Weather forecasting with data science

CS-C3160 Data Science

Aalto-yliopisto

Abstract

In this Data Science project, we will analyse the features of the dataset with mathematical models and how accurately a machine learning classifier can predict the humidity of the day.

For feature selection, we used pandas Python Pandas library correlation and PCA functions. All regression and classifier models are implemented with Python sklearn library.

It was proven that it is possible to build a decent KNeighborsClassifier model with an accuracy of 0.797 and f1-score of 0.772. Furthermore, we compared results with different models and stratifying the dataset affects the result.

Introduction

For a long time, people have been forecasting weather with different methods. A seven-day forecast can accurately predict the weather about 80 percent of the time with traditional mathematical models[1].

For the past decade, we have been using machine learning with predicting stock prices, real estate values and even for cancer detection. This made us wonder: Is it possible to make more accurate predictions with a machine learning model?

We hope to learn how accurately different machine learning models perform on the weather dataset. In addition, we will find out the most important features to predict the weather and how they behave with mathematical models. The data are measurements collected by the weather station 2978 in Helsinki from September 2006 to May 2019. Our motivation is to develop a precise model and gain new insights from the dataset.

Data analysis

The weather dataset is divided into four CSV-files: train_data (features), test_data (features), train_labels (labels) and test_labels (labels). The training set contains 3140 observations whereas the test set contains 1346 observations.

There are 16 different features eg. the atmospheric pressure, the air temperature above the earth's surface. Also, there are two different labels which we use as the target for predictions: the mean relative humidity 2 meters above the earth's surface (U_mu) and dryness of the day (binary classl: yes or no).

Total	Training set	Test set
4486	3140	1346

Figure 1, The dataset. 70% training and 30% test.

	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
0	14.4875	751.3000	751.6375	3.500	13.30	15.95	11.425	12.550	0.926964	1.008571	0.979821	1.142857	0.320	4.205	155.590714	1.994286
1	14.1875	758.0625	758.3625	3.625	11.20	15.95	27.500	11.025	4.801250	7.965536	7.679821	0.267857	5.780	6.125	147.142857	1.942143
2	15.3000	762.1125	762.4375	3.000	13.15	16.70	12.875	12.875	3.754286	1.824107	1.742679	0.857143	1.445	10.580	23.553571	0.122143
3	14.0250	766.4000	766.7625	2.500	12.00	16.65	7.200	12.500	4.896429	0.417143	0.431250	0.285714	0.180	10.125	35.974286	0.968571
4	14.2750	764.7125	765.0500	3.250	12.55	15.80	10.625	12.475	3.659286	1.672679	1.680000	0.785714	3.645	3.920	16.267857	1.005000

"T" is the air temperature, in degrees Celsius, 2 meters above the earth's. surface.
"Po" is the atmospheric pressure at weather station level, in millimeters of mercury.
"P" is the atmospheric pressure reduced to mean sea level, in millimeters of mercury.
"Ff" is the mean wind speed at a height of 10-12 meters above the earth's surface, in meters per second.
"Tn" is the minimum air temperature, in degrees Celsius, over the past day.
"Tx" is the maximum air temperature, in degrees Celsius, over the past day.
"W" is the horizontal visibility, in km.
"Td" is the dewpoint temperature at a height of 2 meters above the earth's surface, in degrees Celsius.

Figure 2, Features of the dataset. var = variance, mu = mean. There are 16 different features.

Wet days	Dry days
1666	2820

Figure 3, The dataset labels.

As we can see in figure 3, class distribution in the dataset is unbalanced. There are more data for dry days compared to wet days. We have to use balanced training- and test set while developing the classifier. If we don't, the model will be biased towards dry days and have lower accuracy with wet days.

There are few interesting insights in the dataset. For example, there are a lot more days with maximum temperature is 20-25 degrees than minimum. In Figure 4 (Tn_mu / Tx_mu), we can see a clear trend in the middle: temperatures are gathered around right side of 0 degrees. Helsinki is a bit warmer than many can expect! We can see same kind of trend in Figure 4 (P_mu, Po_mu): Warm temperature causes atmospheric pressure to rise, and vice versa[2].

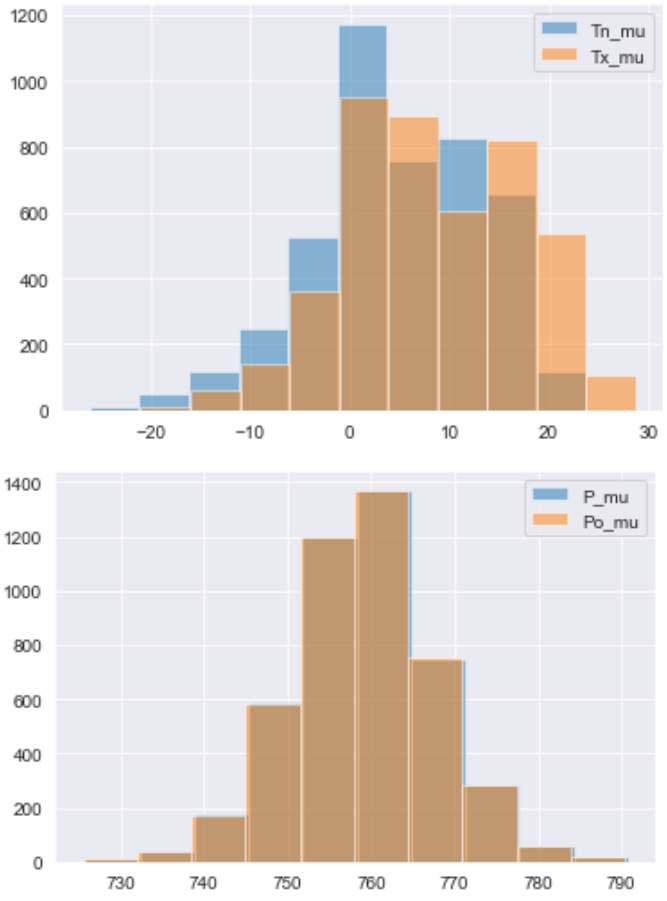

Figure 4, Histograms of Tn_mu/Tx_mu and P_mu/Po_mu.

Since there might be relations between other features, we made pair plots for T mu, P mu, Td mu, Ff mu, VV mu and U mu (Figure 5) and a correlation matrix with all features and labels (Figure 8).

Naturally, there is a positive correlation between Td_mu (mean dewpoint temperature) and T_mu (mean air temperature) since dew temperature is the point where air must be cooled to become saturated with water vapor. Also, there is a negative correlation between W_mu (horizontal visibility) and U_mu (mean relative humidity) since mists and fogs reduce horizontal visibility [3].

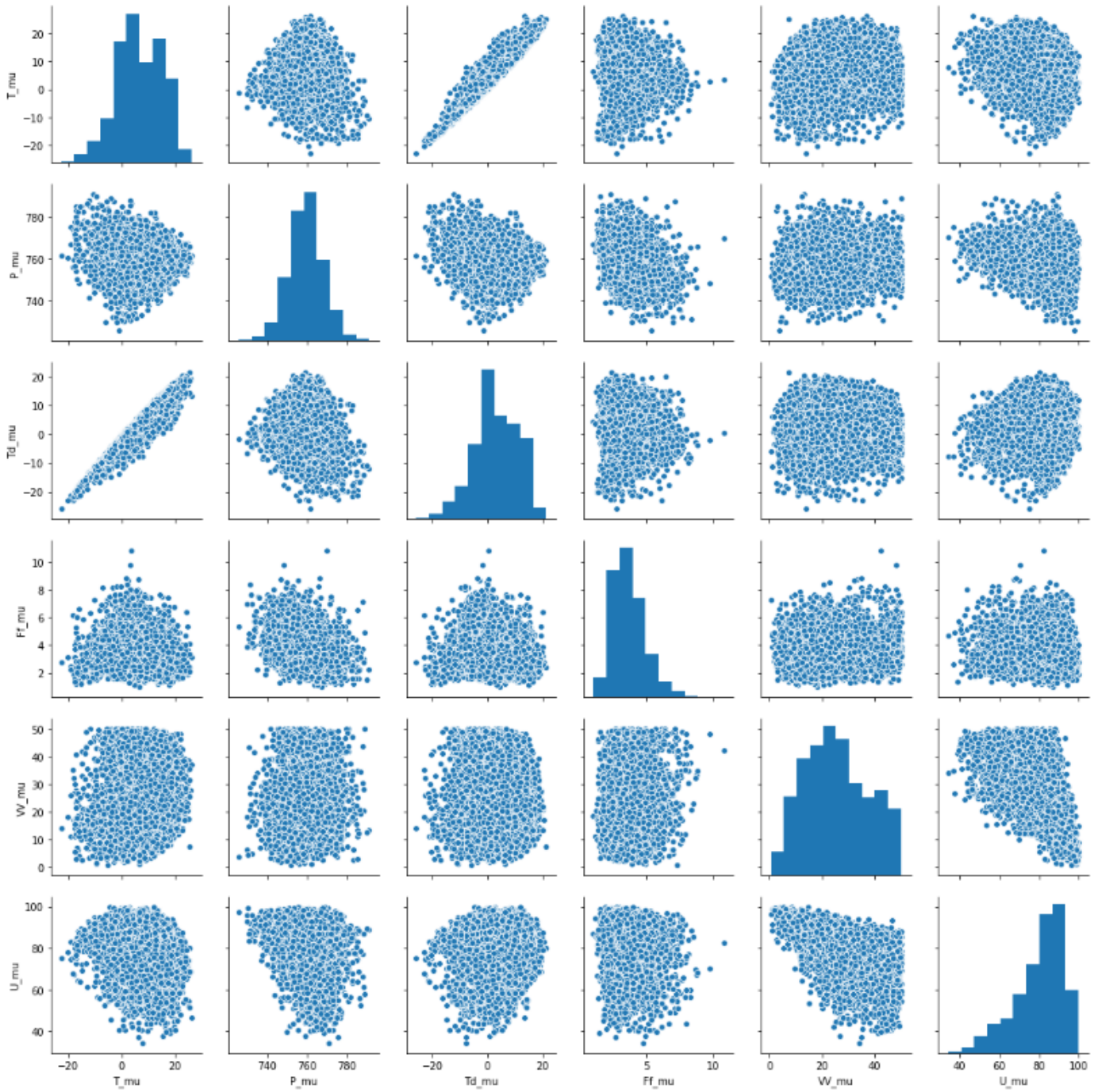

Figure 5, Pair plots of T_mu, P_mu, Td_mu, Ff_mu, VV_mu, U_mu.

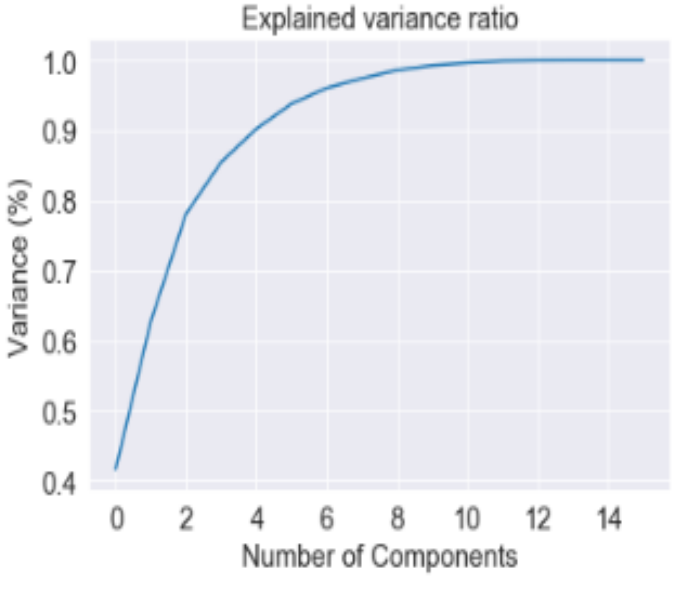

Figure 6, PCA components and variance %. 0th in axis is the 1st component.

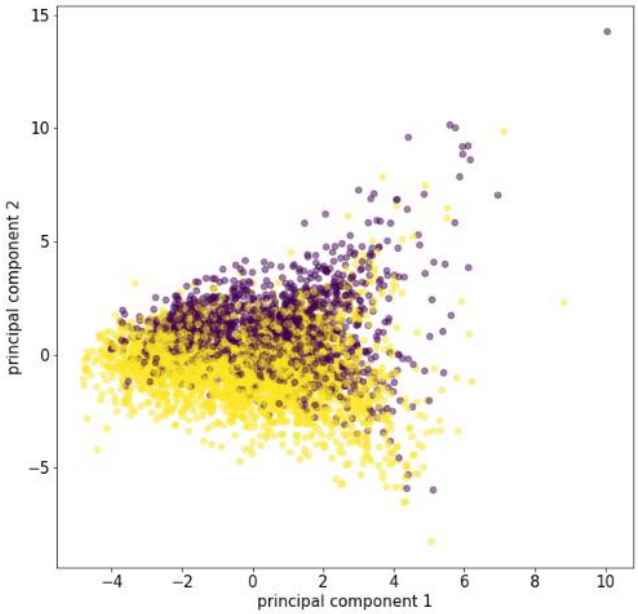

Figure 7, PCA projection with two components. Yellow = Dry, Purple = Wet

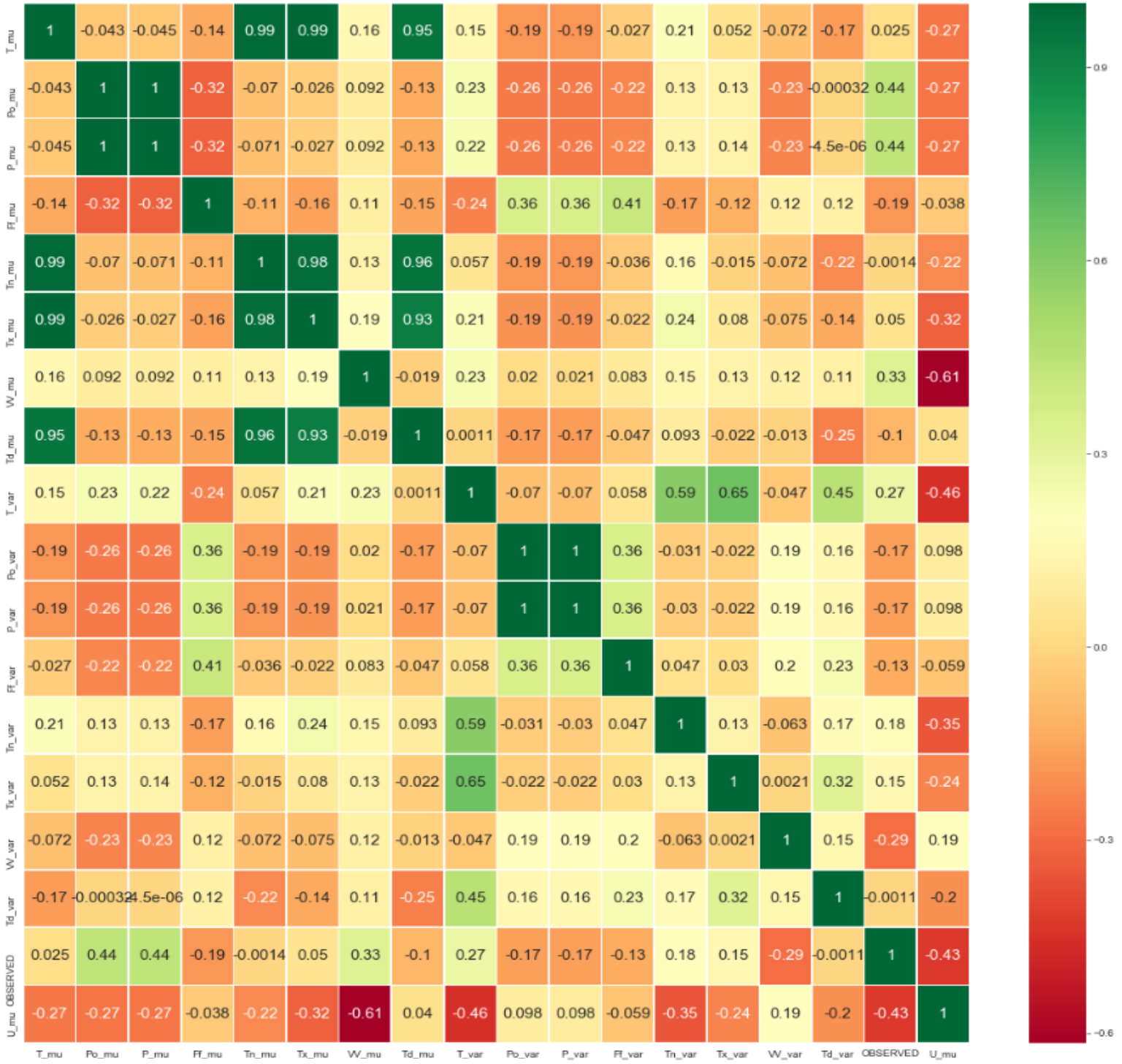

Figure 8, Correlation matrix of features and labels

We implemented PCA to reduce the dimension of the dataset and select the right number of components by studying explained variances. In figure 6, after the 12th component (the 11th in the figure), the change is insignificant. Therefore the first twelve components have the great portion (99.5%) of information about the dataset features.

In Figure 7, There are the first and second components of PCA. There is a clear relationship between the two components. Also, we colored points by OBSERVED label value.

Methods

Our process is presented step by step in Figure 9.

First, we can use stratifying to balance training and testing sets by classes. This method guarantees that the model doesn't overfit only for one class. The goal is that sets has the same amount of data for wet days and dry days.

After that, we standardized features since the range of values in raw data varies widely. Temperature values differ a lot from air pressure which uses millimetre mercury as a unit. For this reason, we used standardization for feature scaling.

Since there are a lot of dimensions in the dataset, we tested to perform experiences with and without PCA. We used twelve components, which contain 99.5% of information (Figure 6).

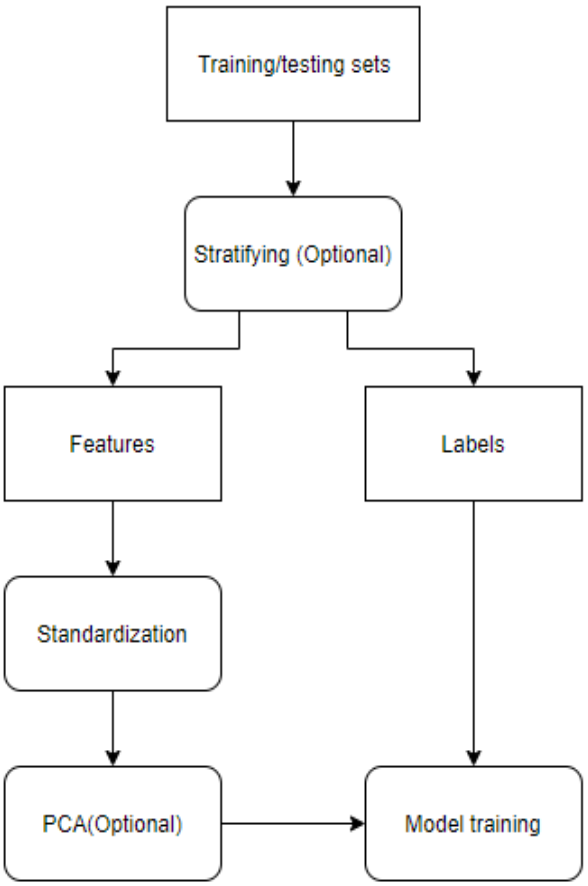

Figure 9, Whole process with regressions and classifiers.

For regressions, we selected linear regression and lasso regression. Since features follow a normal distribution (Figure 5) and aren't extremely correlated with each other, we use linear models. Main differences between linear- and lasso regression is how they adjust weights. Linear regression doesn't penalize for its choice weights whereas lasso would. By using lasso regression, we are preventing overfitting the model on the dataset. Alpha and max iteration parameters of Lasso were set 1e4 and 1e6. With regressions, we want to predict U_mu (relative humidity). By looking at Figure 8, W_mu and T_var have the most correlation with U_mu. As air temperature increases, air can hold more water molecules, and its relative humidity decreases whereas when temperatures drop, relative humidity increases. Naturally, a horizontal viewpoint decreases when there are mists and fogs, which are caused by humidity.

For classifying, we selected the K-nearest neighbours algorithm and logistic regression. In the training phase, K-nearest neighbours store the feature vectors of the training set in the multidimensional feature space. It predicts the target by examining its nearest feature vectors in the space. K-value is about how many boundaries there are between different classes' feature vectors. After iterating in range 1 to 100 and comparing results between different K-values, we concluded that 48 is the best K-value for the accuracy of the model. Logistic

regression is a statistical model which uses a logistic function to predict binary values. Logistic function predicts probabilities by studying the features and chooses right class for data. We use classifiers to predict if the day is dry or wet.

Experiments and results

	Mean Squared Error
Linear Regression	2.429
Linear Regression with PCA	4.266
Lasso Regression	2.430
Lasso Regression with PCA	4.265

Figure 10, Regression Performances.

	F1-Score	Accuracy
KNN	0.772	0.797
KNN with PCA	0.772	0.797
KNN with stratified datasets	0.764	0.786
Logistic Regression	0.768	0.794
Logistic Regression with PCA	0.760	0.786
Logistic Regression with stratified datasets	0.776	0.791

Figure 11, Classifier Performances.

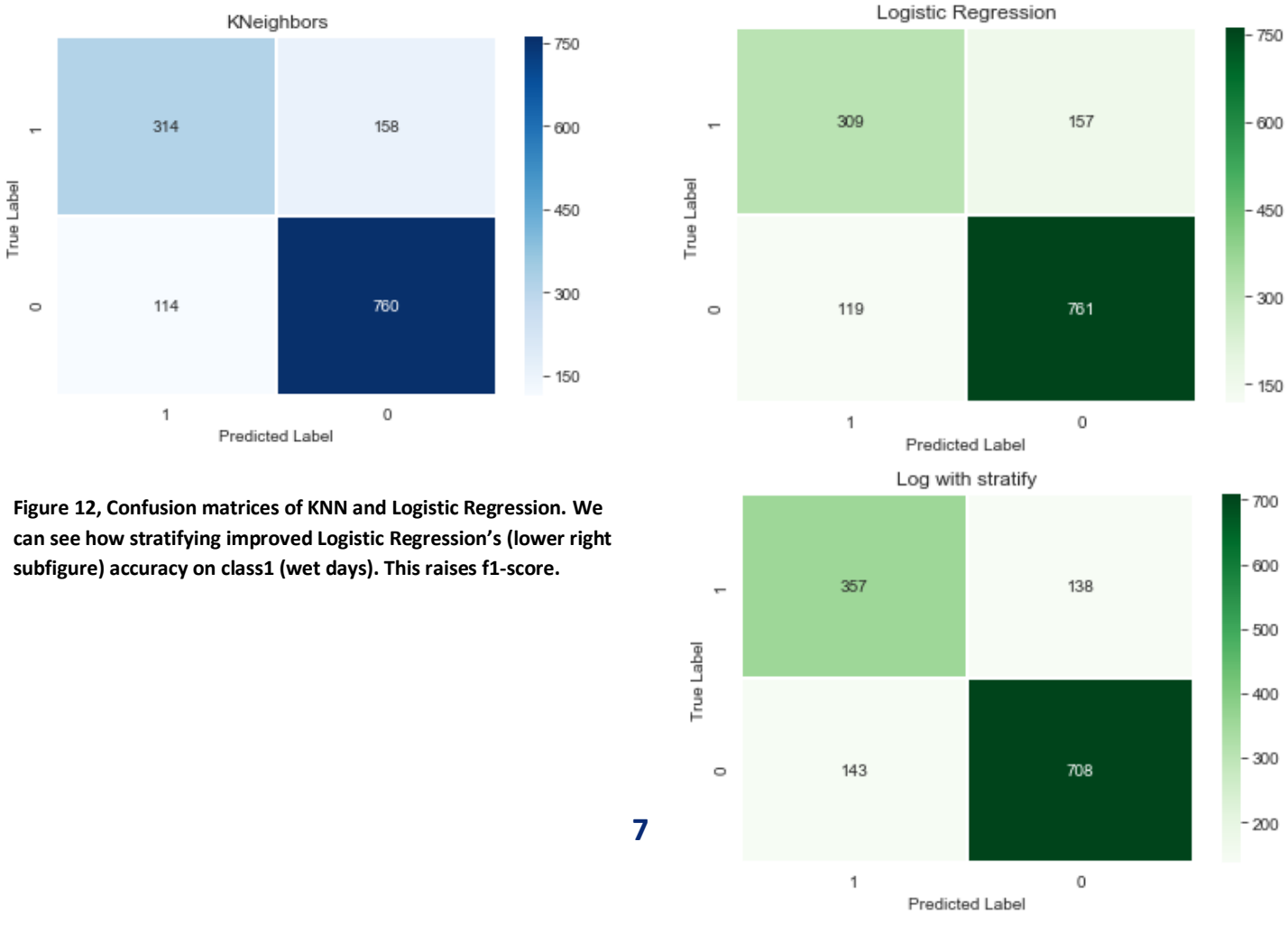

Figure 12, Confusion matrices of KNN and Logistic Regression. We can see how stratifying improved Logistic Regression's (lower right subfigure) accuracy on class1 (wet days). This raises f1-score.

Conclusion and discussion

For measuring to performances, we selected mean square error, f1-score and accuracy. Mean squared error measures the average squared difference between the estimated values and the actual value. We used this measurement for regressions. F1-score measures the accuracies between classes. This is important for testing if the model overfits for one class. Accuracy measures how many predictions are correct. We used these measurements for classifying.

As we can see from results, PCA didn't improve the accuracy of the model. Linear models' performances were almost identical. We didn't expect by reducing components would double mean squared error of regressions. Probably the few last components got result changing information for predictions. Removing a strong feature increases mean squared error whereas a feature whose coefficient in linear regression is approx 0 doesn't change the result so much. K-Nearest neighbour algorithm have a bit better performances than logistic regression. The reason for KNN's performance is non-linearity of the model which can take into account borderline cases of the dataset. In a nutshell, PCA lowered or kept an accuracy same.

Overall performances of our model are close to what we can achieve with traditional methods. In a confusion matrix, classifiers had better accuracy on dry days than wet days. The cause for that could be unbalanced classes in the dataset. This could be fixed by stratifying training/testing sets and acquiring more data of wet days. We think that deep learning with neural networks would improve the accuracies of models since it is a non-linear model and take into account borderline cases better than linear models.

References

1. https://scijinks.gov/forecast-reliability/
2. https://sciencing.com/temperature-affect-barometric-pressure-5013070.html
3. https://ambientweather.net/help/what-humidity-level-does-fog-mist-and-rain-occur/