
       ['S'],
       ['S'],
['C'],
['C'],
['S'],
['S'],
['S'],
['S'],
['C'],
['S'],
['S'],
['S'],
['S'],
['S'],
['S'],
['S'],
['S'],
['C'],
['S'],
['S'],
['S'],
['S'],
['S'],
['S'],
['Q'],
['Q'],
['S'],
['S'],
['S'],
['S'],
['S'],
['S'],
['S'],
['C'],
['C'],
['Q'],
['S'],
['S'],
['S'],
['S'],
['C'],
['S'],
['C'],
['Q'],
['Q'],
['C'],
['Q'],
['S'],
['C'],
['S'],
['C'],
['S'],
['Q'],
['S'],
['S'],
['S'],
['S'],
['S'],
['S'],
['S'],
['C'],
['S'],
       ['S'],
       ['C'],
       ['S'],
       ['C'],
       ['C'],
       ['Q'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['Q'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['C'],
       ['S'],
       ['S'],
       ['Q'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['Q'],
       ['S'],
       ['S'],
       ['Q'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['Q'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['Q'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['Q'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['Q'],
       ['Q'],
       ['S'],
       ['C'],
       ['S'],
       ['Q'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['Q'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['C'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['Q'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['Q'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['C'],
       ['S'],
       ['Q'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['Q'],
       ['S'],
       ['S'],
       ['C'],
       ['Q'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['Q'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['Q'],
       ['C'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['Q'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['Q'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['Q'],
       ['Q'],
       ['S'],
       ['Q'],
       ['Q'],
       ['Q'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['C'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['Q'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['C'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['Q'],
       ['Q'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['Q'],
       ['S'],
       ['S'],
       ['Q'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['Q'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['C'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['Q'],
       ['Q'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['C'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S'],
       ['S']], dtype=object)
```

In [46]:

```
# one hot encoding Sex and Embarked

ohe_sex = OneHotEncoder(sparse=False,handle_unkno
ohe_embarked = OneHotEncoder(sparse=False,handle_

X_train_sex = ohe_sex.fit_transform(X_train[['Sex
X_train_embarked = ohe_embarked.fit_transform(X_t

X_test_sex = ohe_sex.transform(X_test[['Sex']])
```

```
X_test_embarked = ohe_embarked.transform(X_test_e
```

In [48]:

```
X_train_embarked
```

Out[48]:

```
array([[0., 0., 1.],
       [0., 0., 1.],
       [0., 0., 1.],
       ...,
       [0., 0., 1.],
       [0., 0., 1.],
       [0., 0., 1.]])
```

In [18]:

```
X_train.head(2)
```

Out[18]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
331	1	male	45.5	0	0	28.5	S
733	2	male	23.0	0	0	13.0	S

In [19]:

```
X_train_rem = X_train.drop(columns=['Sex','Age','
```

In [20]:

```
X_test_rem = X_test.drop(columns=['Sex','Age','Em
```

In [49]:

```
X_train_transformed = np.concatenate((X_train_rem
X_test_transformed = np.concatenate((X_test_rem,X_
```

In [52]:

```
X_test_transformed.shape
```

Out[52]:

```
(179, 10)
```

In [53]:

```
clf = DecisionTreeClassifier()
clf.fit(X_train_transformed,y_train)
```

Out[53]:

```
DecisionTreeClassifier()
```

In [54]:

```
y_pred = clf.predict(X_test_transformed)
y_pred
```

Out[54]:

```
array([0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0,
0, 1, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0,
1, 1, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0,
1, 0, 1, 1, 1, 0, 0, 1,
       0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1,
0, 1, 1, 0, 0, 0, 1, 1,
       0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0,
1, 0, 0, 1, 1, 0, 0, 0,
1, 0, 0, 1, 1, 0, 0, 0,
       1  0  1  1  0  0  0  0  1  0  0  1  1  1
```