Python - Machine Learning Continuous Assessment

Prepared by Team 4

11 November 2019

Name:	**Student ID:**
Seah Yong Wei, Gabriel	A0210292U
Khiangsu Myat Noe	A0210324B
Win Kyi Phyu Kyaw	A0210314A
Sriram Manda	A0210296L
Zhang Shu Yuan	A0210293R
Zeng Man Man	A0210294N

Contents

Problem Statement and Introduction

Being an island nation in the tropics and approximately 111kilometres north of the equator, Singapore is faced with higher levels humidity compared to countries in temperate regions for example, Finland. The team understands that temperature has a strong connection with humidity where humidity is influenced by rainfall and climate (sunshine etc).

In this project, analysis is done to predict the surface air temperature using monthly environmental data that has been collected over a span of 37 years (from January 1982 to September 2019). The environmental data used in this project was collected from data.gov.sg (Singapore's open data portal).

From the aforementioned data portal, 5 datasets which have a strong correlation with the mean monthly surface air temperature have been selected for analysis and prediction.

The datasets used are as follows:

1. Sunshine Duration – Monthly Mean Daily Duration
2. Relative Humidity – Monthly Mean
3. Rainfall – Monthly Total
4. Rainfall – Monthly Number of Rain Days
5. Surface Air Temperature – Monthly Absolute Extreme Maximum

The above 5 datasets have been considered as independent variables (x-variable) and Mean Surface Air Temperature as the dependent variable (y-variable).

The datasets above are collated into a single dataset. Each original dataset populates a single column (e.g. Sunshine Duration takes up one column etc). The new dataset would hence have a large variance in figures among the distinct columns as each column is a measurement of different factors. For example, the data series of Surface Air Temperature is measured in degrees Celsius ranging from 31.3^{o}C to 36^{o}C whereas the Monthly Total Rainfall is measured in millimetres with a range of 0.2mm to 765.9mm.

The original collated dataset also does not satisfy the Supervised model's requirement as it does not provide a default classification for the Surface Air Temperature.

Using Linear Regression on the original (unnormalized) collated dataset, it was not possible to achieve an accuracy score over 70%. Due to the inadequate accuracy, the team decided to normalize the original collated dataset and ran another round of Linear Regression but were still unable to have an accuracy score of over 70%, the accuracy result was 66.8%.

In this project, it has been found that the data or feature engineering did not improve the Linear Regression model as the accuracy score remained at an unsatisfactory level (i.e. below 70%). From this, it can be deduced that some correlation between the independent variables (multicollinearity) exists. Due to the limitations of Data Collection, addition or reduction of 1 or more independent variables (which is the general solution to multicollinearity) cannot be carried out, as this will result in a lower R^2 when using the Linear Regression model.

Hence, the following techniques have been employed to alleviate the above-mentioned problems:

1. Feature engineering to deal with the original dataset (explained in the Feature Engineering section)
2. Supervised method to achieve a higher accuracy.
3. Using unsupervised methods to perform clustering analysis

***More details of each model's output can be found in the Code File.

Data dictionary

Column Name	Description
month	Represents the period in which the data was recorded, type is “mm-yyyy”
surface air mean_temp	Mean value of monthly surface air temperature
mean_sunshine_hrs	Average hours of sunshine per month
mean_realtive humidity	Mean value of humidity per month
total_rainfall	Total rainfall in the particular month
no_of_rainy_days	Number rainy days per month
max_temperature	Highest recorded temperature in the month
temp_category	Temperature category. Whether the temperature is below or above (or equal to) 27.0^{o}C

The data dictionary shows the description for each distinct column found in the dataset.

As mentioned in the previous section, the environmental data used in this project was collected from data.gov.sg (Singapore’s open data portal).

Training Duration per Model

Model	Approximate Training Duration (in seconds)
Linear Regression	0.001995
Random Forest	0.118682
Decision Tree	0.001995
K-Nearest Neighbour	0.001994
Logistic Regression	0.031914
Principal Component Analysis	0.001994
DBSCAN	0.004986
K-Means	0.045877
Neural Network	22.333300

Table: Approximate time taken for to train each model

We have run the same models several times, each differing slightly in Training Duration. The above shows the results of a single run.

Based on the table above, it can be observed that the Logistic Regression, Linear Regression and Principal Component Analysis are tied as the models with the shortest Training Duration of 0.001994. The Random Forest model had a longer Training Duration as the machine was required to create multiple Decision Trees to populate the Random Forest. The model with the longest Training Duration is the Neural Network which took 22.3333 seconds to complete.

Feature Engineering

1. Data Normalization

In this project, Z-score normalisation was used in the dataset to ensure that every datapoint has the same scale so that each feature is equally significant.

2. Temperature Classification

The temperature of 27 ^{o}C is chosen as the mean recorded surface air temperature of the dataset is 27.65^{o}C. Therefore, the average monthly surface air temperature of 27 ℃ and above has been marked as 1 where temperature below 27^{o}C is marked as 0.

Findings from Machine Learning Modelling

This section explains our findings when running the Machine Learning Models. 4 models have been selected from the and will be explained below. The models are: Linear Regression, Principal Component Analysis, Density-based spatial clustering of applications with noise (DBSCAN), K-Means and Neural Network. Notes on the other models can be found in the code file.

Linear Regression

Findings and learnings:

1. The team used the Pearson Correlation Coefficient Matrix to discover which variables have strong correlation with the Surface Air Mean Temperature. In this project, we have selected variables with a coefficient value of over 60%.
2. Normalization process can impact the accuracy, but only when there is a huge difference between each independent variable. According to our dataset, the difference between two different columns of independent variables generally is not over 300. Consequently, in our case, the accuracy of the post normalization data did not differ greatly from the original data.
3. We have found that Linear Regression has limitations (i.e. accuracy) when using it on the dataset. Therefore, we have included other machine learning models to achieve higher accuracy scores.

Outcome analysis:

The final coefficient values are as follows:

mean_sunshine_hrs	mean_relative humidity	total_rainfall	no_of_rainy_days	max_temperature
0.12072960	-0.30950534	-0.12979502	0.18689082	0.36716467

The above table shows that humidity and total rainfall is inversely related to the surface air mean temperature. It can also be observed that both humidity and maximum temperature have the greatest influence on the mean temperature value.

The Intercept value is 27.659490790613486 which is the average surface air temperature. (See Temperature Classification in the Feature Engineering Section).

Principal Component Analysis (PCA)

The new principal component is a linear combination of the independent variables which means it contains the information from the independent variables. We can analyse the coefficients of these linear combination to find out the real meanings of the new principal component.

As 5 independent variables are used, the PCA can generate 5 principal components.

2 principal components for reducing the original dataset dimension have been chosen, the coefficients are as follows (correspond the below independent variables):

	mean_sunshine_hrs	mean_relative humidity	total_rainfall	no_of_rainy_days	max_temperature
PC1	-0.463171229	0.470711756	0.473879502	0.480041200	-0.330002303
PC2	-0.152626392	-0.0257990196	0.151816098	0.353670823	0.909894728

PC1 and PC2 contain 79% of the total information of the dataset
(Explained variance of 0.7900900197916095). More details on the Explained Variance can be seen in the Code file.

The first principal component (PC1) has negative coefficients for **mean sunshine hours** and **maximum temperature**, hence we did not include the information from these 2 variables. We have considered the 5 columns as the "Water effect".

The second principal component (PC2) has negative coefficients of the **mean sunshine hours** and **mean humidity.** We can consider this component as "Rain effect" component. The new value of this category can describe how raining affects the temperature.

Density-based spatial clustering of applications with noise (DBSCAN)

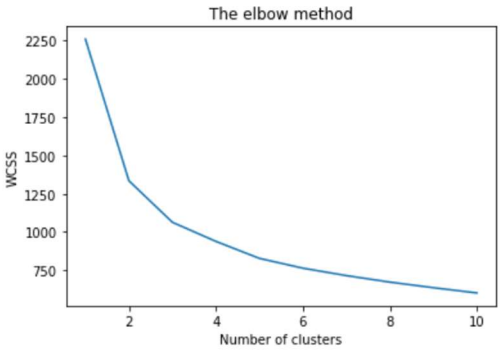
The original normalized dataset did not produce a satisfactory outcome when the DBSCAN model was used as the final cluster output contained too much noise. In our case, by setting an appropriate EPS value, the noise in the dataset was reduced.

When using the new data series, which is the 2 Principal Components generated from PCA, it showed that the noise sample count reduced from 447 to 20. This means that the clustering outcome is now more reliable.

Not only can PCA help us to analyse the independent variables, but can also assist in improving the other unsupervised models.

K-means clustering:

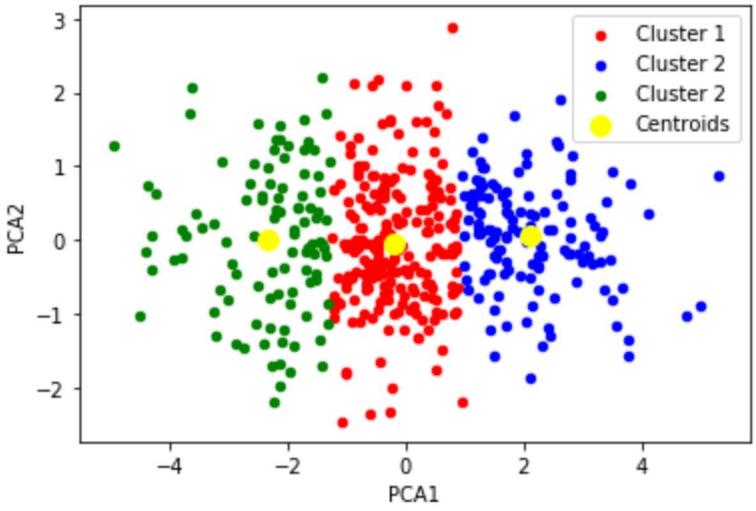
The “within cluster sum of squares(WCSS)” is shown in the diagram below:



It can be deduced that setting 3 clusters will be optimal. We have experimented that, after making 3 clusters, the distance of each value to the centroids will begin to converge. With increasing the number of clusters, the WCSS value changes are minimal (negligible) after 3 clusters.

The above-mentioned PCA helped us to reduce the 5 to 2 dimensions (5 independent variables to 2 Principal Components).

The K-means clustering diagram below illustrates 3 clusters that were generated using the 2 principal components from the above-mentioned PCA process.



Neural Networking

As the Linear Regression Model did not produce a satisfactory accuracy score the Neural Networking model is used to improve the accuracy score. By setting different activation functions, different solver (a method to minimise the loss function) and different number of layers, we were able to adjust the Neural Network model to achieve a higher accuracy score.

For example, in Model 2, with a hidden layer size of 10, the accuracy score was 69.73% as compared to Model 3 (hidden layer size of 20) which had an accuracy score of 67.82%. (see diagram below).

```
In [95]:
#Model 2
mlp = neural_network.MLPRegressor(hidden_layer_sizes=(10), activation="relu",
                  solver='lbfgs',
                 max_iter=10000)

a=mlp.fit(x1_train, y1_train)
y2_pred = a.predict(x1_test)

In [96]:
r2_score(y1_test, y2_pred)#Get the prediction accuracy

Out[96]:
0.6972568607208807

In [97]:
#Model 3
mlp = neural_network.MLPRegressor(hidden_layer_sizes=(20), activation="relu",
                  solver='lbfgs',
                 max_iter=10000)

a=mlp.fit(x1_train, y1_train)
y3_pred = a.predict(x1_test)

In [98]:
r2_score(y1_test, y3_pred)#Get the prediction accuracy

Out[98]:
0.6781737285169734
```

Conclusion and what we have learnt

Model	Manual Splitting Normalized Data	Random Splitting Normalized Data	Manual Splitting Unnormalized Data	Random Splitting Unnormalized Data
Linear Regression	NIL	66.80%	NIL	66.80%
Random Forest	88.46%	88.60%	88.46%	90.35%
Decision Tree	86.54%	90.35%	88.46%	92.11%
K-Nearest Neighbour	88.46%	89.47%	96.15%	82.46%
Logistic Regression	92.31%	92.98%	92.31%	93.86%
Neural Network (Classification)	NIL	84.00%	NIL	75.00%
Neural Network (Regression)	NIL	69.73%	NIL	NIL

Table 3: Model Accuracy Score

Intra-model comparison

We have found that using different methods of splitting data for training and testing can affect the accuracy score. In the project, 2 methods of splitting the dataset have been used. The first method was carried out by randomly splitting the dataset while the other method was done by manually splitting the dataset (done by selecting the first 400 rows of data as training data and the remaining rows as testing data).

It can be deduced that the amount of sample data chosen as training is proportional to the accuracy score. Hence, the higher amount of training data, the higher the accuracy of the model.

Model accuracy score comparison

A comparison on the above-mentioned Machine Learning Models which have been used during the project.

The following points explain the findings:

i. The Supervised model with the highest accuracy is the K-Nearest Neighbour model with an accuracy score of 96.15% whereas the Supervised model with the lowest accuracy score is the Linear Regression model with an accuracy score of 66.80%.
ii. Regardless of how we have optimized the dataset and how we split the dataset for training and testing, the linear regression model always reflected a low accuracy score (lower than 70%).
iii. Neural Network Linear Regression has a higher accuracy score than traditional Linear Regression Model. However, Neural Network Models cannot produce a higher accuracy score as compared to other Supervised Models. Furthermore, Neural Network Models require a longer duration than the other models.