Homework02

- API
- Sklearn ml (mtcars)

```
# free : https://mixedanalytics.com/blog/list-actually-free-open-no-auth-needed-apis/
import pandas as pd
import requests

url = " https://statsapi.web.nhl.com/api/v1/teams"

result = requests.get(url)
data_json = result.json()
```

```
#Check keys
data_json.keys()
```

```
dict_keys(['copyright', 'teams'])
```

```
#Check type of keys
print(type(data_json["copyright"]))
print(type(data_json["teams"]))
```

```
<class 'str'>
<class 'list'>
```

```
#Dataframe
nhl_team = pd.DataFrame(data_json["teams"], columns=["name", "abbreviation", "teamName", "locationName", "firstYearOfPlay"] )
```

```
nhl_team.head(10)
```

	name	abbreviation	teamName	locationName	firstYearOfPlay
0	New Jersey Devils	NJD	Devils	New Jersey	1982
1	New York Islanders	NYI	Islanders	New York	1972
2	New York Rangers	NYR	Rangers	New York	1926
3	Philadelphia Flyers	PHI	Flyers	Philadelphia	1967
4	Pittsburgh Penguins	PIT	Penguins	Pittsburgh	1967
5	Boston Bruins	BOS	Bruins	Boston	1924
6	Buffalo Sabres	BUF	Sabres	Buffalo	1970
7	Montréal Canadiens	MTL	Canadiens	Montréal	1909
8	Ottawa Senators	OTT	Senators	Ottawa	1990
9	Toronto Maple Leafs	TOR	Maple Leafs	Toronto	1917

```
#Create .CSV file
nhl_team.to_csv("nhl_team.csv")
```

HW02 - Sklearn

```
#sklearn => machine learning most popular python
# template in sklearn to train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
import pandas as pd
import numpy as np
```

```
mtcars = pd.read_csv("https://gist.githubusercontent.com/seankross/a412dfbd88b3db70b74b/raw/5f23f993cd87c283ce766e7ac6b329ee7cc2e1d1/mtcar
mtcars.head()
```

	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
1	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
2	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
3	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
4	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2

```
#prepare
X = mtcars[ ["hp", "wt", "am"] ]
y = mtcars["mpg"]
```

```
#split data

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42 )
```

```
#LinearRegression
#train model
model = LinearRegression()
model.fit(X_train, y_train)

#test model/ scoring
pred = model.predict(X_test)

#MAE mean absolute error
mae = np.mean(np.absolute((y_test - pred)))

#MSE
mse = np.mean((y_test - pred)**2)

#RMSE
rmse = np.sqrt(mse)

print(f"LinearRegression\n MAE : {mae}\n MSE : {mse}\n RMSE : {rmse}")

```

```
LinearRegression
 MAE : 2.0710117400488417
 MSE : 7.319771016726086
 RMSE : 2.7055075340360975
```

```
#DecisionTreeRegressor
#train model
model = DecisionTreeRegressor()
model.fit(X_train, y_train)

#test model/ scoring
pred_dt = model.predict(X_test)

#MAE mean absolute error
mae_dt = np.mean(np.absolute((y_test - pred_dt)))

#MSE
mse_dt = np.mean((y_test - pred_dt)**2)

#RMSE
rmse_dt = np.sqrt(mse_dt)

print(f"DecisionTreeRegressor \n MAE : {mae_dt}\n MSE : {mse_dt}\n RMSE : {rmse_dt}")
```

```
DecisionTreeRegressor
 MAE : 2.4714285714285706
 MSE : 9.609999999999994
 RMSE : 3.099999999999999
```

```
#RandomForestRegressor
#train model
model = RandomForestRegressor()
model.fit(X_train, y_train)
```

```python

#test model/ scoring
pred_rf = model.predict(X_test)

#MAE mean absolute error
mae_rf = np.mean(np.absolute((y_test - pred_rf)))

#MSE
mse_rf = np.mean((y_test - pred_rf)**2)

#RMSE
rmse_rf = np.sqrt(mse_rf)

print(f"RandomForestRegressor \n MAE : {mae_rf}\n MSE : {mse_rf}\n RMSE : {rmse_rf}")
```

```
RandomForestRegressor 
 MAE : 2.197142857142857
 MSE : 9.25432657142855
 RMSE : 3.042092465956377
```