Python for Notebook

1

azure\_storage\_account\_name = "uwclddata230public"

azure\_storage\_account\_key = "SoDTkHaHdxAVoRQtjtsqh0gX3RgQc0oKJuAkAXaWA4Urd+9Mj7mPeaBclk0F9dOowzMYfOuS9ktfwnrrF/+IwQ=="

if azure\_storage\_account\_name is None or azure\_storage\_account\_key is None:

raise Exception("You must provide a name and key for an Azure Storage account")

2

!pip install azure-storage==0.32.0

3

from azure.storage.blob import BlockBlobService

blob\_service = BlockBlobService(azure\_storage\_account\_name, azure\_storage\_account\_key)

blob\_service.get\_blob\_to\_path('azure-notebooks-data', 'sample.txt', 'sample.txt')

!cat sample.txt

4

blob\_service.get\_blob\_to\_path('week05hw', 'nba\_2013.csv', 'nba\_2013.csv')

!cat nba\_2013.csv

5

import pandas

nba = pandas.read\_csv("nba\_2013.csv")

6

nba.shape

7

nba.head(1)

8

nba.mean()

9

import seaborn as sns

import matplotlib.pyplot as plt

sns.pairplot(nba[["ast", "fg", "trb"]])

plt.show()

10

from sklearn.cluster import KMeans

kmeans\_model = KMeans(n\_clusters=5, random\_state=1)

good\_columns = nba.\_get\_numeric\_data().dropna(axis=1)

kmeans\_model.fit(good\_columns)

labels = kmeans\_model.labels\_

11

from sklearn.decomposition import PCA

pca\_2 = PCA(2)

plot\_columns = pca\_2.fit\_transform(good\_columns)

plt.scatter(x=plot\_columns[:,0], y=plot\_columns[:,1], c=labels)

plt.show()

12

train = nba.sample(frac=0.8, random\_state=1)

test = nba.loc[~nba.index.isin(train.index)]

13

from sklearn.linear\_model import LinearRegression

lr = LinearRegression()

lr.fit(train[["fg"]], train["ast"])

predictions = lr.predict(test[["fg"]])

14

import statsmodels.formula.api as sm

model = sm.ols(formula='ast ~ fga', data=train)

fitted = model.fit()

fitted.summary()

15

from sklearn.ensemble import RandomForestRegressor

predictor\_columns = ["age", "mp", "fg", "trb", "stl", "blk"]

rf = RandomForestRegressor(n\_estimators=100, min\_samples\_leaf=3)

rf.fit(train[predictor\_columns], train["ast"])

predictions = rf.predict(test[predictor\_columns])

16

from sklearn.metrics import mean\_squared\_error

mean\_squared\_error(test["ast"], predictions)

17

import requests

url = "http://www.basketball-reference.com/boxscores/201506140GSW.html"

data = requests.get(url).content

18

from bs4 import BeautifulSoup

import re

soup = BeautifulSoup(data, 'html.parser')

box\_scores = []

for tag in soup.find\_all(id=re.compile("[A-Z]{3,}\_basic")):

rows = []

for i, row in enumerate(tag.find\_all("tr")):

if i == 0:

continue

elif i == 1:

tag = "th"

else:

tag = "td"

row\_data = [item.get\_text() for item in row.find\_all(tag)]

rows.append(row\_data)

box\_scores.append(rows)

19

data