code

September 12, 2021

[21]:

```
import pandas as pd
from matplotlib import pyplot as plt
import numpy as np
```

[22]:

```
df = pd.read_csv("./FIFA-21 Complete.csv")
df.head()
```

[22]:

```
   player_id               name  nationality  position  overall  age  hits  \
0     158023       Lionel Messi    Argentina  ST|CF|RW       94   33   299
1      20801  Cristiano Ronaldo     Portugal     ST|LW       93   35   276
2     190871          Neymar Jr       Brazil    CAM|LW       92   28   186
3     203376    Virgil van Dijk  Netherlands        CB       91   29   127
4     200389          Jan Oblak     Slovenia        GK       91   27    47

   potential                 team
0         94         FC Barcelona
1         93             Juventus
2         92  Paris Saint-Germain
3         92            Liverpool
4         93      Atlético Madrid
```

[23]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17981 entries, 0 to 17980
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   player_id    17981 non-null  int64
 1   name         17981 non-null  object
 2   nationality  17981 non-null  object
 3   position     17981 non-null  object
 4   overall      17981 non-null  int64
 5   age          17981 non-null  int64
 6   hits         17981 non-null  int64
 7   potential    17981 non-null  int64
 8   team         17981 non-null  object
```

```
dtypes: int64(5), object(4)
memory usage: 1.2+ MB
```

```
[24]: X = df[["overall", "age", "hits", "potential"]]
```

```
[25]: from sklearn.preprocessing import MinMaxScaler
      X_sca = MinMaxScaler()
      X = X_sca.fit_transform(X)
```

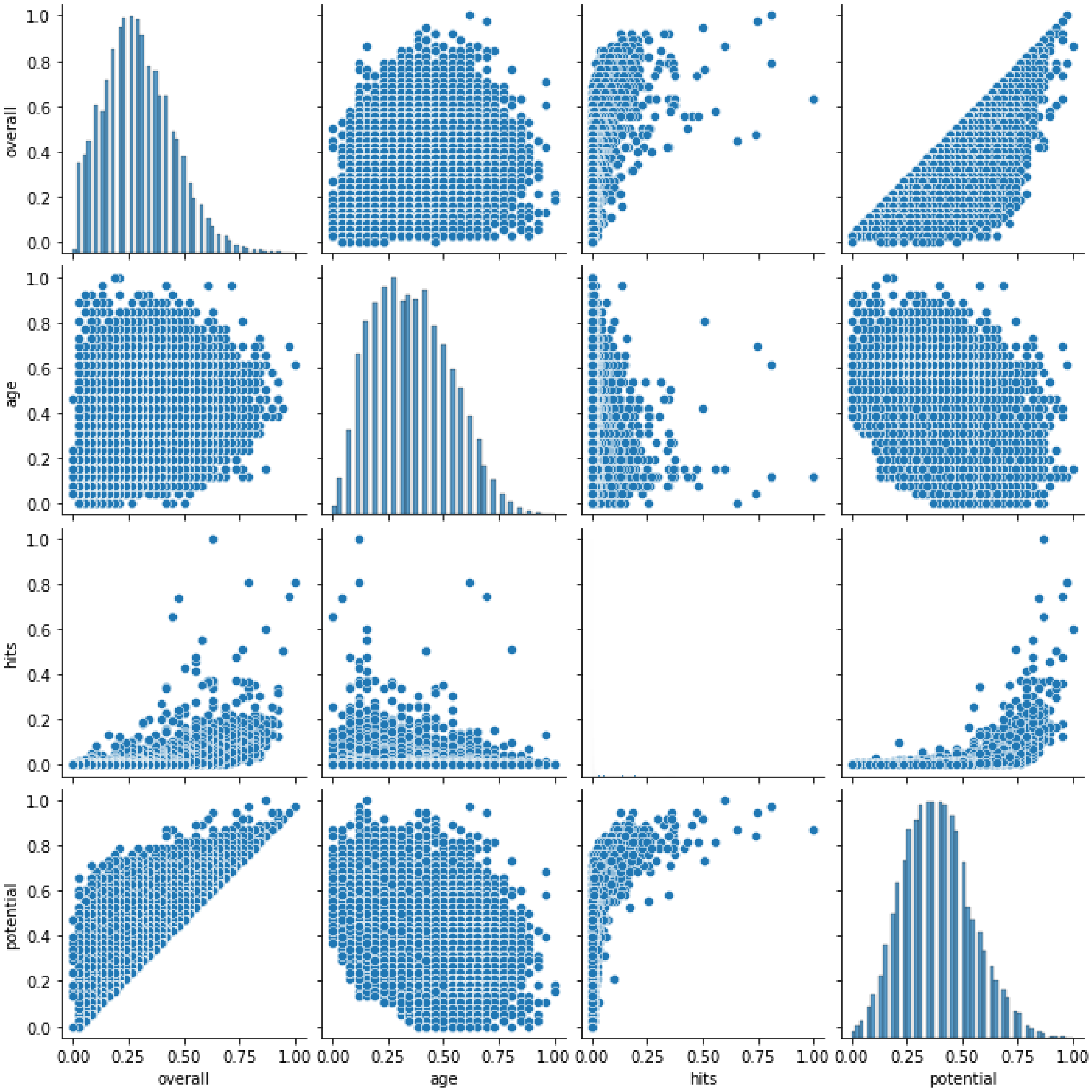
```
[26]: dfq = pd.DataFrame(data=X, columns=["overall", "age", "hits", "potential"])
```

```
[27]: dfq.describe()
```

```
[27]:             overall           age          hits     potential
      count  17981.000000  17981.000000  17981.000000  17981.000000
      mean       0.296693      0.358132      0.007249      0.387844
      std        0.155905      0.175234      0.029235      0.156894
      min        0.000000      0.000000      0.000000      0.000000
      25%        0.184211      0.230769      0.000000      0.263158
      50%        0.289474      0.346154      0.000000      0.368421
      75%        0.394737      0.500000      0.005391      0.500000
      max        1.000000      1.000000      1.000000      1.000000
```

```
[28]: import seaborn as sns
      sns.pairplot(dfq)
```

```
[28]: <seaborn.axisgrid.PairGrid at 0x7fdd880bb850>
```



[29]:

```
X = np.array(dfq["hits"])
y = np.array(dfq["potential"])
X = X.reshape(-1, 1)
y = y.reshape(-1, 1)
print(X.shape, y.shape)
```

```
(17981, 1) (17981, 1)
```

[30]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.20, random_state=0)
```

[31]:
```
from sklearn.linear_model import LinearRegression
reg = LinearRegression()
print(X_train.shape, y_train.shape)
reg.fit(X_train, y_train)
```

```
(14384, 1) (14384, 1)
```

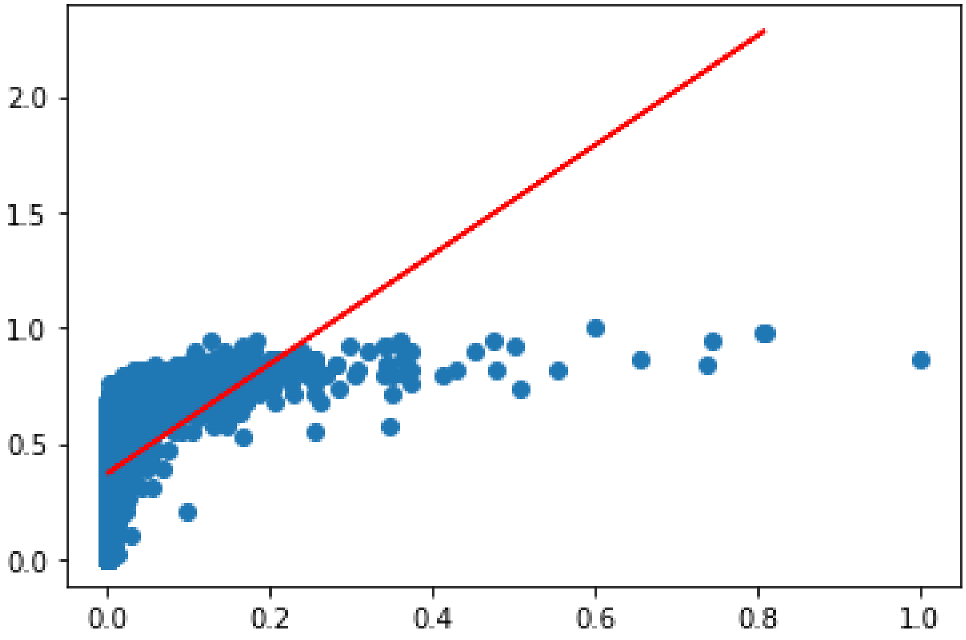
[31]:
```
LinearRegression()
```

[32]:
```
y_pred = reg.predict(X_test)
```

[33]:
```
reg.score(X_test, y_test)
```

[33]:
```
0.16111153706859516
```

[34]:
```
plt.scatter(X, y)
plt.plot(X_train, reg.predict(X_train), color="red")
plt.show()
```



[35]:
```
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
kf = KFold(n_splits=5)
model = LinearRegression()
scores = cross_val_score(model, X_train, y_train, scoring='r2', cv=kf)

print("Avg accuracy: {}".format(scores.mean()))
```

```
Avg accuracy: 0.18585985402502103
```