Analysis of Weather measurements using Regression, Classification and PCA

Data Science, December 2019, Espoo

Abstract

The work considers an analysis of the data measurements collected by weather station 2978 in Helsinki from September 2006 to May 2019. The primary aim of the study was to pre-process and explore the data, predict temperature of the weather and classify the daily conditions as dry or not dry, where latter correspond to the numerical bound expressed in millimetres of moisture. In addition, efficiency and relevancy of used machine learning methods have been revealed, tested and compared with proper conclusions deduced. In particular, the analysis was implemented using linear regression, polynomial regression, k-nearest neighbors' classification, logistic regression, and principal component analysis (PCA). It was shown that: with the given weather dataset linear regression and polynomial regression are able to yield exactly the same accuracy and errors without the use of principal data analysis. However, optimized knn algorithm is capable to slightly improve results. Principal components analysis could considerably reduce the complexity of the data without losing variance and keep the quality of results almost on the same level or even enhance the accuracy of the analysis. As a result, discussion for further analysis and optimization has been done.

For the reader: Dear reader, if you familiar with machine learning techniques, please skip the chapter "Methods" and jump straight to the chapter 4. Experiments and Results. However, if you interested in "how things work under the hood", we tried to give you a simple description of intuition and mathematics behind them. Please use the table of contents to in the text.

1. **Introduction.**

Nowadays, the significance of data-driven activities has grown tremendously. Indeed, most of the activities related to the service design, engineering, analytics, decision-making process rely on the analysis of past statistical data. Therefore, significance of the up-to-date data science tools, which are able to provide powerful computational methods undoubtfully important for the modern industry. One of core technologies is machine learning.

The project has been assigned by the Data Science course and it aims to fulfil two goas: 1) Based on the available statistical data, predict the weather temperature in Helsinki and classify the conditions of the weather as dry or not-dry. 2) Compare different machine learning method for the aforementioned task in order to reveal their effectiveness, drawbacks and potential for optimization. As a personal improvement, the participants aim to get good basic understanding of aforementioned machine learning methods in order to understand their applicability, restrictions and also practice with processing, visualization and analysis of real data.

The project explores processing, computation, analysis and utilization of the numerical weather measurement data through the implementation of machine learning methods. Due to the fact that there have been done quite significant amount of research in the area, one of the main approaches was to rely on already existent methods and intend to show their relevancy as well as to identify potentially weak points. In particular, the used methods were linear, polynomial and logistic regression, classification with K-Nearest neighbours and Principle component analysis. In addition to that, optimization of some algorithms would be done in order to reveal their potential capabilities for improvement.

The implementation of the task is important since it helps describe usage of machine learning method for the particular scientific/engineering problem at hand and answer the basic questions about implementation related to the topic.

2. **Data analysis**

2.1 Data overview

First of all, we get the X_train set from the file ‘weather_data_train.csv’, in addition to the ‘datetime’ column as the index, the training set consists of 16 columns containing the means and variances of 8 different features with the data type float 64. There are 3140 data points in the set. Apart from this, we have the X_test set from the file 'weather_data_test.csv' with 1346 data points to perform validation on our experiment. The X_train and X_test have the same number of columns.

Secondly, corresponding to 2 data sets, there are two sets of labels - labels and labels_test with dimension (3140 x 2) and (1346 x 2) respectively, derived from 'weather_data_train_labels.csv' and 'weather_data_test_labels.csv'. The sets contain two columns “OBSERVED” and “U_mu”. The OBSERVED (int64) variable is our label, 0 indicates not dry, 1 indicates dry, the set will be used mainly for the classification task. At the same time, U_mu (float64) variable represents the humidity in percentage - which will be used for the prediction task using Linear Regression and Polynomial Regression.

There are no null fields, which was consecutively verified with ‘X_train.isna.sum()’, ‘X_test.isna.sum()’, ‘labels.isna.sum()’ and ‘labels_test.isna.sum(). Therefore, the data can be used for analysis as given, data-cleaning step is not necessary.

2.2 Plots

With the view to better understanding our dataset by visualization, as well as, getting insight on how different features affecting each other and between features and output, given datasets were plotted using different methods including histogram, pair plots, and correlation matrix.

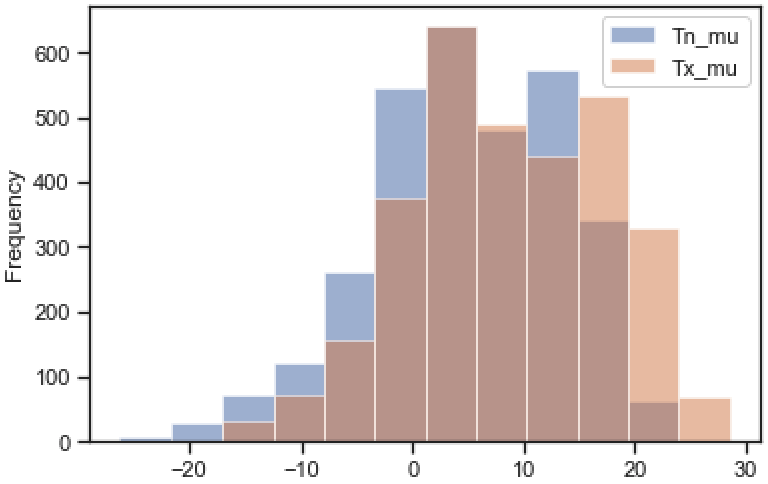


Figure 1. Histogram of Tn_mu, Tx-mu

The histogram of Tn_mu and Tx_mu looks as expected, visually if one shifts the graph of Tn 5 degrees to the right, the two graphs almost converge. From the graph, the maximum air temperature, in degrees Celsius, varies from -15 to ~27 and the minimum air temperature, in degrees Celsius varies from -20 to ~25, with an almost identical variation of frequency.

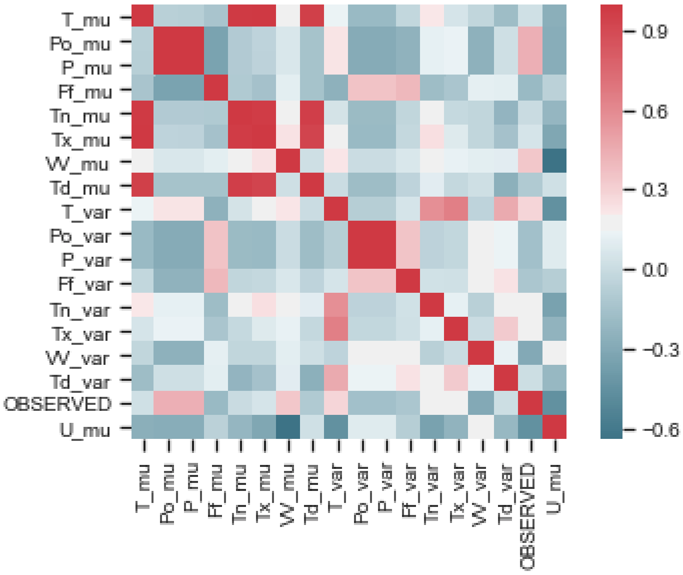


Figure 2 Correlation matrix of the features.

By examining the graph of correlation, it is recognized that there is a highly positive correlation between Po_mu and P. At the same time, T, Tx, and Tn are also highly positively correlated, the matrix can be simplified by combining these fields.

Whereas U_mu and VV_mu are highly negatively correlated to each other, which means, once, one of the factors increases the other one might decrease.

Furthermore, because U_mu is the mean of humidity and VV_u is the mean of horizontal visibility, in km. In conclusion, horizontal visibility is the strongest predictor of humidity. This means that when there is a rise in the horizontal visibility, it is likely that there is a fall in the humidity level.

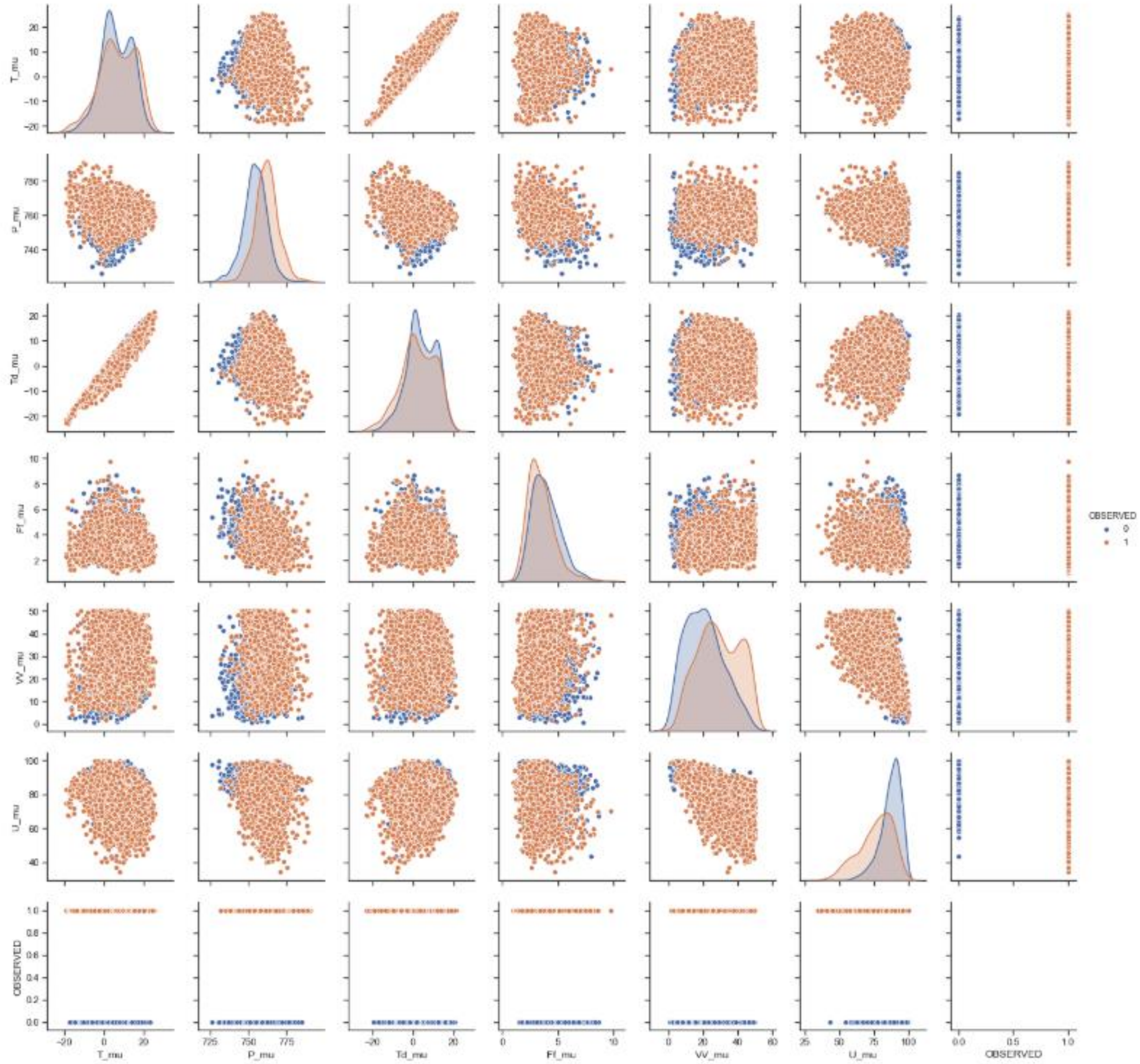


Figure 3 Pairplot of the features.

Last but not least among the graphs of preprocessed data, the data was pair-plotted for T mu, P mu, Td mu, Ff mu, VV mu, U mu with each aspect plotted in 2 different colors corresponding to its OBSERVED label with blue for 0 (not dry) and orange for 1 (dry).

In terms of significant correlation, it is easily noticeable that in Tn_mu and T_mu pair-plot graph, the data is converged to the shape of a line, which means they have a positive linear relationship.

2.3 Principle Component Analysis

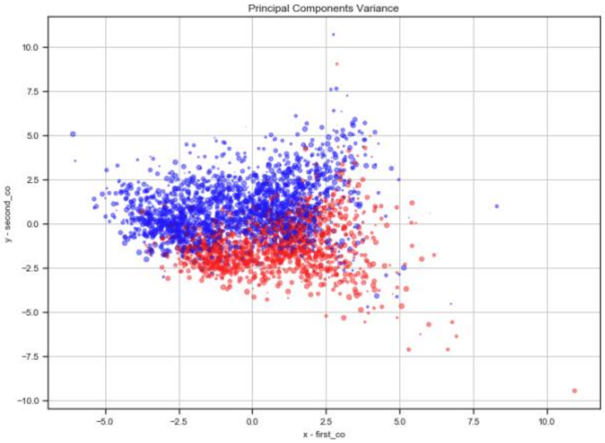


Figure 4. Projection of 1st and 2nd principal components to the axis.

Principal Component Analysis was performed on the dataset using 2 components by firstly standardizing our X_train dataset using the StandardScaler model, then fitting and transforming our data with “PCA(n_components=2)”. Which resulted in a (3140, 2) matrix of projection of 2 first components. The scatterplot of Principal Components Variance was colored according to the OBSERVED labels, where blue means dry and red means not dry.

Based on the observation, the blue dots and red dots can almost be separated by drawing a line along the x-axis. This means there is a very strong correlation between the value of the second component and the OBSERVED label and possibly PCA can be used to improve the accuracy of the classification.

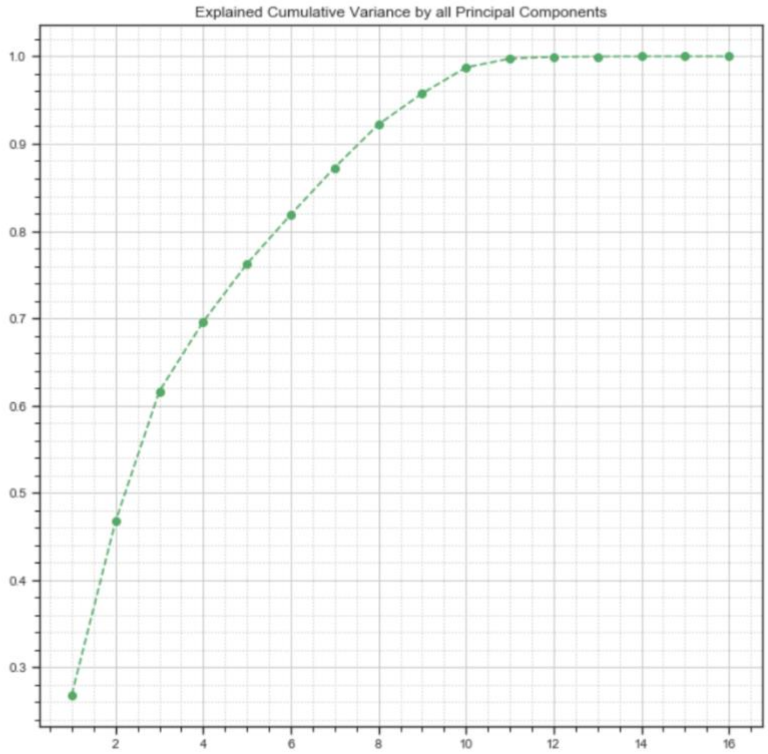


Figure 5. Graph of Variance explained by 16 principal components.

The graph of Explained Cumulative Variance of all Principal Components by ratio showed that the Y-axis reached 1.0 in the 12th component, which means, components from 13th to 16th explain only a small portion of data, and therefore, can be dropped. For the purpose of analysis tasks such as Regression and Classification, one can select from 10 to 12 dimensions for optimal computation and results.

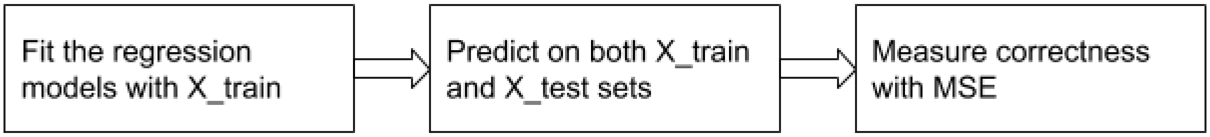
3. **Methods.**

3.1 Prediction with different regression models.

3.1.1 General approach:

Throughout different experiments, the humidity level was predicted using both Linear Regression and Polynomial Regression without PCA, and with PCA, in which the data was first standardized using "StandardScaler" model. Afterward, the accuracy of the models is measured with Mean Squared Errors (MSE).

The diagram below shows the whole approach applied to both Linear Regression model and Polynomial Regression model, in which the data were not preprocessed with PCA:


Fit the regression models with X_train
Predict on both X_train and X_test sets
Measure correctness with MSE

On the other hand, this diagram shows how the experiments were done with the adoption of PCA.

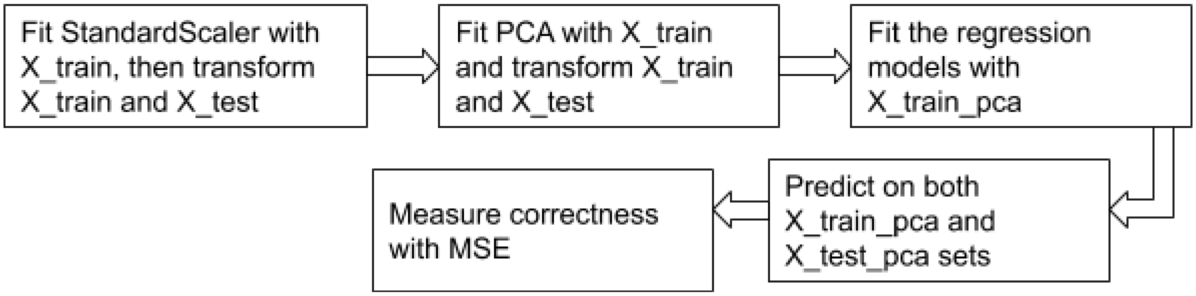

Fit StandardScaler with X_train, then transform X_train and X_test
Fit PCA with X_train and transform X_train and X_test
Fit the regression models with X_train_pca
Predict on both X_train_pca and X_test_pca sets
Measure correctness with MSE

Figure 6. - 6.1 Block diagram of regression process.

Linear Regression:

Linear Regression is a technique to understand relationships between different features and the target parameter, one can use linear regression to forecast a parameter based on collected data. The problem can be formulated as:

$$y = h(X) = \theta_0 + \theta_1 \cdot x_1 + \ldots + \theta_m \cdot x_m$$

Figure 7. Linear regression formula.

Where Y is the output variable, x_1 … x_m is input variables coming directly from our training data, and thetas are the unknown about the unknown coefficients for a hyperplane that is best fitted to the data. Linear Regression model finds them by solving a cost function that determining how good they are. Linear Regression was adopted to the experiment because first of all, it was one of the most simple Regression techniques, secondly by plotting the humidity (U_mu) and the mean of horizontal visibility (VV_m) it is noticeable that they have a linear relationship, one can almost draw a line with a negative slope on the figure.

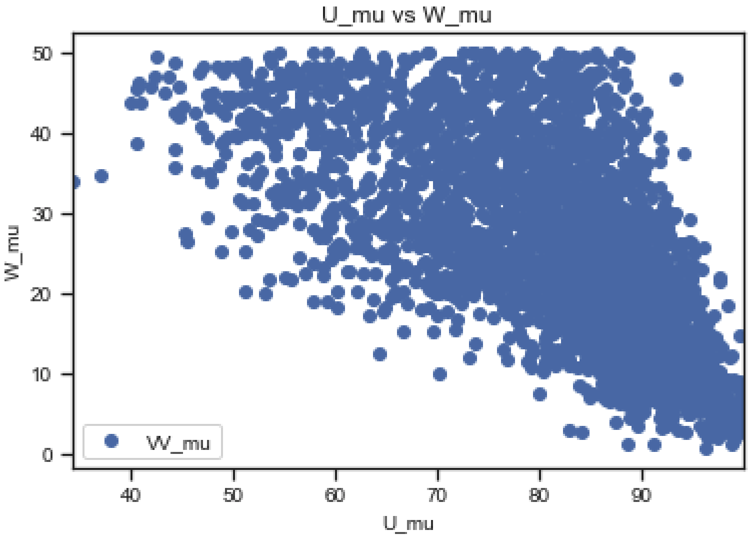


Figure 8. Regression Scatterplot of humidity (U_mu) and horizontal visibility (VV_m).

Polynomial Regression:

However, as most of the features are not linearly related, so when they are used to fit a line like in linear regression model, it might result in underfitting. In order to better fit the data, polynomial regression was experimented. Under the hood, the model firstly transforms the input features into higher degree features. Then, the transformed data is fitted into the linear regression model.

3.2 Classification

K-Nearest neighbours classification algorithm. The idea behind the KNN algorithm is rather simple: it assumes that all «similar» examples in the feature space are living close to each other. Starting from this assumption, one needs to choose the potential number of available classes (nearest neighbours) in the space and establish a similarity measure (for instance geometric distance like Euclidean norm). Relying on these two facts, algorithm computes the measure for every instance of the feature space and distributes the inhabitants into the respective classes.
Classification.

The fact why KNN algorithm was chosen is closely related to its advantages:

1) Implementation of the KNN is rather simple and can be done with couple of lines of python code.
2) It is quite robust and can classify even linearly non separable data quite efficiently.

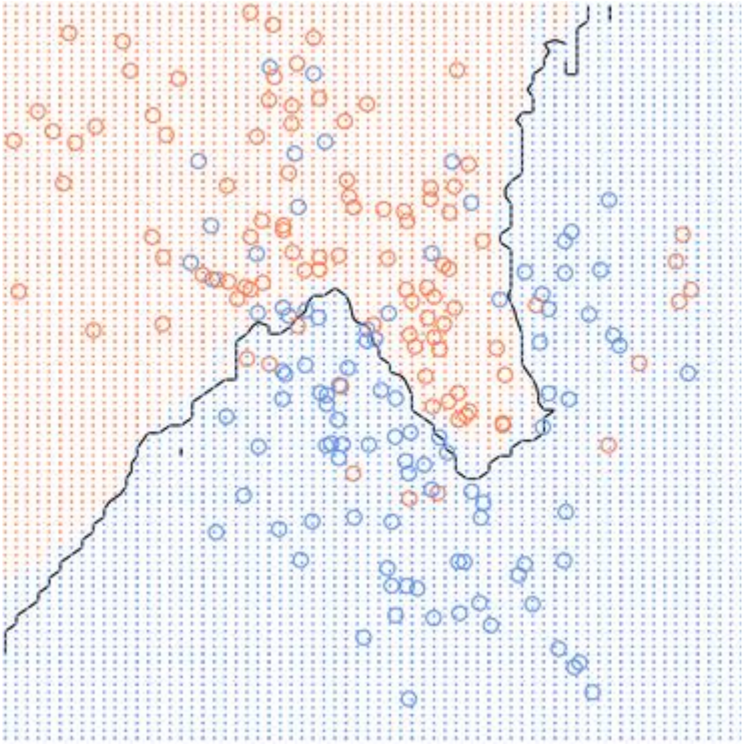


Figure 9 kNN Classification vizualization.

One of the most important questions related to the KNN is how to choose optimal number of classes (neighbours) in order to separate the space into respective patterns. In order to answer this question, one more machine learning method is introduced.

K-Fold cross validation is one of the particular ways to measure the quality of build ML model. The idea behind K-fold cross validation is connected to the splitting the data into testing and training set. In a nutshell: the dataset is split into K-equal sections such that every section is iteratively used as a training and validation sets. That is, if one chooses K to be equal to 5, then data is split into 5 parts and first part is initially a validation set, while the latter four parts are used to fit the model. This procedure repeats unless all of the section has been training and validation parts respectively. With the implementation, at every new iteration each part of the data is treated as new observation pattern and if one would measure testing or misclassification error during each of the K-iteration, as a result good averaged estimate would be obtained. That is, this approach of the estimation of different quality measure reveals more objective view on the quality and “skills” of build ML model. The purpose of implementation of K-Fold cross validation was to measure misclassification error with different number of neighbours and take the ones yielding smallest misclassification error. The implementation is thoroughly described at chapter «Experiments and results”.

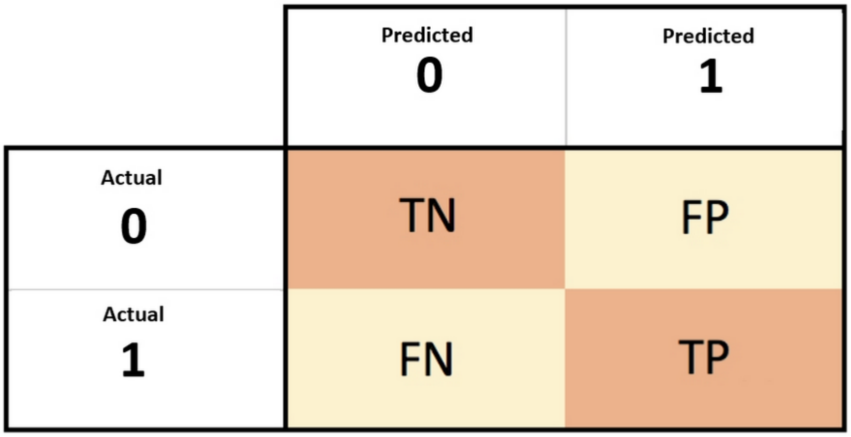
Logistic regression. One can think of logistic regression in the same way as of linear regression, but instead of line discriminator data is fit into logit function. The mathematics of the model intuitively comes from the name: logistic or sigmoid function is fit into the feature space separating classification variables into respective classes. The reason behind usage of logistic function in this project is simple: to give good comparison with kNN classification technique, in order to obtain understanding from better perspective.

$$h\theta(x) = g(\theta T x) = 1/ 1 + e -\theta T x$$

Figure 10. Logit or Sigmoid function.

How to interpret classification? Confusion Matrix.

Confusion matrix is a performance measurement tool for classification techniques. In principal, it is just a tool for visualization of amount correctly and incorrectly predicted class instances.



	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

Figure 11. Example of confusion matrix.

Confusion matrix on the figure corresponds to the most conventional view of confusion matrix. The values of the intersection of diagonal (0,0) and (1,1) correspond to the correctly classified negative and positive instances: true positive and true negative. Values of the opposite places present an amount of faulty classified instances.

3.3 Dimensionality Reduction.

Principal Component Analysis. PCA is dimensionality reduction technique, which is has solid mathematical background. On the conceptual level, PCA aims to eliminate extra unnecessary complexity of the data through the reduction of its dimensionality. Most of the data used in data science analysis and machine learning are high dimensional and has many “not so statistically significant” features. Nevertheless, their existence adds up significant cost to the computational complexity, since it requires more resource to run the algorithms. In addition to that, high dimensional complex data tend to overfit the model. Therefore, dimensionality reduction is necessary procedure almost in every data science project. In a nutshell, PCA works so: 1) It takes the data, finds out the most relevant and most irrelevant features, transforms the data so that relevant features are taking the “importance of most irrelevant” and produces new transformed dataset with reduced dimensionality. On mathematical side, PCA is done through normalization and singular value decomposition of the matrix with data (eigendecomposition if the matrix is positive definite), where the biggest singular values are taking most of the data variance and then transforming into linearly uncorrelated feature vectors, which are indeed new dataset.

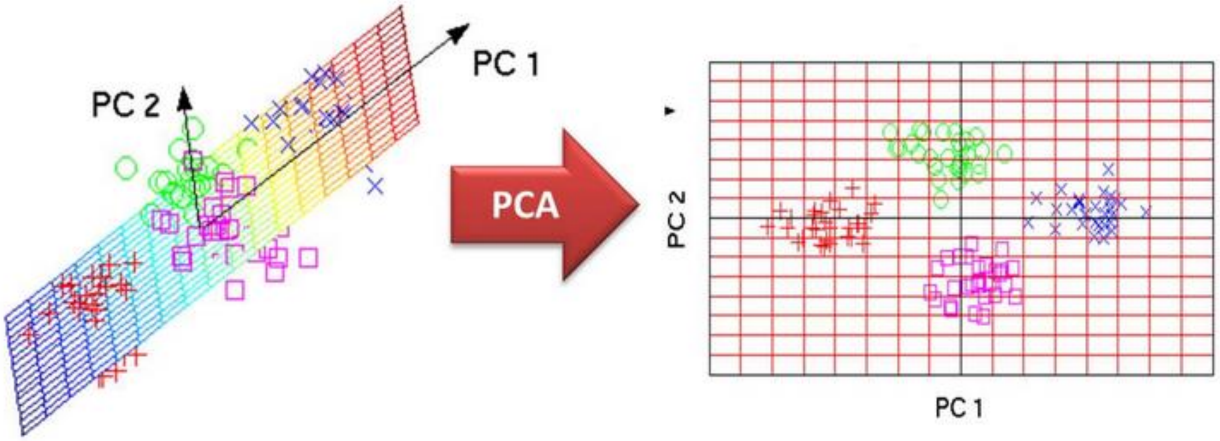


Figure 12. Principal component analysis visualization.

Cumulative explained variance.

Cumulative variance is the sum of all variance covered by all components. When the dimensionality is reduced, the fraction of variance taken by principal component and total variance covered by all components is explained variance.

3. 4 Error Measurement:

Mean Squared Errors:

Mean Squared Errors is a measure of quality of an estimator by calculating the mean of the squares of the difference between the predicted parameters and the actual one.
Formula:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

Figure 13. Mean Square Error formula.

4. **Experiments and Results.**

Both Linear Regression and Polynomial Regression were performed to predict the humidity with and without PCA.

4.1 Linear Regression & Polynomial Regression:

Firstly, Linear Regression was used to predict the humidity level on both X_train dataset (dimension: 3140 x 16) and X_test dataset (dimension 1346 x 16). The MSE measured on the training set X_train was 1.85 and on the testing set X_test was 2.12.

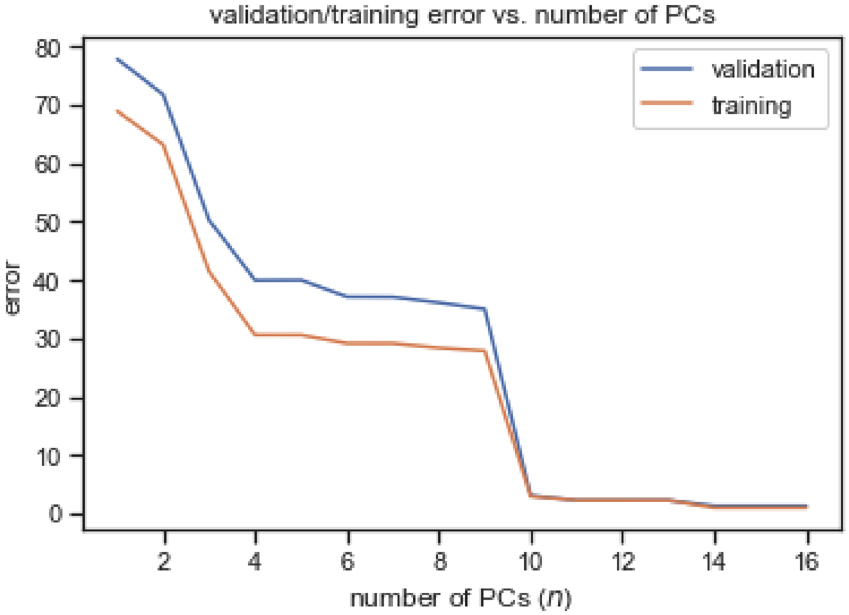
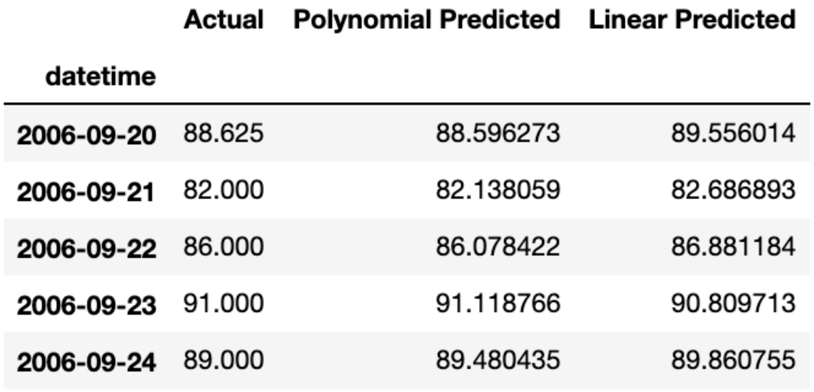


Figure 14 Validation/Training error vs number of features.

Secondly, we experimented with preprocessing the data with PCA on different numbers of components ranging from 1 to 16 on both training and testing sets, then applied Linear Regression. The validation errors were higher than the training error when from 1 to 10 components were used, and from 10 to 16 components, the validation and training errors were almost the same. However, the results from PCA was not very optimal comparing to performing the linear regression without PCA. On using 12 components, the training errors were ~ 4.39, and testing error was ~ 4.26.



	Actual	Polynomial Predicted	Linear Predicted
datetime			
2006-09-20	88.625	88.596273	89.556014
2006-09-21	82.000	82.138059	82.686893
2006-09-22	86.000	86.078422	86.881184
2006-09-23	91.000	91.118766	90.809713
2006-09-24	89.000	89.480435	89.860755

Figure 15 Comparison of the actual humidity values with predicted values using Polynomial and Linear Regression.

Thirdly, the prediction was performed with the Polynomial Regression model, the accuracy was significantly improved compared to the Linear Regression model with training error ~ 0.29 and testing error ~ 0.27.

Lastly, with the same approach as in linear regression, we pre-processed data using PCA with a range of components from 1 to 16. However, in this case, the testing errors were higher in most cases, and the errors were significantly higher when applying PCA. On using 12 components, the training errors were ~ 3.09, and the testing error was ~ 3.18.

4.2 Classification

As it was mentioned before, the classification has been done with K-Nearest Neighbours. The idea was to implement kNN with and without PCA and deduce how dimensionality reduction might affect the accuracy of the results. Both approaches have been done with optimal number of k-nearest neighbours, however, initially the kNN algorithm has been run with random number of neighbours yielding 73% of correctly classified examples. Nevertheless, as it was mentioned, the optimal number of components have been chosen with K-fold cross validation. In scope of the given project, the K-fold cross validation have been implemented in a such way: 1) Dataset has been divided into K = 20 sections. 2) For neighbour i in a range 1<=i<=50 (excluding even numbers) the K-fold algorithm has run and recorded respective accuracy measure. 3) After CV have been done for all i'th neighbours, accuracy errors have been transformed to misclassification errors and the i'th number of neighbours yielding the lowest error have been chosen.

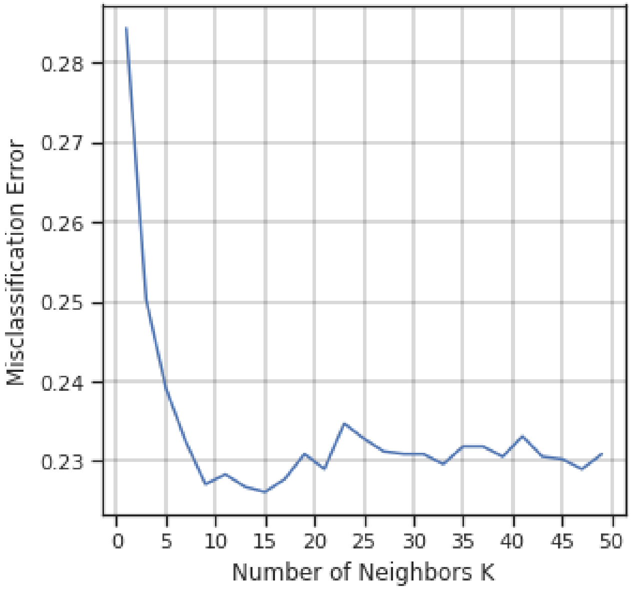


Figure 16. Misclassification vs number of K neighbours.

From the graph it is clearly seen that choosing 11 neighbours yields the smallest misclassification error of approximately 20%.

Therefore, respective confusion matrix with and without normalization is:

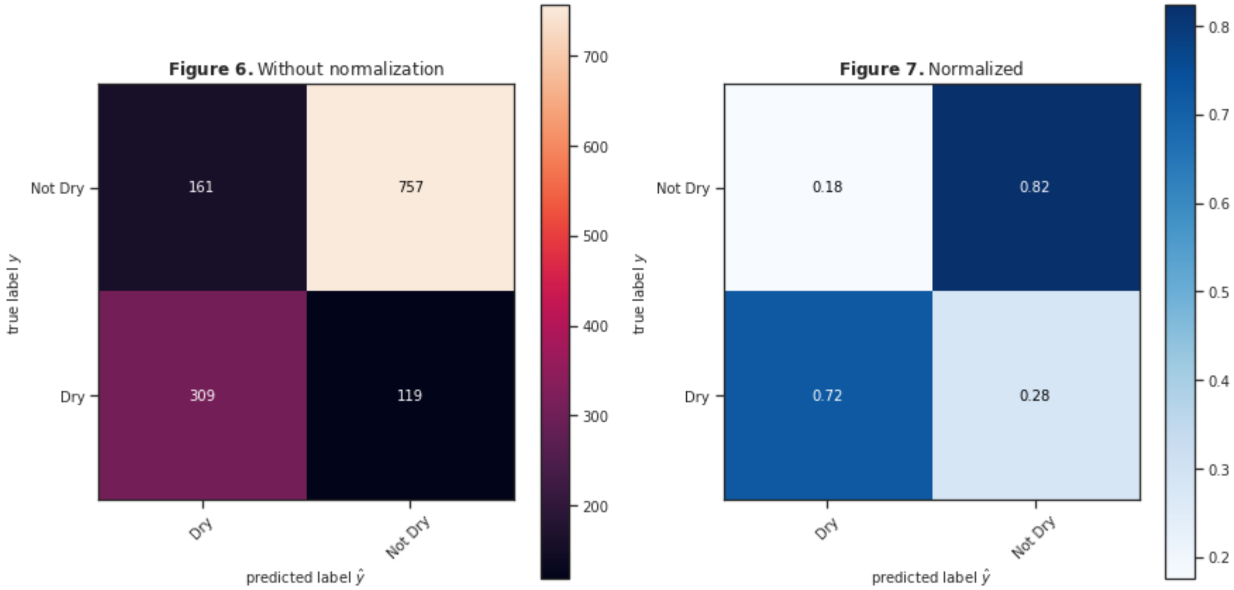


Figure 17. Confusion matrix of kNN before PCA.

From the confusion matrix one can deduced that 309 days in a period have been classified as dry days indeed turned out to be dry, while 757 days have indeed been wet. At the same time, misclassification of 119 days as false not dry and 161 days as false dry takes place yielding approximately 20% of misclassification

Second attempt to classify the weather conditions was intended to be done after dimensionality reduction of the data. Choosing correct number of Principal Components is Another very important issue to deal with and this decision has been done with help of the plot of cumulative explained variance taken by the components of the data. The solution has been to choose number of components which are taking 98-99% of data variance.

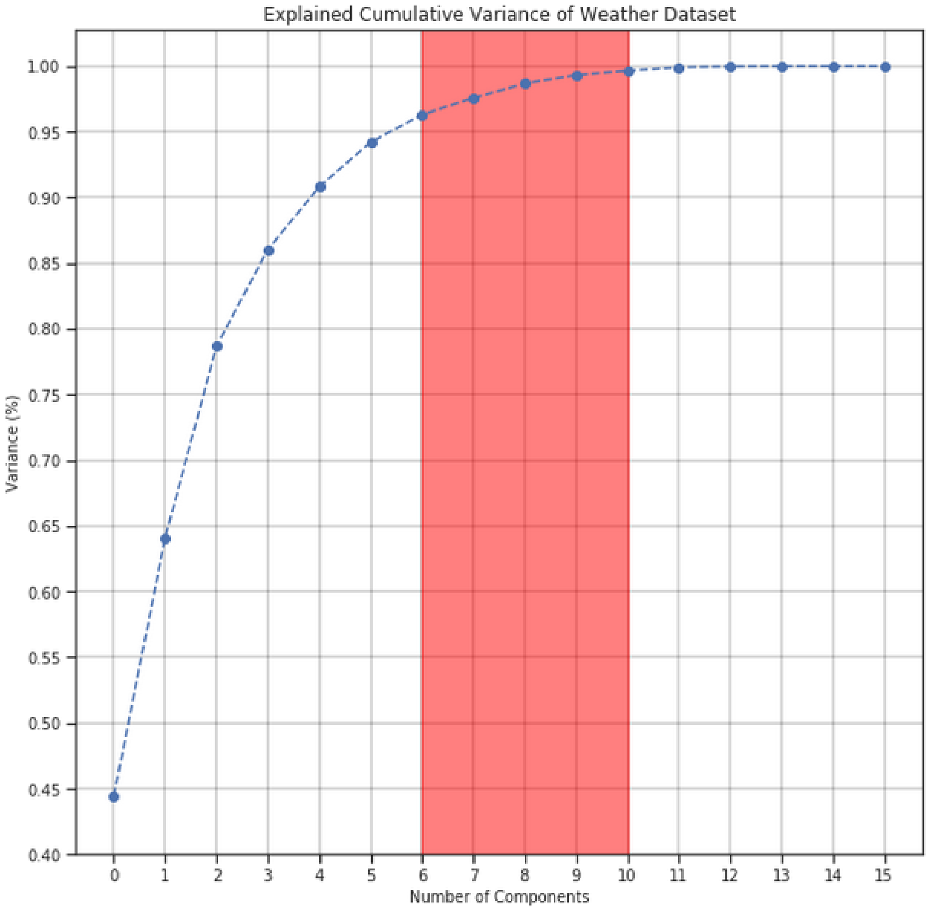


Figure 18. 98-99% of CEV taken by 9 components.

One can deduct from the plot that 9 components are taking 98-99% of the data variance. Therefore, kNN classification have been implemented again with 9 principal components, which resulted in optimal number of neighbours to be equal to 47 and respective misclassification error of approximately 19.2%.

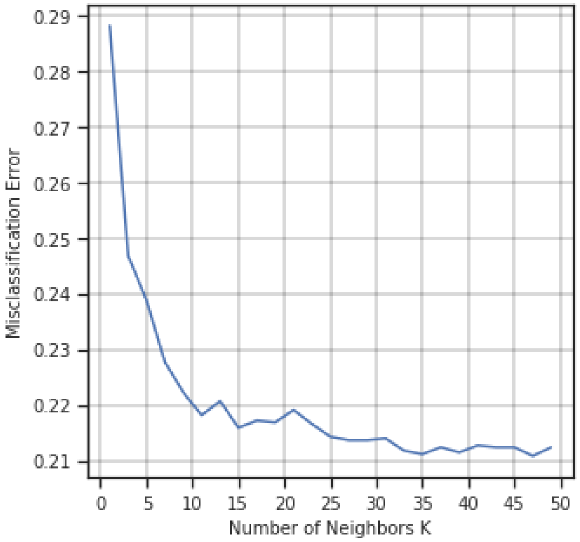


Figure 19. Misclassficiation error vs K number of neighbours.

Respective classification results are seen from the confusion matrix.

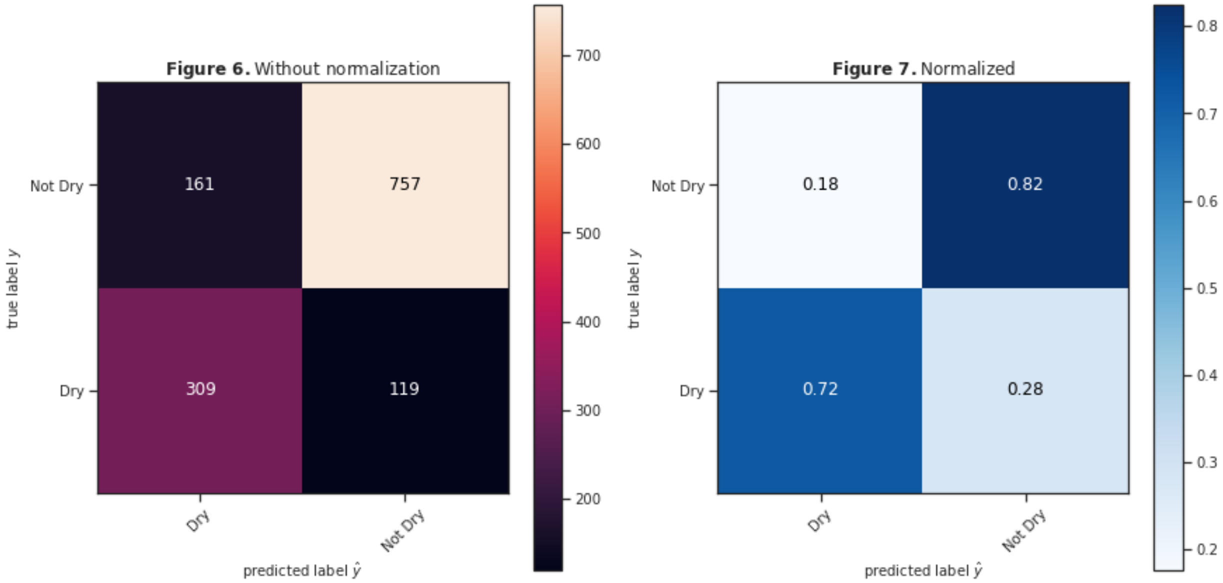


Figure 20. Confusion matrix after PCA.

As a result, one can easily notice that amount of correctly and incorrectly classified instances hasn't changed after PCA, however, the misclassification error was lowered on 0.8%, yielding total misclassification error equal to approximately 19.2%.

Finally, in order to obtain good classification comparison for kNN, the data has been classified with logistic regression. The latter one has produced misclassification error without PCA equal to 23% and with PCA equal to the 20%. The confusion matrices of logistic regression are eliminated here.

5. Conclusion and discussion

5.1 Regression

By comparing the experiment results two Regression models Linear Regression and Polynomial Regression with and without PCA, it can be concluded that, first of all, the Polynomial Regression gives a more accurate prediction with MSE ~ 0.27 compared to MSE ~ 2.12 using Linear Regression on the testing set, which means that the correlation between the data-set and the targeted value is non-linear, and transforming our linear equation into a non-linear equation was necessary. The results computed using PCA with 12 components gave a slightly worse result with MSE ~ 4.26 using Linear Regression and MSE ~ 3.09 using Polynomial Regression, which proves that PCA might be effective in decorrelating the data and allows one to remove dimensions not describing any variance in your data, improving the computation performance, but still is capable of giving decently accurate results. In this particular case, 4 components were removed, but only ~ 2.14 loss in accuracy, which might be particularly helpful in computing a very large-scale dataset. However, in this particular case, PCA did not enhance the correctness of the regression tasks, while it might in other data set.

5.2 Classification

The analysis of results obtained by the kNN algorithm is closely tightened to the analysis of obtained confusion matrices and respective misclassification errors. In general, kNN algorithm has shown a relatively good performance. Misclassification error on the bounds of 20-19.2 % is indicator of averagely tuned algorithm, however, not performing its full potential. In addition, optimization procedure with K-Fold cross validation and determination of the optimal number of nearest neighbours has slightly improved the quality of classifier by lowering the misclassification error. One of the main reasons for this to happen is the type of data we had and classification analysis we perform. At this point it is good to pay attention to the fact that in principle, kNN algorithm is able to yield a good performance on linearly non-separable data with low dimensionality. In our case, the data high dimensional and non-linearly separable. Therefore, it is not the best work field for kNN, but nevertheless, it managed to give quite decent degree of classification. One of the reasons for that was optimization to choose optimal number of k-nearest neighbours. Indeed, in high dimensional data incorrectly chosen k can significantly lower the misclassification error. In the case of given project, the dataset has been heuristically tested with randomly chosen k which has yield around 73% of correctly classified instances in full dimensionality with all components, which is 10 percent less than the result obtained after K-Fold cross validation. Another progression in lowering of misclassification error has been motivated by PCA, which reduced dimensionality to 9 components instead of 16 and manage to improve classification metric on 0.8%. This is related to the aforementioned fact that kNN better deals with low dimensional data. Another benefit of this is reduced computational complexity, which results in more useful resource utilization. The latter is also clearly reflected in the reduced amount of computational time for 9 components compared to 16.

In conclusion, Principle Component analysis turned out to be quite useful and effective tool in the scope of given project and machine learning techniques since for both tasks (regression and classification) it managed to keep the outline of different error measurement (MSE for regression and misclassification error for kNN) on the same level while significantly lowering the dimensionality and therefore improving computational performance of the algorithms. K-Fold cross validation is useful technique for obtaining more objective perspective of the particular dataset, however, the number of K-folds has to be also chosen smartly. kNN Classification is a good tool, but one has to take into consideration its limitations, one of the is quadratic growth of computational complexity with amount of training data. Finally, aspect of bias haven’t been taken into account and in order to improve quality measures of the given model, one can study it in potential future research.

References.

https://towardsdatascience.com/polynomial-regression-bbe8b9d97491

https://towardsdatascience.com/linear-regression-using-python-b136c91bf0a2

https://towardsdatascience.com/linear-vs-polynomial-regression-walk-through-83ca4f2363a3

https://www.linkedin.com/pulse/math-behind-linear-regression-enrico-d-urso/

https://en.wikipedia.org/wiki/Mean_squared_error

http://theprofessionalspoint.blogspot.com/2019/02/advantages-and-disadvantages-of-knn.html

https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/

https://www.statisticssolutions.com/what-is-logistic-regression/

https://www.quora.com/Whats-the-point-of-polynomial-regression-if-I-can-just-use-multiple-linear-regression

https://towardsdatascience.com/understanding-and-reducing-bias-in-machine-learning-6565e23900ac