Humidity is Difficult to Precisely Predicted and Classified

Polynomial works better than linear regression model

PCA works well with classification but might not with linear model

Data Science

Final Project Report

1st December 2019

Contents

1 Abstract

This report shows the process of handling the measurement data downloaded from the website *http://rp5.ru/*. The data are collected from September 2016 to May 2019 by the weather station 2978 located in Helsinki, Finland. The goal is to predict the relative humidity and classify the weather conditions based on those measurement data with high accuracy. In order to achieve that, it is important to identify the appropriate method using relevant features. The data analysis and different methods of approach are considered to conclude the best option for both prediction and classification. The final results and further discussion are also mentioned. This report is important as it is the final project for the Data Science course at Aalto University. This project is such a great opportunity to fully implement what we have learned throughout the course. The tasks are interesting and challenging for us.

2 Introduction

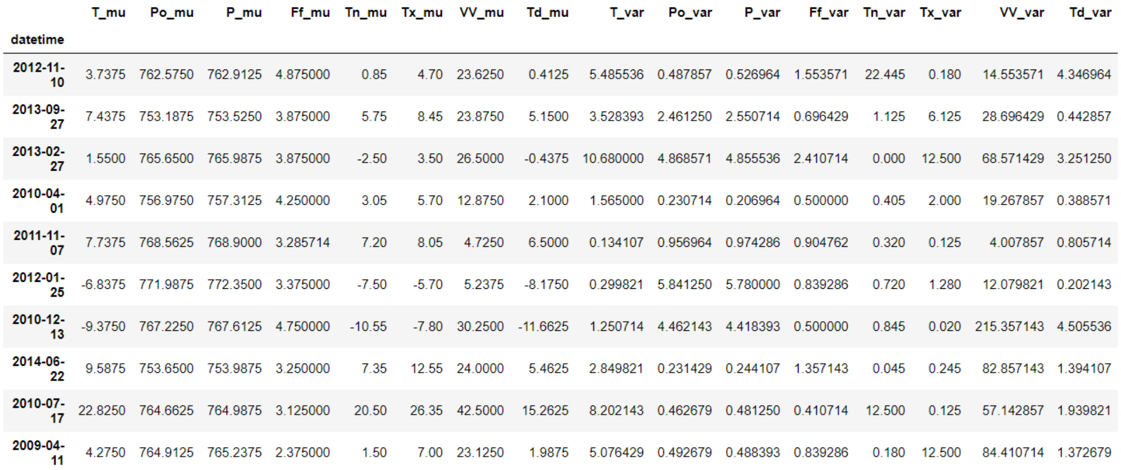
As the weather forecast is the key factor probably in saving millions of lives in many terrible cases, the prediction should be both correct and fast for our preparations. Computers ensure both conditions and machine learning is applied for higher accuracy and better results. That is also the project's tasks, using machine learning to predict and classify. After this, we hope to learn useful tools and the approach to come up with a good methods for weather prediction.

The measurement data is downloaded from the "Reliable Prognosis" website supported by Raspisaniye Pogodi Ltd., St. Petersburg, Russia. The website itself provides weather forecasts from 172.500 locations as well as weather observation reports from more than 16.000 stations. Those measurement data is then resampled, simplified and divided into 4 smaller .csv files. 2 files for both training and testing data, another 2 files for the labels. The training and testing .csv include 16 attributes such as air temperature, atmospheric pressure, wind speed, etc. As mentioned, the task is to predict the relative humidity and classify the weather condition either as "dry" or "not dry". Firstly, the data is needed to be analyzed, filtered and then used to build the model. Secondly, the model uses the testing data and returns the results. Lastly, the results are compared with the labels and accuracies are calculated.

3 The Process

3.1 Data Exploration

The training data is a 2-dimension data frame whose shape is (3140, 16). *Figure 1* shows 10 samples drawing from the training data. The data is checked to ensure there are no missing values.



datetime	T_mu	Po_mu	P_mu	Ff_mu	Tn_mu	Tx_mu	VV_mu	Td_mu	T_var	Po_var	P_var	Ff_var	Tn_var	Tx_var	VV_var	Td_var
2012-11-10	3.7375	762.5750	762.9125	4.875000	0.85	4.70	23.6250	0.4125	5.485536	0.487857	0.526964	1.553571	22.445	0.180	14.553571	4.346964
2013-09-27	7.4375	753.1875	753.5250	3.875000	5.75	8.45	23.8750	5.1500	3.528393	2.461250	2.550714	0.696429	1.125	6.125	28.696429	0.442857
2013-02-27	1.5500	765.6500	765.9875	3.875000	-2.50	3.50	26.5000	-0.4375	10.680000	4.868571	4.855536	2.410714	0.000	12.500	68.571429	3.251250
2010-04-01	4.9750	756.9750	757.3125	4.250000	3.05	5.70	12.8750	2.1000	1.565000	0.230714	0.206964	0.500000	0.405	2.000	19.267857	0.388571
2011-11-07	7.7375	768.5625	768.9000	3.285714	7.20	8.05	4.7250	6.5000	0.134107	0.956964	0.974286	0.904762	0.320	0.125	4.007857	0.805714
2012-01-25	-6.8375	771.9875	772.3500	3.375000	-7.50	-5.70	5.2375	-8.1750	0.299821	5.841250	5.780000	0.839286	0.720	1.280	12.079821	0.202143
2010-12-13	-9.3750	767.2250	767.6125	4.750000	-10.55	-7.80	30.2500	-11.6625	1.250714	4.462143	4.418393	0.500000	0.845	0.020	215.357143	4.505536
2014-06-22	9.5875	753.6500	753.9875	3.250000	7.35	12.55	24.0000	5.4625	2.849821	0.231429	0.244107	1.357143	0.045	0.245	82.857143	1.394107
2010-07-17	22.8250	764.6625	764.9875	3.125000	20.50	26.35	42.5000	15.2625	8.202143	0.462679	0.481250	0.410714	12.500	0.125	57.142857	1.939821
2009-04-11	4.2750	764.9125	765.2375	2.375000	1.50	7.00	23.1250	1.9875	5.076429	0.492679	0.488393	0.839286	0.180	12.500	84.410714	1.372679

Figure 1. Training data samples.

It can be seen that there are totally 16 attributes. The prefixes of attributes are *_mu* and *_var* indicates the mean and the variance values for each day respectively. Numerical attributes meaning is shown in the following table.

Name	Meaning
T	The air temperature, in degrees Celsius, 2 meters above the earth's surface.
Po	The atmospheric pressure at weather station level, in millimeters of mercury.
P	The atmospheric pressure reduced to mean sea level, in millimeters of mercury.
Ff	The mean wind speed at a height of 10-12 meters above the earth's surface, in meters per second.

Tn	The minimum air temperature, in degrees Celsius, over the past day.
Tx	The maximum air temperature, in degrees Celsius, over the past day.
VV	The horizontal visibility, in km.
Td	The dewpoint temperature at a height of 2 meters above the earth’s surface, in degrees Celsius.

Table 1. Definitions of numerical attributes.

We now start to explore the data and complete the required tasks. *Figure 2* below is the histogram plot of the mean of both *Tn_mu* and *Tx_mu* or the minimum and maximum air temperature in degrees Celsius. The outcome is not as we expected: The mean of the minimum air temperature turns out higher than the maximum one.

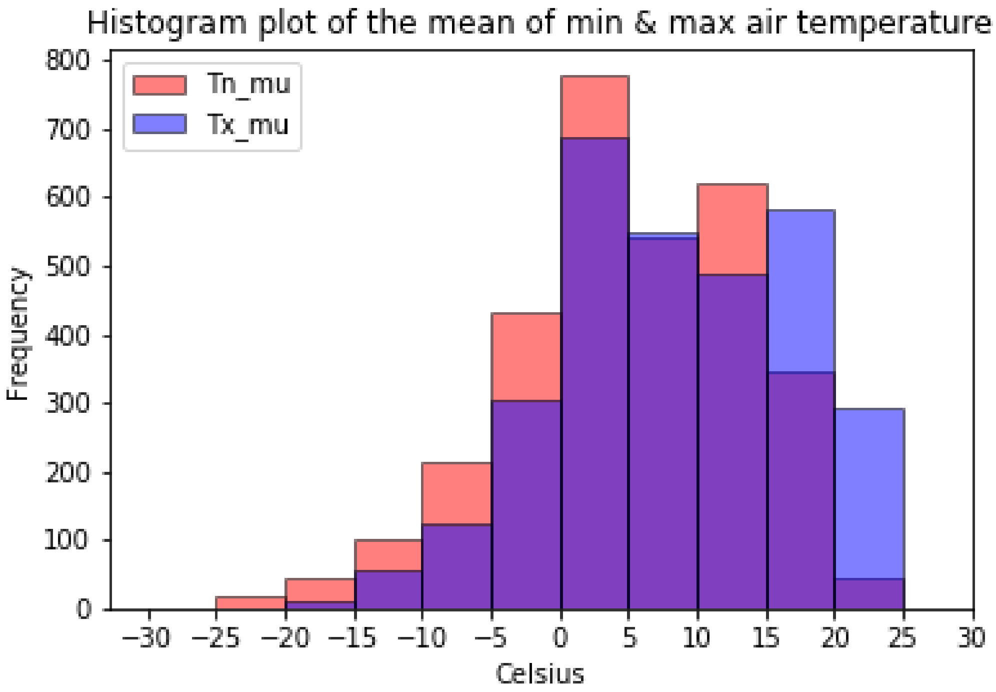


Figure 2. Histogram plot of *Tn_mu* and *Tx_mu*.

Figure 3 shows pair plots of 6 attributes, the last one *U_mu* is the mean of the relative humidity from the labels data. Pair plots are used to pair features and plot them together as scatter plots. The diagonal shows the distribution of a single variable.

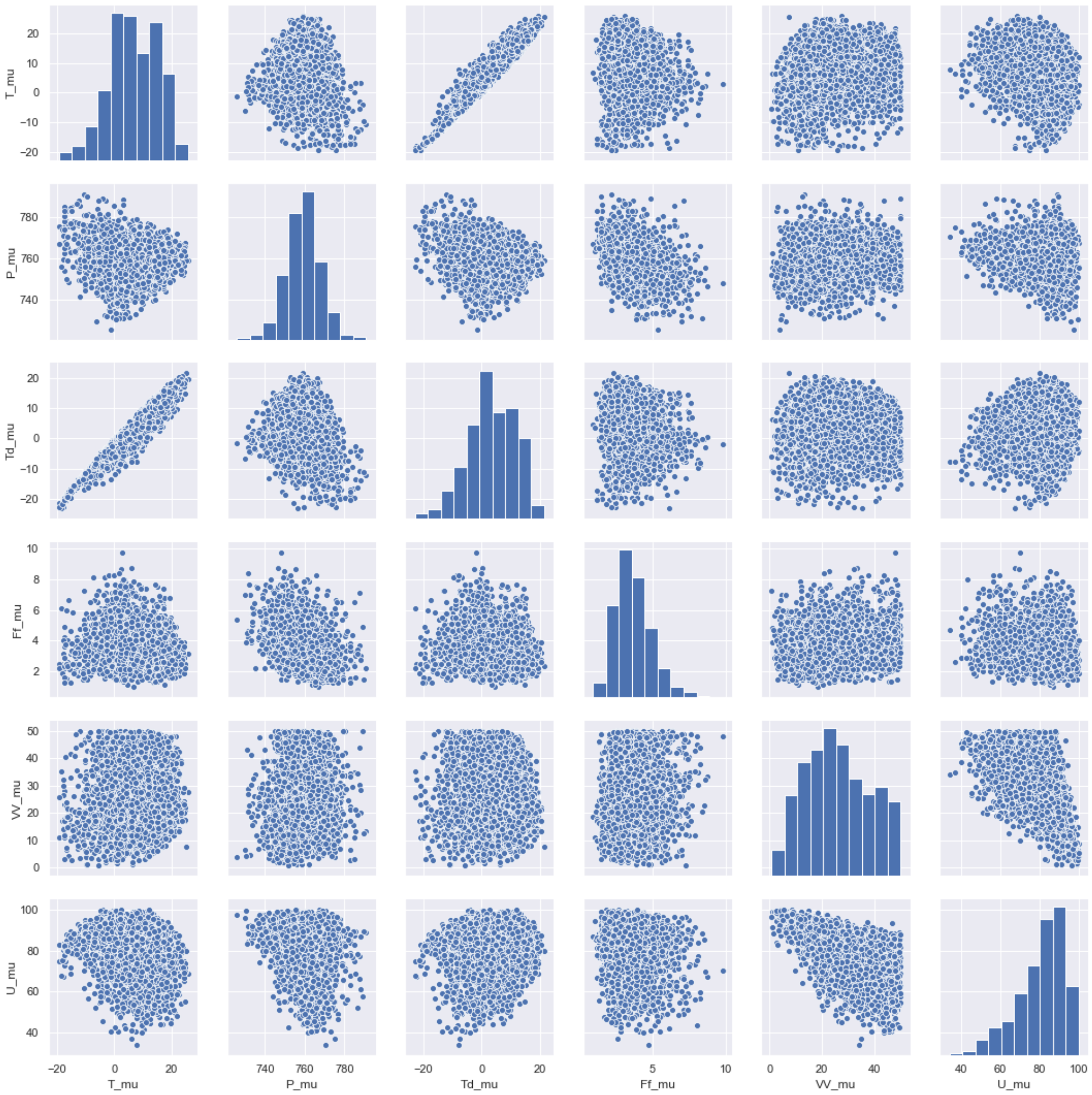


Figure 3. Pair plots for *T_mu, P_mu, Td_mu, Ff_mu, VV_mu* and *U_mu*.

As the correlation matrix of features is required, we add *OBSERVED* feature from labels data and display the whole correlation matrix in *Figure 4*. The correlation matrix tells the relationship between variables. Each variable has the highest correlation with itself; therefore, the diagonal of the matrix has all 1.0 correlation. The higher the positive correlation, the more likely one is in direct proportion with another; while the lower negative, the correlation indicates the opposite. Correlation value 0 shows that 2 variables do not relate. From *Figure 4*, *U_mu* does not have any strong positive related to any variable, but it has a quite high negative correlation with *VV_mu*. Also, *OBSERVED* is likely to relate to *Po_mu* and *P_mu* because they have higher positive correlations compared with others.

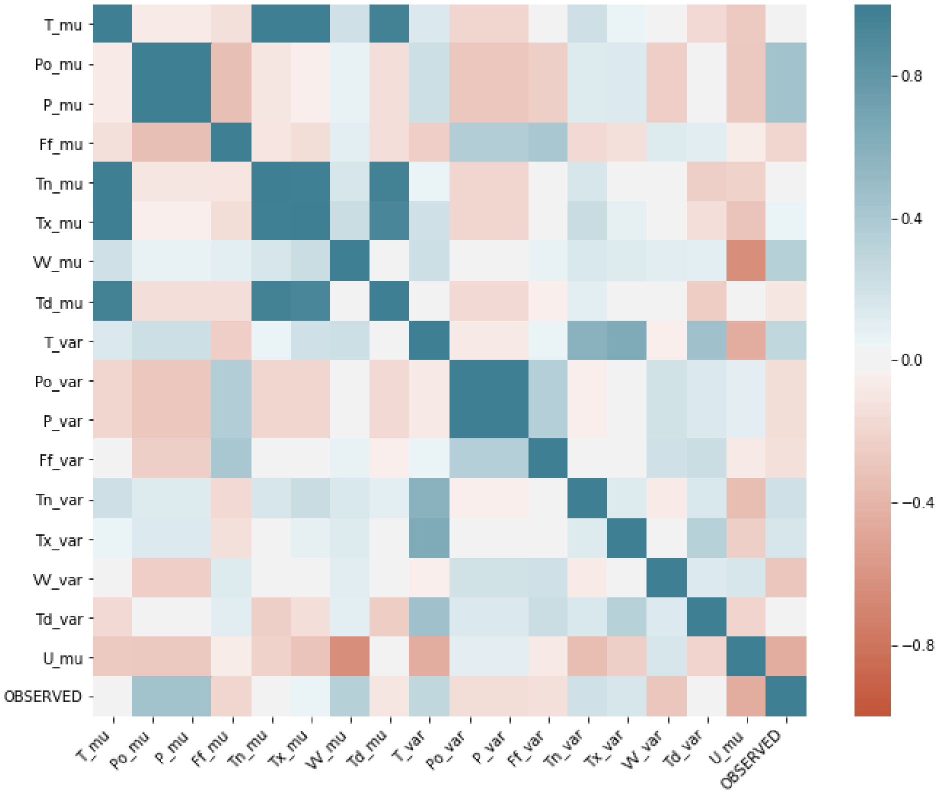


Figure 4. The correlation matrix of the features.

After that, we standardize data using *StandardScaler* and implement PCA. The number of components is selected by observing the variance explained by components

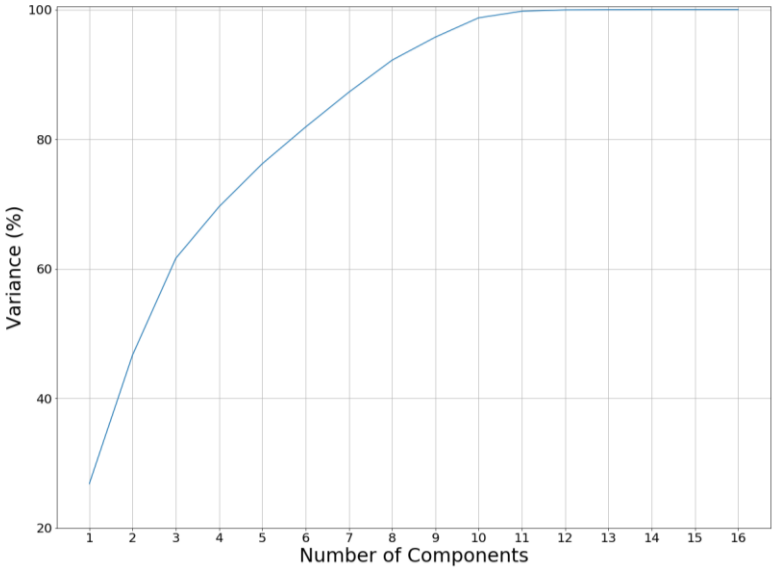


Figure 5. The cumulative explained variance plot.

Figure 5 shows that selecting 10 components we can preserve around 99% of the total variance of the data, only 1% is lost, but the numbers of dimensions reduce from 16 to 10. It makes sense, we'll not use 100% of our variance, because it denotes all components, and we want only the principal ones.

3.2 Methods

In the process of predicting the relative humidity, the first step is to normalize the test set and train set. Afterwards, we train data using linear regression model, fit it with the normalize train set and the train-set-labels *U_mu*. The coefficient table shows the high absolute value for *T_mu* and *Td_mu*, meaning they are likely to be strong predictors. After fitting, the model then used to predict the test data. The figure below plots the line of best fit, actual values and the predicted one.

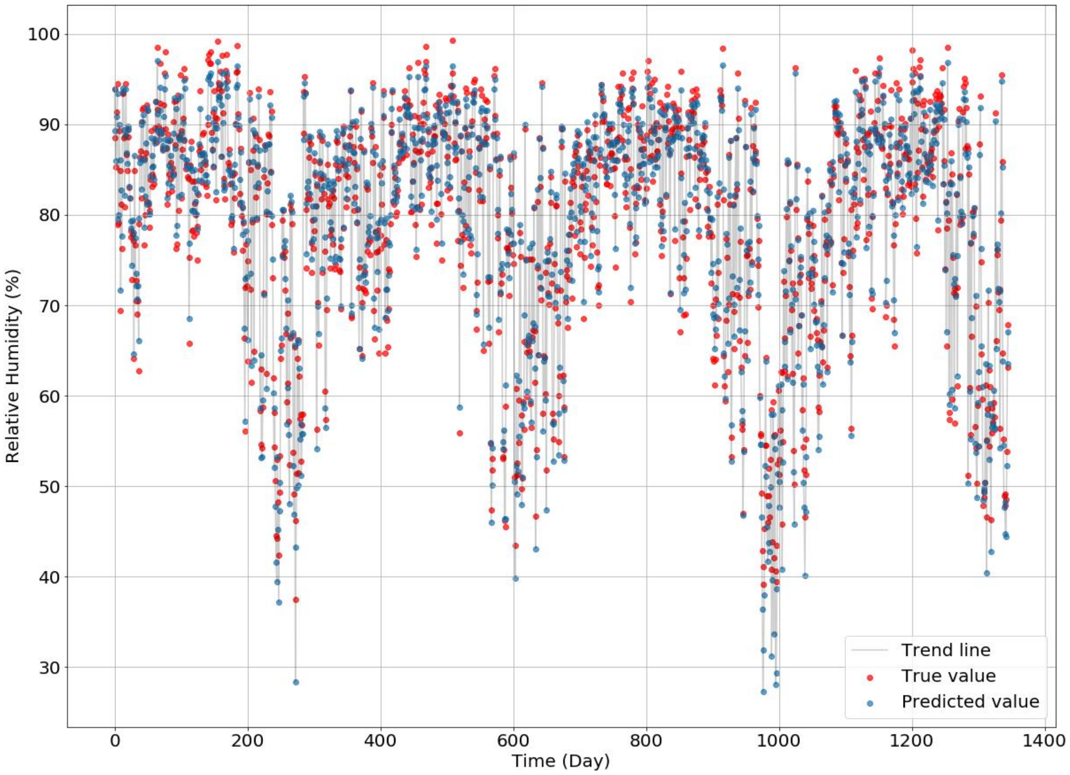


Figure 6. Plot of predicted values and true values of the linear model (without PCA).

Later on, we adopt PCA to reduce the dimension from 16 to 10. As mentioned above, 10 principal components hold almost 99% of the variance so we choose this number. After transforming both train and test data using PCA, the linear process is repeated and the mean square error is calculated.

As we plotted the true *U_mu* labels in *Figure 6,* we believe the polynomial model or quadratic model would result in a smaller mean square error compared to the linear model because of the complicated *U_mu* pattern. Without adopting PCA, using *PolynomialFeatures* from *sklearn* to transform and fit the train and test data, we run the prediction again to validate our assumption.

Moving onto the classification step, where we attempt to classify the weather conditions either as “dry” or “not dry”, based on the available measurements. Firstly, we try KNN-classification, with an optimal number of neighbors (K = 15) accomplished by trying *KNeighborsClassifier* with value in range from 1 to 50 and getting the minimum corresponding error (see *Figure 7).* We also try to apply logistic regression to compare the result.

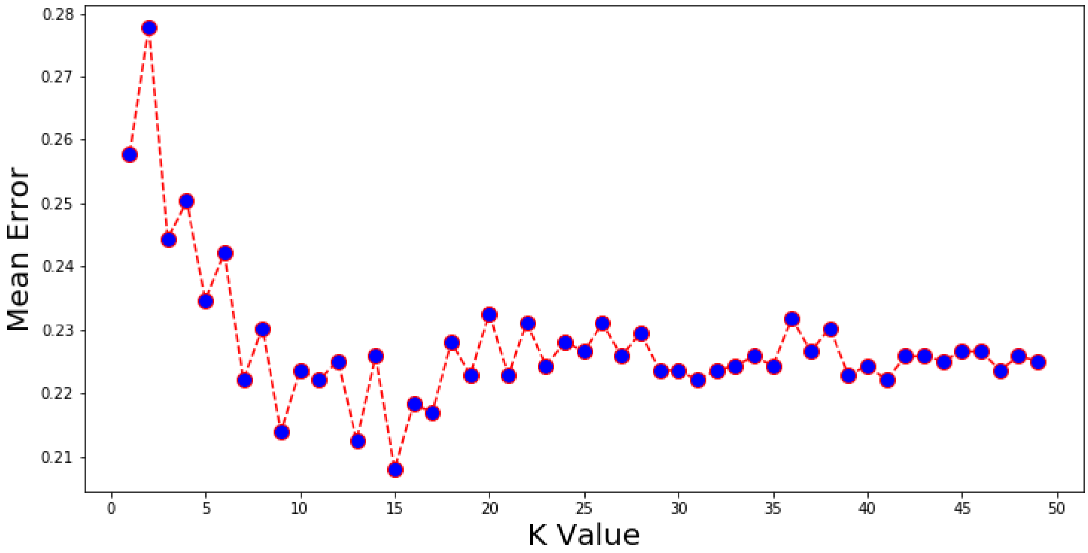


Figure 7. Plot of number of neighbors with corresponding mean error.

Since there are plenty of important predictors, we adopt PCA (10 components) to the train dataset and the test dataset for reducing the dimensionality, use them to classify the weather condition with KNN-classification and logistic regression again. The optimal number of neighbors found by the reduced data is different (K = 39), but its mean error is roughly same as that of the previous number of neighbors (see *Figure 8*).

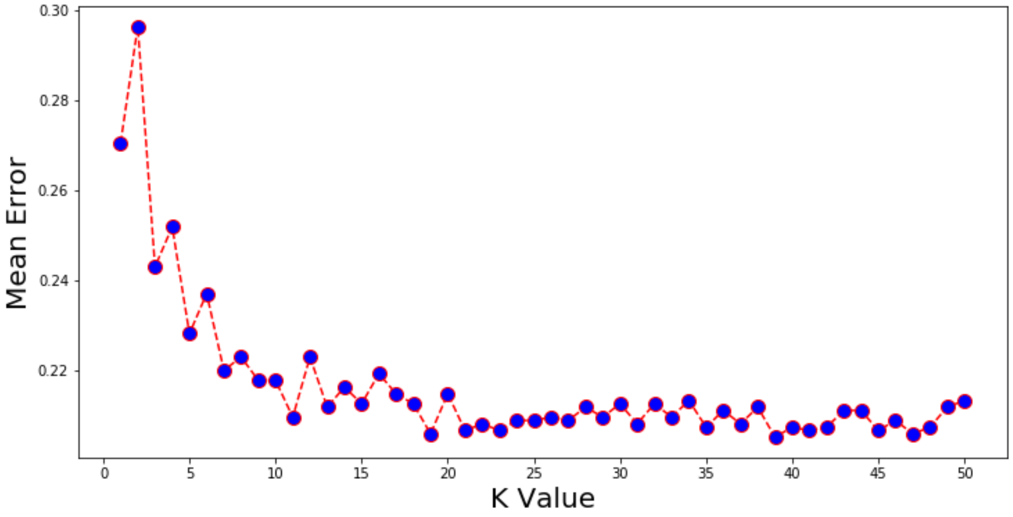


Figure 8. Plot of number of neighbors with corresponding mean error (after PCA).

3.3 Testing and Results

For the relative humidity prediction, the final result is shown in table 2. Checking MSE of the prediction without PCA on the train test, we guess there might be a negligible overfit because their MSE difference is not significant. Also, PCA adoption results in significantly higher MSE and as expected, the MSE in the polynomial model is much smaller than the linear one.

Methods	MSE of the test data	MSE of the train data
Linear model without PCA	≈ 2.6114	≈ 1.9713
Linear model with 10-component PCA	≈ 66.7218	≈ 51.8744
Polynomial model without PCA	≈ 0.281	≈ 0.3

Table 2. The MSE of both test and train data for linear model with and without PCA.

For the weather condition classification, the final results are plotted as confusion matrixes in *Figure 9*.

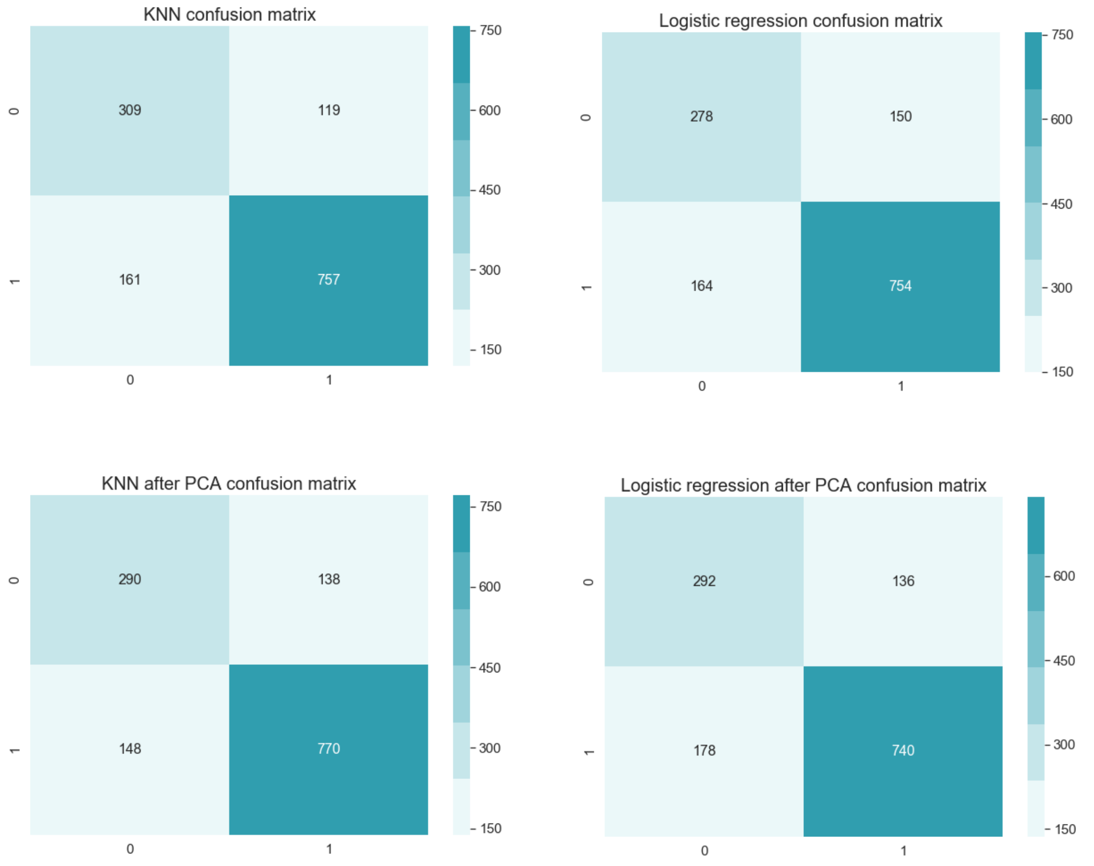


Figure 9. Confusion matrices.

Firstly, the confusion matrices and accuracy of KNN-classification and logistic regression indicates that both methods result in quite similar numbers. The accuracy of KNN-classification is higher by 2% than that of logistic regression; however, 79% and 77% are just acceptable levels of accuracy.

Secondly, after feature selection (PCA), the only change is the number of neighbors using in KNN-classification which seems not to have a big influence on the confusion matrices and accuracy levels.

4 Conclusion and Discussion

For the relative humidity prediction, our prediction with 10-component PCA (MSE ≈ 66.7) does not result in the smaller mean square error compared with prediction without PCA

(MSE ≈ 2.6). We run *i*-component PCA with *i* in the range from 1 to 16; the result is printed in the figure below. We found out that from 11-component to 12-component PCA, the mean square error significantly dropped; starting from 14 components, the value does not change much. We assume that the strong predictor is the 12th principal component. Therefore, the reason why the mean square errors were significant high is that the strong predictor was not projected on the first 11th component as our teaching assistant Letizia mentioned earlier.

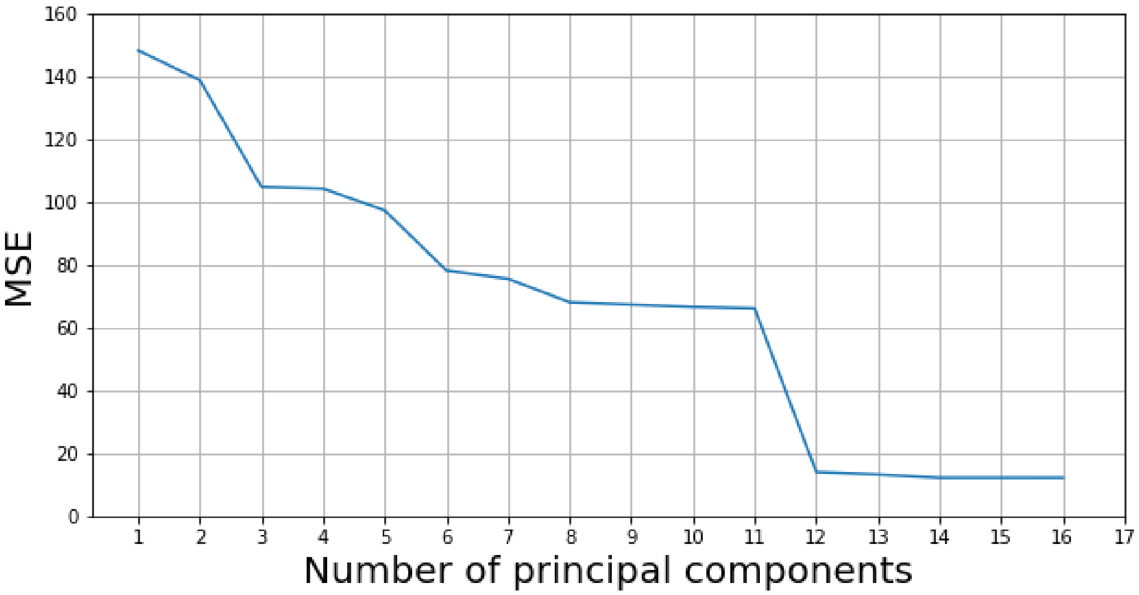


Figure 10. The mean squared error corresponds to number of principal components plot.

Besides, we did find out that *Td_mu* is the strongest predictor through trial and error by manually remove a column from the original data and run the linear regression without PCA adoption. According to the correlation matrix from *Figure 4*, *Td_mu* does not have a strong related to *U_mu* and there are some features we thought could be strong predictors because of their higher absolute correlations but they were not when we tested it. The discovery surprised us and eventually, we assume that the correlation matrix truly shows the linear relationship between variables and cannot foretell the strong predictors. Also, we did check the train labels data 0 and 1 size and found that they are slightly unbalanced. However, the results did not change much when we tried to balance them so the process is not included in the report. In conclusion, the polynomial model without PCA is the best method we got for the relative humidity prediction.

		Logistic regression	KNN
Fit with test dataset	Without PCA	77	79
	With 10-component PCA	77	79
Fit with train data (overfit testing)	Without PCA	79	81
	With 10-component PCA	79	81

Table 3. Accuracy of 2 classifiers (%).

Talking about “dry” or “not dry” classification, KNN-classification appears to be an acceptable method with roughly 80% accuracy. As we also compare it with logistic regression and achieve almost the same result, we start thinking to conclude that with available basic classifier the weather condition can just be classified in an acceptable accuracy level. Feature selection (PCA) is not likely to be supportive in this task: Before and after PCA, the results don’t have a considerable change.

We also test those classification methods with train dataset to make sure that the classifiers are not overfit and acquire a negligible accuracy difference (2%).

References

1 Arthur Gonsales.An Approach to Choosing the Number of Components in a Principal Component Analysis; 2018.
URL: https://towardsdatascience.com/an-approach-to-choosing-the-number-of-components-in-a-principal-component-analysis-pca-3b9f3d6e73fe
Accessed 22nd November 2019.

2 Onel Harrison.Machine Learning Basics with the K-Nearest Neighbors Algorithm; 2018.
URL: https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761
Accessed 23rd November 2019.

3 Michael Galarnyk. PCA using Python (scikit-learn). 2017.
URL: https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60
Accessed 27th November 2019.

4 LD Free Man. A Data Science Framework: To Achieve 99% Accuracy. 2017.
URL: https://www.kaggle.com/ldfreeman3/a-data-science-framework-to-achieve-99-accuracy?fbclid=IwAR2gWXSFXIdBArMVIH4skRD-UKPpmcJW-nPOt_vancgvkpYB1-0HopVrtSQ
Accessed 15th November 2019.

5 Serigne. Stacked Regressions: Top 4% on LeaderBoard. 2017.
URL: https://www.kaggle.com/serigne/stacked-regressions-top-4-on-leaderboard?fbclid=IwAR0emZqXuwpxCIOKqpnv2vyrCpf8aBe-xyKtQ2ZdHShKd_l8GOKm4WkPyEk
Accessed 20th November 2019.