**Data Analytics with Python – Assignment 2**

**Group 6 Members:**

Ma Estela Arenas

Sean Howman

Yuxiao Liu

Feng Nie

**1. Background**

It has often been observed that energy consumption tends to be at its highest on days with hotter temperatures. In this project, our group will develop models that predict the maximum daily energy usage and pricing category based on provided weather data.

**2. Purpose**

These models can be used to predict likely energy demands based on a weather forecast, which can help energy companies understand plan for future usage, and help businesses plan when to conduct energy-intensive operations.

**3. Process**

**3.1 Data Mining**

Two datasets were provided in csv format, one for weather data with 243 rows and 21 columns, which has blanks and columns with float, integer and string data types. The other dataset contains the price-demand data with 11,664 rows and 4 columns. Date range used in this project is between 1st of January and 31st of August 2021.

**3.2 Data Cleaning**

*What wrangling and aggregation methods have you applied? Why have you chosen these methods over other alternatives?*

SQL, Python, Excel and OpenRefine were used to wrangle and aggregate the datasets. SQL, Excel and OpenRefine were chosen for ease of use in data cleaning while Python enabled us to apply our learnings from this course.

The price-demand dataset was modified in such a way that each day was represented by one instance (one row) with each day containing the maximum energy demand and the maximum price category for that day. The column containing the information on region was removed as superfluous and the date format was modified to match the date column in the weather dataset.

Using numeric and text facets in OpenRefine, the weather dataset was explored for blanks, outliers and nonnumeric/non-text data. The data was found to contain several values which were missing or did not match the prevalent data type in the column.

Where the wind direction and wind speed data columns contained the wrong data type, “0” was substituted for “Calm”, and “Calm” for “0” in these columns respectively. The columns were then transformed to numeric or text data as necessary.

Missing rainfall and wind direction values were imputed by inferring similar values from related data on that day and surrounding days, while missing times of maximum wind gusts were filled using the average time of maximum wind gusts. This method was chosen as opposed to simple imputation or linear regression as it was thought values were likely to correspond with surrounding values rather than values from the larger feature dataset.

For the remaining missing values, scatterplot facets were generated in OpenRefine as a quick-look correlation with other features. A feature that correlated with the missing value feature was then selected and the two features plotted against each other in Excel with a linear relationship determined and the resulting equation used to impute the missing value for each instance. This method was chosen as simple imputation using the whole feature dataset would give a figure calculated across 8 months. An example in Figure 1 and Figure 2 below show a strong correlation between the minimum temperature and the 9am temperature and the equation which was used to impute a missing minimum temperature value. One row in particular (data from 08/07/2021) contained several missing values. Some of the imputations for these data were based on lower confidence correlations, however after discussing whether the row should be deleted from the dataset, it was decided that the imputations were robust enough not to greatly affect the model output.

After cleaning the data, the price-demand dataset and weather dataset were combined, with the date as the common feature.

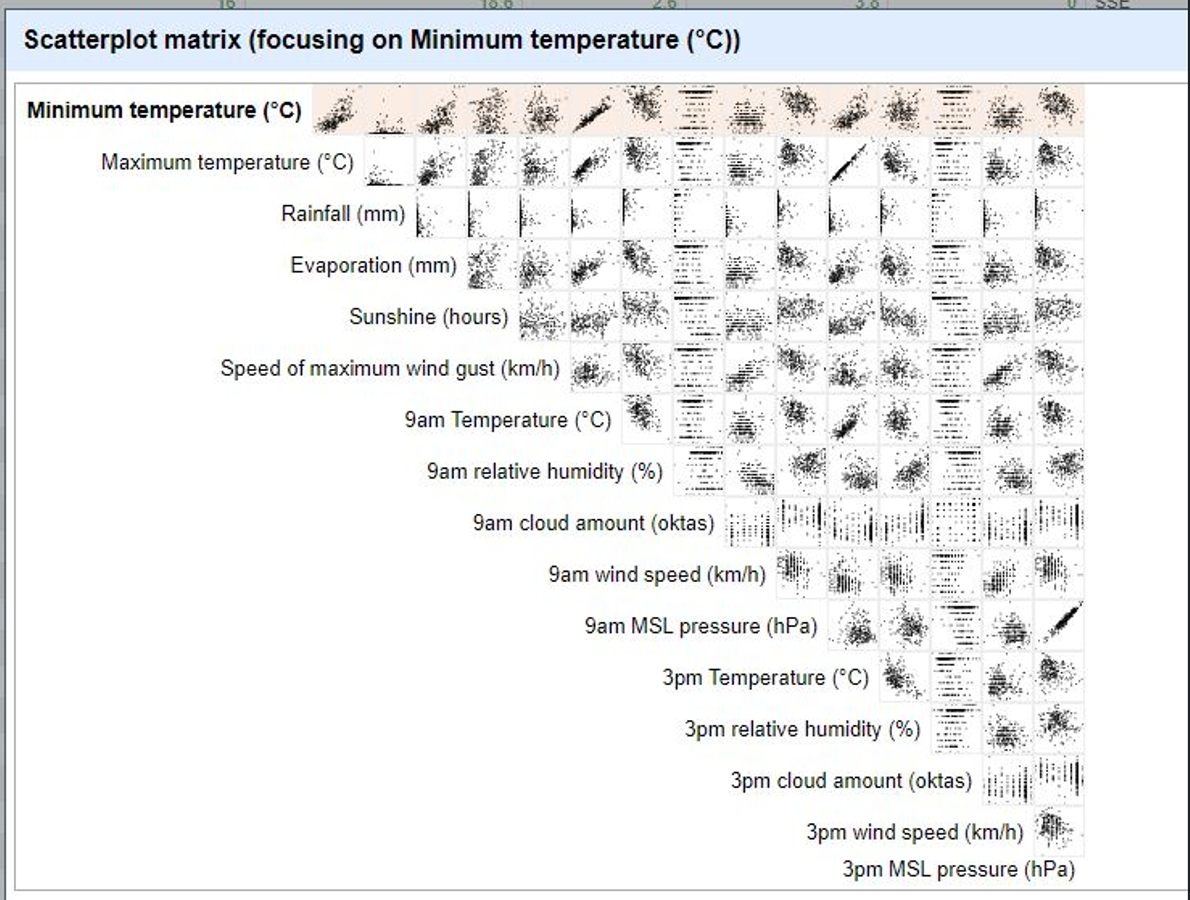


Figure 1: Scatterplot from Google OpenRefine showing correlations between features in the weather dataset. These plots used for data imputation and model feature selection.

Figure 2: Excel plot showing linear relationship and equation used for imputation of missing minimum temperature value.

Figure 2: Scatterplot from Excel for minimum temperature vs 9 am Temp in oC

**3.3 Data Exploration**

We’ve selected features by observing scatterplots in OpenRefine, and where features appeared to be correlated with other features, one of those features was removed (usually the 9am or 3pm “snapshot” data). We’ve also removed wind direction features as they are discretized features and even if we change it to other data formats it will not add value to our model and by domain knowledge, we think they are unnecessary. We also chose minimum temperature as it works better than any other temperature features.

**3.4 Model Building and Prediction**

*How have you gone about building your models and how do your models work? How effective are your models? How have you evaluated this?*

**3.4.1 Linear Regression Model**

The goal of this model is to predict the maximum daily energy usage based on provided weather data. The output is numerical data thus we will be using linear regression to build our model.

Our assumptions for using linear regression are:

* The dependent and independent variables are both numerical.
* There is a linear relationship between the dependent and independent variables.
* There are no significant outliers.
* There is independence of observations.
* The data shows homoscedasticity, which is where the variances along the line of best fit remain similar as you move along the line

In order to create the model using linear regression algorithm, we did the following:

1. Import required libraries.
2. Load the combined data set.
3. Select the features.

We’ve used Pearson correlation coefficient to see which features are relevant to this model. We’ve checked the correlation between the highest demand and the numerical weather features. We’ve also reviewed the independence of the possible features. We’ve selected the features with an absolute Pearson correlation coefficient over 0.3. Finally, we’ve chosen the one with the highest correlation with the output, which is the minimum temperature.

1. Separate dataset into 70% train and 30% test parts.
2. Train the model and predict the result with test data.

We’ve done k-fold cross validation with k value of 10 and we have observed that as the k-fold value was higher, the accuracy score is better.

1. Evaluate the result.

Based on the results of our linear regression model, with an r2 score of 0.25, we can say that there is some linear relationship existing between the minimum temperature and the demand usage. This indicates that some but not all of the variation in the demand is explained by variation in the features. The minimum temperature seems to be the most valuable feature in predicting the maximum demand usage.

**3.4.2 Classification Model**

The goal of this model is to predict the maximum price category based on provided weather data. The output is categorical data thus we will be using either Decision Tree or K nearest neighbor (KNN) classifier to build our model. Our group has decided to try both algorithms and see later on which one of them will produce a better model.

Now that the data is totally prepared, the classifier is instantiated and the model is fit onto the data. The criterion chosen for this classifier is entropy. Once our model fits the data, we tried predicting values using the classifier model. This is often done in order to perform an unbiased evaluation and get the accuracy score of the model. We’ve done parameter tuning to select the best model.

**3.4.2.1 Decision Tree**

In order to create the model using the Decision Tree algorithm, we did the following:

1. Import the required libraries.
2. Load the combined data set.
3. Select the features.

We’ve chosen minimum temperature, rainfall, evaporation, sunshine and max wind speed to be the features that will predict the maximum price category. These features were selected because they improve our model’s performance, they are correlated with the class label, they are dependent of the class label and they are not correlated with the other features.

1. Separate dataset into 70% train and 30% test parts.
2. Train the model and predict the result with test data.

We’ve done k-fold cross validation with a variety of k-folds and tree depths, with accuracy scores as shown below. We have observed that the higher the k-fold value there were fewer sample in each fold and the accuracy score is better. We have chosen the k value with the highest accuracy score. With that, we’ve used k-fold cross validation with 20 k-folds and maximum depth of 12.

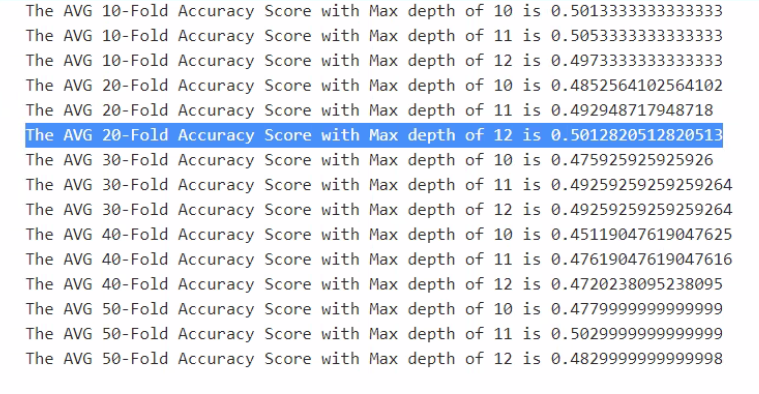


Figure 3: Sample accuracy scores for a variety of k-folds and max depths variation in our Decision Tree prediction model

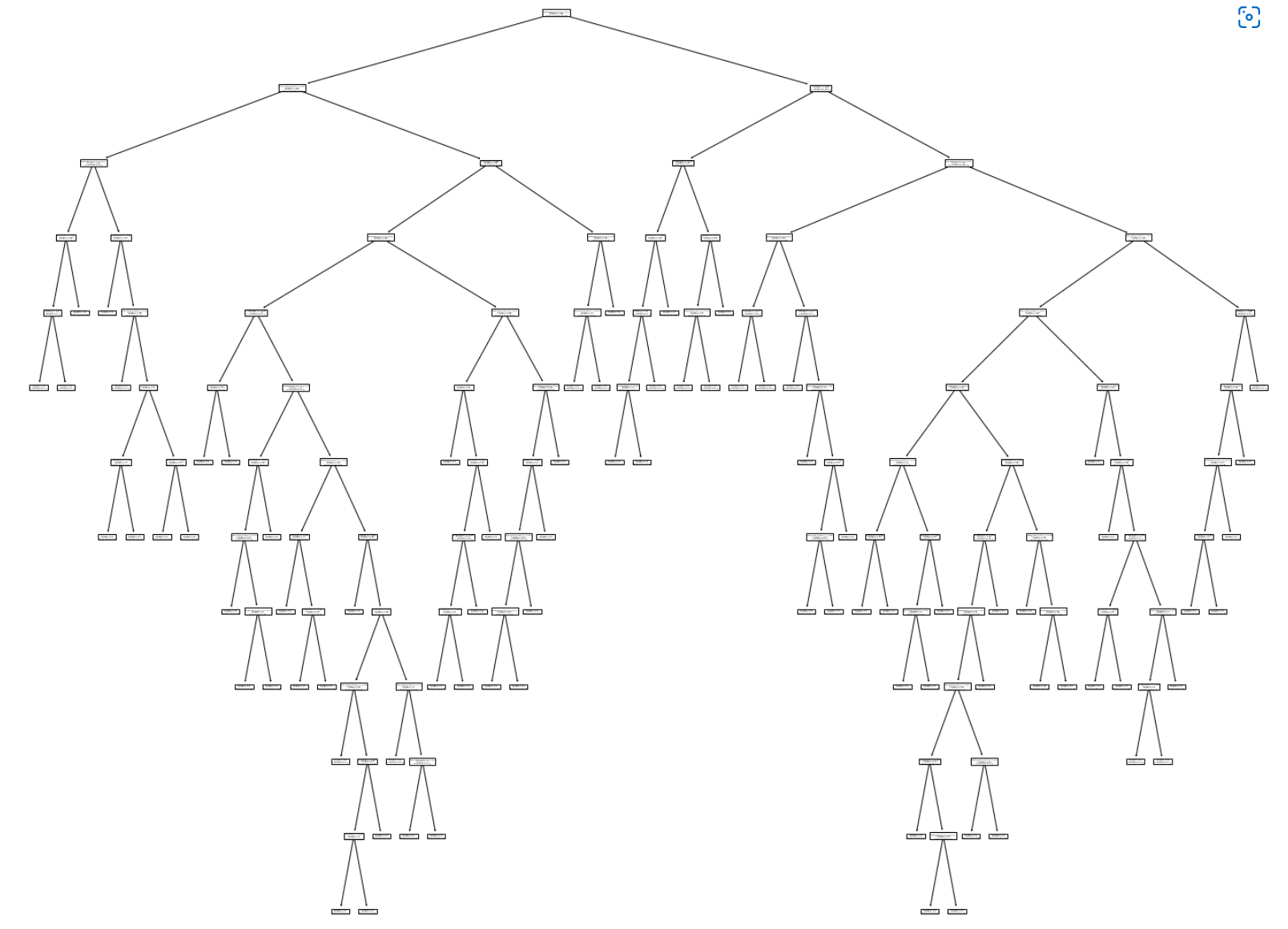


Figure 4: Decision Tree model with max depth of 12

1. Evaluate the result.

Based on the results of the Decision Tree algorithm, an accuracy score of approximately 50% indicates that our model can somewhat predict the maximum price category using the selected features. We’ve experimented by using chi-square test to check the independence of pairs of the features but this resulted to a lower accuracy score. Using different random states, tree depths, k-folds and train/test splits will result to different accuracy scores.

**3.4.2.2 K-nearest neighbour (KNN)**

In order to create the model using the KNN algorithm, we did the following:

1. Import the required libraries.
2. Load the combined data set.
3. Select the features.

We’ve chosen minimum temperature, rainfall, evaporation, sunshine and max wind speed to be the features that will predict the maximum price category. These features were selected because they improve our model’s performance, they are correlated with the class label, they are dependent of the class label and they are not correlated with the other features.

1. Separate dataset into 70% train and 30% test parts.
2. Train the model and predict the result with test data.

We’ve done k-fold cross validation with a variety of k-folds, with accuracy scores as shown below. We think that if k was too small it’s sensitive to noise points while if it’s too big the neighbourhood may include points from other classes, thus we decided to do 10-fold cross validation with a k value of 8. We have observed that the higher the k-fold value there were fewer sample in each fold and the accuracy score is better. We have chosen the k value with the highest accuracy score.

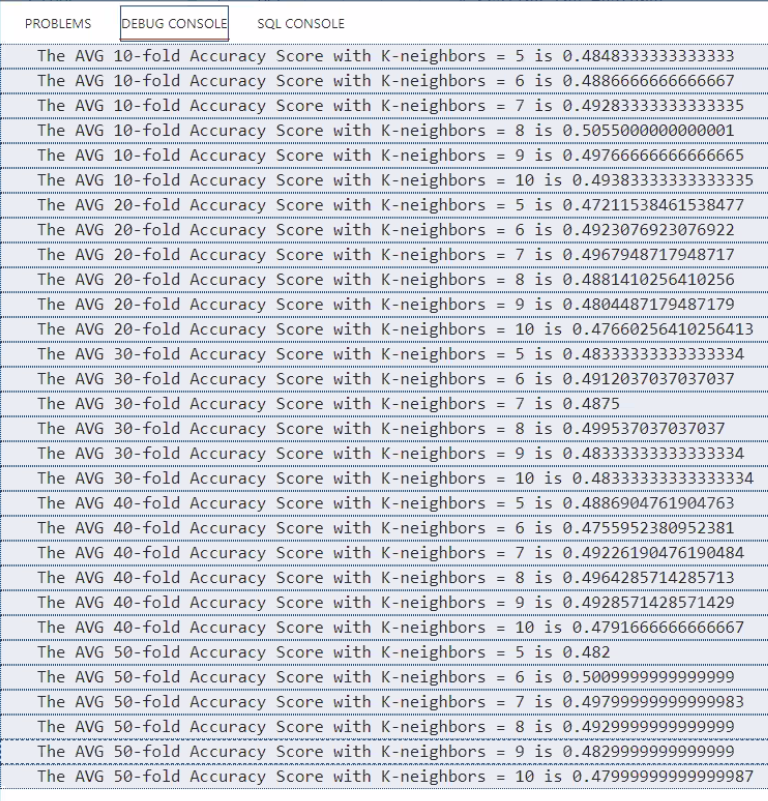


Figure 5: Sample accuracy scores for different k-folds and k-neighbours in our KNN prediction model

1. Evaluate the result.

Based on the results of the KNN algorithm, an accuracy score of approximately 50% indicates that our model can somewhat predict the maximum price category using the selected features, namely, minimum temperature, rainfall, evaporation, sunshine and max wind speed. We’ve experimented by using chi-square test to check the independence of pairs of the features but this resulted to a lower accuracy score. Using different random states, k neighbour values, k-folds and train/test splits will result to different accuracy scores.

**4. Discussion and Conclusion**

*What insights can you draw from your analysis? For example, which input variables are most valuable for predicting energy usage/price? Why are your results significant and valuable? What are the limitations of your results and how can the project be improved for future?*

Given the results of our maximum demand usage prediction model, we can confirm that the underlying relationship existing between the minimum temperature and the demand is somewhat linear. The r2 score for this model indicates that some but not all of the variation in the demand is explained by variation in the minimum temperature.

On the other hand, based on the results of our maximum price category prediction model, we therefore conclude that a change in the selected features will result in a change in the price category. Both Decision Tree and KNN produced similar results to validate this. Like the demand usage prediction model, it seems that the minimum temperature is the most valuable predictor of the price category.

The models can be improved by feature selection. Also, using different random states, k neighbour values, k-folds and train/test splits will result to different accuracy scores.

The results are significant insofar as they can help to predict the energy demand and price category for energy usage planning and costing. For a given weather forecast, we may be able to make some assumptions about energy demand and cost at a future date.

The models however cannot be used to predict outside of the range of weather data contained in the models. Also, the low accuracy scores may indicate that there are other factors beyond localized/regional weather data that affect energy demand and price, such as industry energy consumption or gas prices which fluctuate depending on supply constraints, transport prices and gas export demand.

The project could potentially be improved by using a larger dataset for modelling and perhaps incorporating other features outside of weather data that are found to correlate with energy demand and price. With larger datasets, other methods for data cleaning, feature selection, modelling and perhaps machine learning may become more practical.