RandomForestClassifier

January 3, 2023

[1]:
```
import pandas as pd
import matplotlib as plt
import numpy as np
```

[2]:
```
df = pd.read_csv('basic_info.csv')
df
```

[2]:

	Unnamed: 0	ID	Name	Age	Nationality	Overall	Potential	Club	Contract	Value	Wage	Total stat
0	0	236988	Eddie Nketiah	22	England	72	79	Arsenal	2016 ~ 2022	€4.8M	€45K	1698
1	1	225863	Olivier Boscagli	23	France	77	82	PSV	2019 ~ 2025	€14.5M	€15K	1961
2	2	241721	Rafael da Conceição Leão	22	Portugal	82	90	AC Milan	2019 ~ 2024	€68.5M	€52K	1959
3	3	224371	Jarrod Bowen	24	England	79	82	West Ham United	2020 ~ 2025	€24M		
4	4	200104	Heung Min Son	28	Korea Republic	89	89	Tottenham Hotspur	2015 ~ 2025	€104M		
…	…	…	…	…	…	…	…	…	…	…		
19820	20036	245534	Carl Spellman	18	England	52	64	Tranmere Rovers	2018 ~ 2020	€80K		
19821	20037	245535	Abdulkadir Parmak	26	Turkey	71	74	Yukatel Kayserispor	May 31, 2022 On Loan	€2.2M		
19822	20038	245536	Andrea Errico	19	Italy	57	70	Frosinone	2018 ~ 2021	€200K		
19823	20039	245540	Stratos Svarnas	23	Greece	69	74	AEK Athens	2018 ~ 2025	€1.9M		
19824	20097	259356	Carney Chukwuemeka	17	England	64	87	Aston Villa	2021 ~ 2023	€1.8M		

```
3       €63K         1966
4      €240K         2141
…        …            …
19820   €500         1506
19821   €15K         2008
19822    €2K         1385
19823   €500         1589
19824    €2K         1630

[19825 rows x 12 columns]
```

[3]:
```
df = df.drop('Unnamed: 0', axis=1)
df
```

[3]:
```
           ID                      Name  Age     Nationality  Overall  \
0      236988             Eddie Nketiah   22         England       72
1      225863          Olivier Boscagli   23          France       77
2      241721  Rafael da Conceição Leão   22        Portugal       82
3      224371              Jarrod Bowen   24         England       79
4      200104             Heung Min Son   28  Korea Republic       89
…       …                         …  …            …      …
19820  245534             Carl Spellman   18         England       52
19821  245535         Abdulkadir Parmak   26          Turkey       71
19822  245536             Andrea Errico   19           Italy       57
19823  245540           Stratos Svarnas   23          Greece       69
19824  259356        Carney Chukwuemeka   17         England       64

       Potential                Club             Contract   Value   Wage  \
0             79             Arsenal          2016 ~ 2022   €4.8M   €45K
1             82                 PSV          2019 ~ 2025  €14.5M   €15K
2             90            AC Milan          2019 ~ 2024  €68.5M   €52K
3             82     West Ham United          2020 ~ 2025    €24M   €63K
4             89   Tottenham Hotspur          2015 ~ 2025   €104M  €240K
…           …                  …                  …     …    …
19820         64     Tranmere Rovers          2018 ~ 2020    €80K   €500
19821         74  Yukatel Kayserispor  May 31, 2022 On Loan   €2.2M   €15K
19822         70           Frosinone          2018 ~ 2021   €200K    €2K
19823         74          AEK Athens          2018 ~ 2025   €1.9M   €500
19824         87          Aston Villa          2021 ~ 2023   €1.8M    €2K

       Total stat
0            1698
1            1961
2            1959
3            1966
4            2141
…             …
```

```
19820        1506
19821        2008
19822        1385
19823        1589
19824        1630

[19825 rows x 11 columns]
```

[4]:
```
is_spanish = []
for rows in df['Nationality']:
    if rows == 'Spain':
        is_spanish.append(1)
    else:
        is_spanish.append(0)
len(is_spanish)
```

[4]:
```
19825
```

[5]:
```
df['is_spanish'] = is_spanish
df
```

[5]:

	ID	Name	Age	Nationality	Overall	\
0	236988	Eddie Nketiah	22	England	72	
1	225863	Olivier Boscagli	23	France	77	
2	241721	Rafael da Conceição Leão	22	Portugal	82	
3	224371	Jarrod Bowen	24	England	79	
4	200104	Heung Min Son	28	Korea Republic	89	
…	…	…	…	…	…	
19820	245534	Carl Spellman	18	England	52	
19821	245535	Abdulkadir Parmak	26	Turkey	71	
19822	245536	Andrea Errico	19	Italy	57	
19823	245540	Stratos Svarnas	23	Greece	69	
19824	259356	Carney Chukwuemeka	17	England	64	

	Potential	Club	Contract	Value	Wage	\
0	79	Arsenal	2016 ~ 2022	€4.8M	€45K	
1	82	PSV	2019 ~ 2025	€14.5M	€15K	
2	90	AC Milan	2019 ~ 2024	€68.5M	€52K	
3	82	West Ham United	2020 ~ 2025	€24M	€63K	
4	89	Tottenham Hotspur	2015 ~ 2025	€104M	€240K	
…	…	…	…	…	…	
19820	64	Tranmere Rovers	2018 ~ 2020	€80K	€500	
19821	74	Yukatel Kayserispor	May 31, 2022 On Loan	€2.2M	€15K	
19822	70	Frosinone	2018 ~ 2021	€200K	€2K	
19823	74	AEK Athens	2018 ~ 2025	€1.9M	€500	
19824	87	Aston Villa	2021 ~ 2023	€1.8M	€2K	

```
       Total stat  is_spanish
0            1698           0
1            1961           0
2            1959           0
3            1966           0
4            2141           0
…             …           …
19820        1506           0
19821        2008           0
19822        1385           0
19823        1589           0
19824        1630           0

[19825 rows x 12 columns]
```

[6]:
```
df = df.drop(['Contract', 'Name', 'Nationality'], axis=1)
df
```

[6]:
```
            ID  Age  Overall  Potential                Club   Value   Wage  \
0       236988   22       72         79             Arsenal   €4.8M   €45K
1       225863   23       77         82                 PSV  €14.5M   €15K
2       241721   22       82         90           AC Milan  €68.5M   €52K
3       224371   24       79         82    West Ham United    €24M   €63K
4       200104   28       89         89  Tottenham Hotspur   €104M  €240K
…          …   …        …          …                  …      …     …
19820   245534   18       52         64    Tranmere Rovers    €80K   €500
19821   245535   26       71         74  Yukatel Kayserispor  €2.2M   €15K
19822   245536   19       57         70           Frosinone   €200K    €2K
19823   245540   23       69         74          AEK Athens  €1.9M   €500
19824   259356   17       64         87        Aston Villa   €1.8M    €2K

       Total stat  is_spanish
0            1698           0
1            1961           0
2            1959           0
3            1966           0
4            2141           0
…             …           …
19820        1506           0
19821        2008           0
19822        1385           0
19823        1589           0
19824        1630           0

[19825 rows x 9 columns]
```

[7]:
```
def clean_data_money(df):
    temp_data = df.str.replace('[M, K, €]', '').astype(float)
    for values, indx in zip(df, df.index):
        if values[-1] == 'M':
            temp_data[indx] = temp_data[indx] * 1000000
        elif values[-1] == 'K':
            temp_data[indx] = temp_data[indx] * 1000
        else:
            pass
    return temp_data
df['Value'] = clean_data_money(df['Value'])
df['Wage'] = clean_data_money(df['Wage'])
df
```

```
/tmp/ipykernel_74338/3880873135.py:2: FutureWarning: The default value of regex
will change from True to False in a future version.
  temp_data = df.str.replace('[M, K, €]', '').astype(float)
```

[7]:

	ID	Age	Overall	Potential	Club	Value	Wage	Total stat	is_spanish
0	236988	22	72	79	Arsenal	4800000.0	45000.0	1698	0
1	225863	23	77	82	PSV	14500000.0	15000.0	1961	0
2	241721	22	82	90	AC Milan	68500000.0	52000.0	1959	0
3	224371	24	79	82	West Ham United	24000000.0	63000.0	1966	0
4	200104	28	89	89	Tottenham Hotspur	104000000.0	240000.0	2141	0
…	…	…	…	…	…	…	…	…	…
19820	245534	18	52	64	Tranmere Rovers	80000.0	500.0	1506	0
19821	245535	26	71	74	Yukatel Kayserispor	2200000.0	15000.0	2008	0
19822	245536	19	57	70	Frosinone	200000.0	2000.0	1385	0
19823	245540	23	69	74	AEK Athens	1900000.0	500.0	1589	0
19824	259356	17	64	87	Aston Villa	1800000.0	2000.0	1630	0

[19825 rows x 9 columns]

[8]:
```
X = df.drop('is_spanish', axis=1)
y = df['is_spanish']
```

[9]:
```
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer

categorical_features = ['Club'] # columnas que queremos transformar en nums
one_hot = OneHotEncoder()
transformer = ColumnTransformer([('one_hot',
                                  one_hot,
                                  categorical_features)], # aqui le pasamos la␣
 ↪columnas
                                  remainder='passthrough') # esto es para no␣
 ↪alterar el resto de columnas
transformed_X = transformer.fit_transform(X)
transformed_X
```

[9]:
```
<19825x940 sparse matrix of type '<class 'numpy.float64'>'
        with 157896 stored elements in Compressed Sparse Row format>
```

[10]:
```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier

X_train, X_test, y_train, y_test = train_test_split(transformed_X, y,␣
 ↪test_size=0.2)

model = RandomForestClassifier()
model.fit(X_train, y_train)
```

[10]:
```
RandomForestClassifier()
```

[11]:
```
model.score(X_test, y_test)
```

[11]:
```
0.9621689785624212
```

[14]:
```
y_pred = model.predict(X_test)
y_pred
```

[14]:
```
array([0, 0, 0, …, 0, 0, 0])
```

[12]:
```
from sklearn.model_selection import cross_validate

scoring = {'acc': 'accuracy',
           'prec_macro': 'precision_macro',
           'rec_micro': 'recall_macro'}
scores = cross_validate(model, transformed_X, y, scoring=scoring,
                        cv=5, return_train_score=True)
```

```
print(scores.keys())
print(scores['test_acc'])
```

```
dict_keys(['fit_time', 'score_time', 'test_acc', 'train_acc', 'test_prec_macro',
'train_prec_macro', 'test_rec_micro', 'train_rec_micro'])
[0.93139975 0.96343001 0.95889029 0.96544767 0.95863808]
```

[26]:
```
scores
```

[26]:
```
{'fit_time': array([2.65853691, 3.59602189, 2.97926378, 3.02867246,
3.64138937]),
 'score_time': array([0.08427024, 0.09197307, 0.06920719, 0.07654119,
0.08398628]),
 'test_acc': array([0.93139975, 0.96343001, 0.95889029, 0.96544767,
0.95863808]),
 'train_acc': array([1.        , 1.        , 1.        , 0.99993695,
0.99993695]),
 'test_prec_macro': array([0.69216933, 0.84438022, 0.89000583, 0.89313257,
0.88987633]),
 'train_prec_macro': array([1.        , 1.        , 1.        , 0.99996654,
0.99996654]),
 'test_rec_micro': array([0.71559908, 0.80023178, 0.68304669, 0.75621184,
0.6822391 ]),
 'train_rec_micro': array([1.        , 1.        , 1.        , 0.99945474,
0.99945415])}
```

[27]:
```
for key, values in scores.items():
    print(f'{key} --> {values.mean()}')
```

```
fit_time --> 3.1807768821716307
score_time --> 0.08119559288024902
test_acc --> 0.955561160151324
train_acc --> 0.9999747793190416
test_prec_macro --> 0.8419128558677207
train_prec_macro --> 0.9999866171501072
test_rec_micro --> 0.7274656989496103
train_rec_micro --> 0.9997817784402336
```

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: