

knn

May 29, 2023

1 PREDICTION WITH KNN

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

```
[2]: # read dataset may2023
df = pd.read_pickle("../data/20230319_RTU_Dataset_PPC-Lab/combined_may2023.
↳pk1")
```

```
[3]: df
```

```
[3]:
```

	MEM_USAGE	CPU_USAGE	PS1_V	TEMP
0	35.555417	27.343750	5.435294	28.687
1	35.555417	6.367041	5.435294	28.687
2	35.555417	7.142857	5.435294	28.687
3	35.555417	27.306273	5.435294	28.687
4	35.555417	5.639098	5.435294	28.687
...
3798	25.962425	8.396947	5.383530	29.562
3799	25.962425	6.766917	5.383530	29.562
3800	25.962425	6.000000	5.383530	29.562
3801	25.962425	8.045977	5.383530	29.562
3802	25.962425	13.229572	5.383530	29.562

[3733 rows x 4 columns]

```
[58]: training_size = int(len(df) * 0.8)

x_train = [[i] for i in df["TEMP"]][:training_size]
y_train = [i for i in df["CPU_USAGE"]][:training_size]

x_test = [[i] for i in df["TEMP"]][training_size:]
y_test = [[i] for i in df["CPU_USAGE"]][training_size:]

# Fitting Random Forest Regression to the dataset
```

```

# import the regressor
from sklearn.ensemble import RandomForestRegressor

# create regressor object
regressor = RandomForestRegressor(n_estimators=100, random_state=0)

# fit the regressor with x and y data
regressor.fit(x_train, y_train)

```

```
[58]: RandomForestRegressor(random_state=0)
```

```
[53]: Y_pred = regressor.predict(x_test)
```

```
[54]: def mean_absolute_percentage_error(y_true, y_pred):
        y_true, y_pred = np.array(y_true), np.array(y_pred)
        return np.mean(np.abs((y_true - y_pred) / y_true)) * 100
```

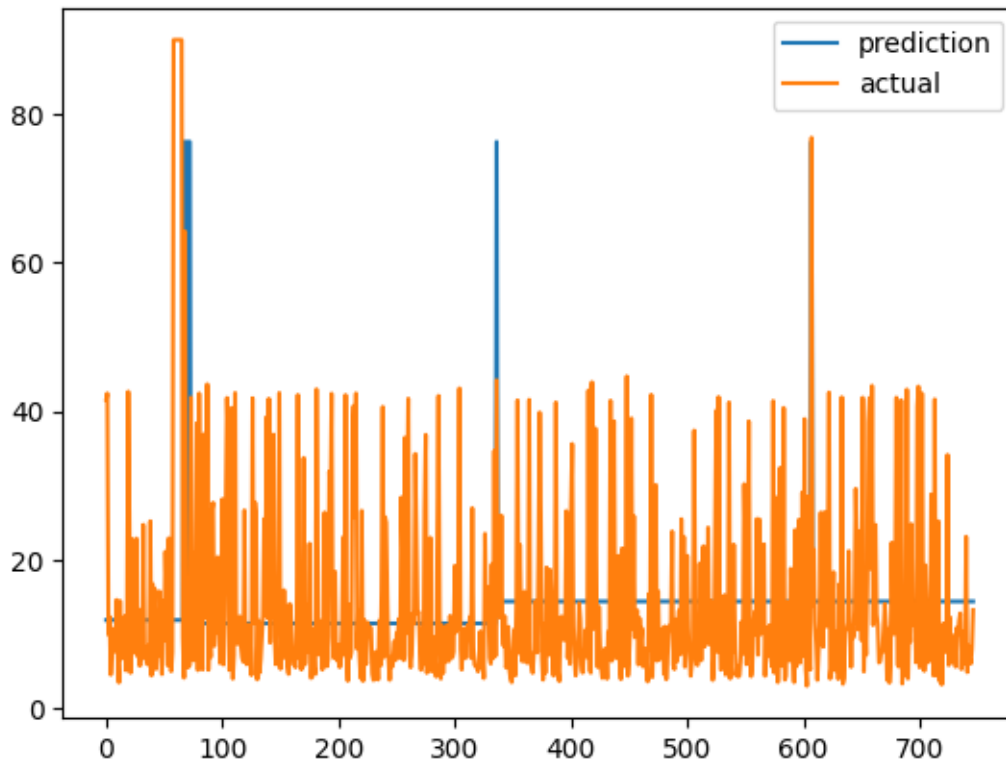
```
[55]: print(mean_absolute_percentage_error(list(Y_pred), y_test))
```

67.58065678765071

```
[56]: import matplotlib.pyplot as plt
import numpy as np

x = range(len(list(Y_pred)))
y_pred = list(Y_pred)
y_actual = y_test

plt.plot(x, y_pred, label="prediction")
plt.plot(x, y_actual, label="actual")
plt.legend()
plt.show()
```



```
[57]: training_size = int(len(df) * 0.8)

x_train = [[i for i in df["CPU_USAGE"]][:training_size]
y_train = [i for i in df["TEMP"]][:training_size]

x_test = [[i for i in df["CPU_USAGE"]][training_size:]
y_test = [[i for i in df["TEMP"]][training_size:]

# Fitting Random Forest Regression to the dataset
# import the regressor
from sklearn.ensemble import RandomForestRegressor

# create regressor object
regressor = RandomForestRegressor(n_estimators=100, random_state=0)

# fit the regressor with x and y data
regressor.fit(x_train, y_train)
```

```
[57]: RandomForestRegressor(random_state=0)
```

```
[35]: Y_pred = regressor.predict(x_test)
```

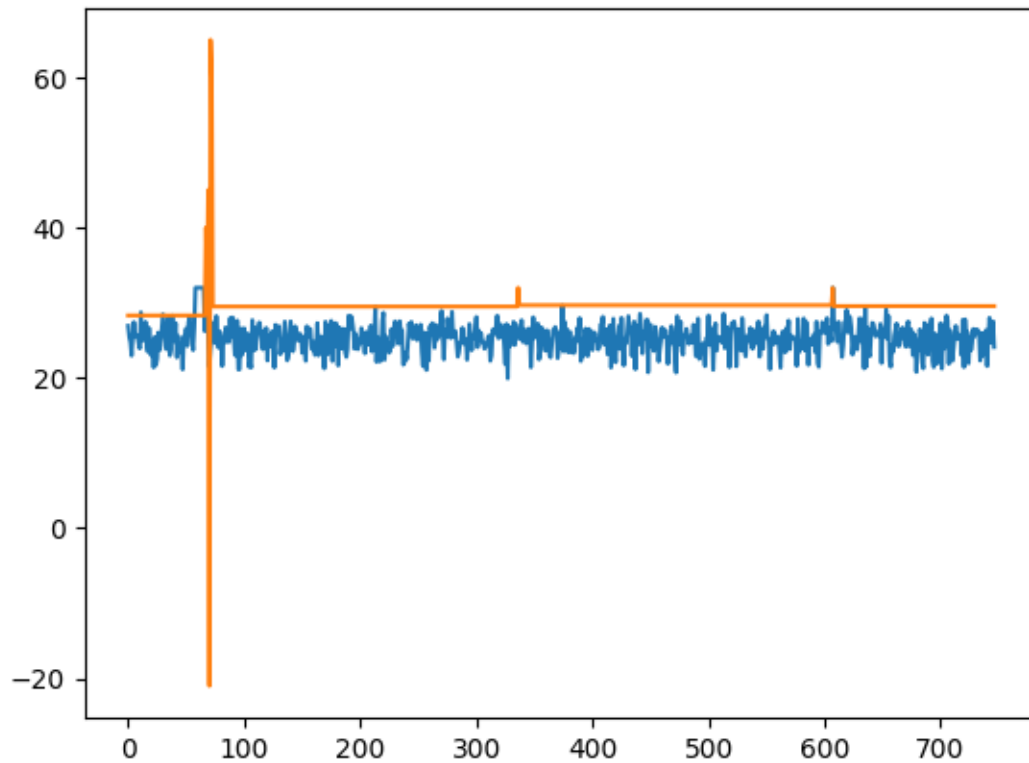
```
[37]: print(mean_absolute_percentage_error(list(Y_pred), y_test))
```

18.15146843031329

```
[44]: import matplotlib.pyplot as plt
import numpy as np

x = range(len(list(Y_pred)))
y_pred = list(Y_pred)
y_actual = y_test

plt.plot(x, y_pred)
plt.plot(x, y_actual)
plt.show()
```



```
[ ]:
```