Analysis of UK Weather and Weather's Impact on Happiness

Nabil Shadman

Abstract - In this paper, I analyse weather data of the United Kingdom using both unsupervised and supervised learning algorithms. Due to the natural variation of weather observed by the UK's weather stations at the tails of their distributions, some clusters of stations are identified. The variation of weather observed regionally enables the building of classification models to predict the region of a station with decent accuracies. Also, I investigate whether the weather in the UK has a linear relationship with happiness in the UK. The analysis demonstrates that the weather in the UK has a negligible effect on happiness in the UK. The entire workflow of this analysis is automated using Python and Bash, and the code and the data are made available for anyone interested in expanding on this work.

Date: April 20, 2021

Table of Contents

1. Introduction

In this paper, I analyse the weather in the United Kingdom (UK). Also, I investigate whether happiness in the UK is affected by the weather in the UK. I use two publicly available datasets for the analysis. The first dataset is the historic weather of the UK with monthly records of weather stations, made available by the Met Office of the UK[1]. Figure 1 below maps the weather stations. The second dataset is the personal well-being estimates of the UK from the Annual Population Survey (April 2014 to March 2015) with geographical breakdown, made available by the Office for National Statistics (ONS) of the UK[2]. Table 1 describes the features used from these datasets for the analysis.

Dataset	Feature	Definition
Historic station data	tmax	Mean daily maximum temperature (°C)
	tmin	Mean daily minimum temperature (°C)
	af	Days of air frost
	rain	Total rainfall (mm)
	sun	Total sunshine duration (hours)
Personal well-being estimates	average rating	Mean happiness rating on a scale of 0-10 where 0 is 'not at all happy' and 10 is 'completely happy'. The question asked in the survey was, 'Overall, how happy did you feel yesterday?'

Table 1: Features from the two publicly available datasets used in this analysis[1][2].

In Section 2 of this paper, I cluster the stations using the weather data. In Section 3, I build classification models based on the weather data to predict the region of a station. In Section 4, I investigate whether happiness in the UK is influenced by the weather in the UK. Finally, in Section 5, I describe the end-to-end automation of this project and how anyone interested in expanding on this work can easily rerun the code. As I follow the data science pipeline approach to perform this analysis, I recommend the reader to read the paper sequentially as the insights from one section are carried over to the following sections.

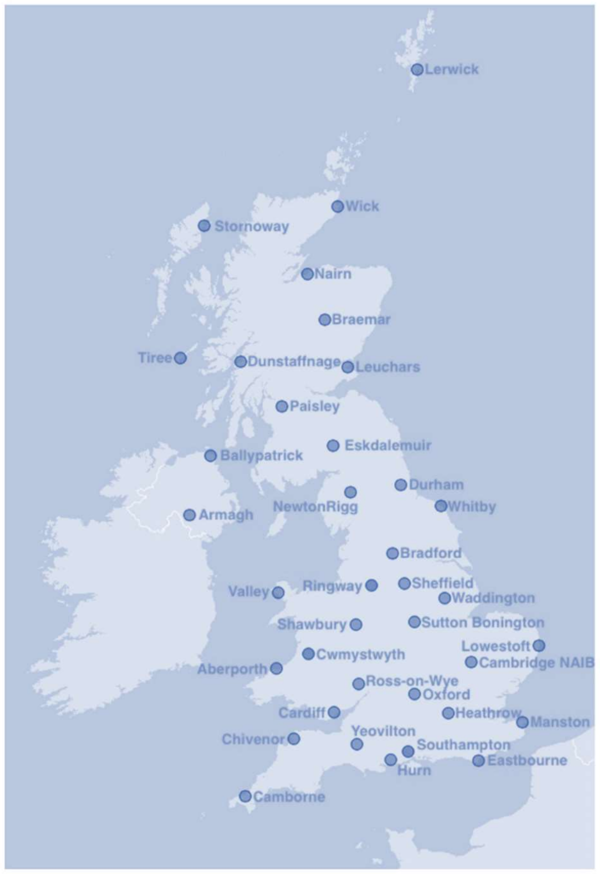


Figure 1: Map of weather stations in the UK.

2. Clustering

In this section, I cluster the weather stations into groups that have similar weather. Clustering is a type of unsupervised learning algorithm where the model groups a set of items so that each item is assigned to the same group as other items that have similarities with it.

In a data science project, datasets typically require a fair amount of cleaning before modelling. So, I think it is worth discussing how I cleaned and prepared the data before analysing the results of the clustering models. After I downloaded the data files, I noticed that each station file has the first few lines filled with some metadata (e.g., geographic coordinates of the station, technology used to record sunshine data). Also, some files have a 'Site Closed' message appended at the end of the file (e.g., Cwmystwyth, Ringway). As I am interested in loading the numeric values into a pandas dataframe later for analysis (pandas is an open-source library providing data structures and data analysis tools for Python, and dataframe is the primary data structure of pandas), I separate the data from the metadata in the original text files of the stations and copy them over into new text files[3]. I create new files instead of deleting the metadata from the same files because I want to keep the metadata for later analysis if needed. To see the implementation, please refer to the 'download_weather_data.ipynb' file in the supplementary 'code' folder.

After a brief inspection of the newly created text files, I notice that there are a few issues with the data. For instance, the missing data have been marked with "---". Several data points have been appended with special symbols such as '*', '#', or '$' to denote special information about those data points. Some datasets have special messages appended at the end of some lines with no spacing in between the record and the message (e.g., Lowestoft, Whitby).

To resolve these issues in the datasets, I loaded the datasets into a Python dictionary with the station as the key, and the data represented in a pandas dataframe as the value for each station. In this way, it is easy to loop through each dataset in the dictionary and clean it one after the other. I wrote an algorithm that checks for each of the issues mentioned above in the data and cleans them. For instance, the "---" is replaced with 'nan' (i.e., not a number), the special characters at the end of the data points are removed, and so on. For the implementation of the algorithm, please refer to the 'perform_clustering_weather_data.ipynb' file. I realise that this is a brute force algorithm, and it does take a few seconds to go through all the files and clean them. But I do this to prevent issues with messy data appearing anywhere in the future datasets, so that the program can be easily rerun with the monthly updated datasets.

Based on the dictionary of cleaned datasets, I then create two new pandas dataframes, one containing the historic averages of the weather data of each station, and the second containing the extreme values of the weather data of each station. When looking at the dataframe containing the historic averages of the weather stations, it does not seem particularly interesting as information at the tails of the distributions of the variables is lost due to averaging. Due to the large number of observations in each station and the natural seasonality in the data, the variation in the data (that would be of particular interest for clustering the stations) is reduced (Table 2). For instance, the mean of tmax from the average dataset is around 12.95 °C and the mean of tmin is around 6.11 °C. These would be the sort of temperatures we can expect to see around Spring (March-May) and Fall in the UK (September-November). The interquartile range of average tmax is only around 1.75 °C. With these low statistical dispersions in the dataset, it is not much informative to cluster the weather stations as they look quite similar. Therefore, I think it is better to consider the extreme values to cluster stations as including the information at the tails of the distributions are more informative (Table 3).

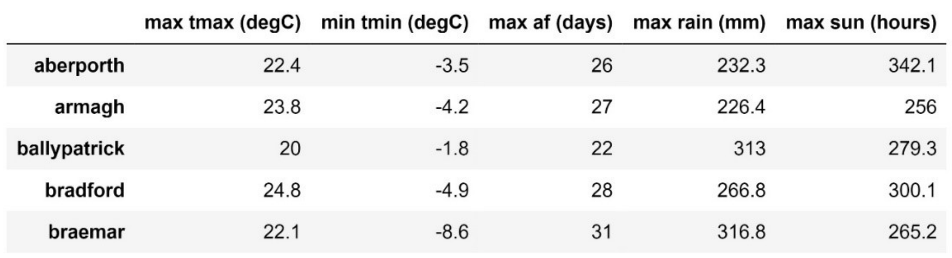
	avg tmax (degC)	avg tmin (degC)	avg af (days)	avg rain (mm)	avg sun (hours)
count	37.000000	37.000000	37.000000	37.000000	37.000000
mean	12.824506	6.019045	3.497373	75.278727	118.979958
std	1.333273	1.186896	1.585485	26.400515	15.727152
min	9.485701	2.749529	0.759295	46.328907	89.394557
25%	12.162115	5.319455	2.640063	54.802526	105.891299
50%	12.946581	6.115317	3.299679	67.892966	118.030108
75%	13.914859	6.762788	4.379971	90.132877	129.096523
max	14.920364	8.375342	8.687752	149.432712	155.036412

Table 2: Summary statistics of the weather dataset that includes averages (i.e., means) of the historic data of the stations.

When considering the dataset with the extreme weather records of the stations, we pick up the information at the tails of the distributions of variables (Table 3). The standard deviations of the variables are higher compared to those of Table 2. Assuming the climate of an area stays constant over a long period of time (say, centuries), this dataset informs us more about how hot it can get in a particular area, how cold it can get, how much rainfall it can receive, and so on. So, we know much more about the upper bound, or the lower bound (in case of tmin) of the weather variables. With tmin, it is sensible to look at the lower bound (i.e., minimum) as that is what the metric is intended for. Also, this dataset picks up more of the seasonal effects as the hottest months of the stations would be around summer, and the coldest months would be around winter. So, I picked the dataset with the extreme values for clustering (Table 4).

	max tmax (degC)	min tmin (degC)	max af (days)	max rain (mm)	max sun (hours)
count	37.000000	37.000000	37.000000	37.000000	37.000000
mean	23.881081	-4.251351	26.675676	261.086486	297.627027
std	2.733927	1.719596	2.708567	83.892418	30.748619
min	17.200000	-8.600000	19.000000	163.900000	249.100000
25%	22.100000	-5.500000	25.000000	193.600000	270.900000
50%	23.900000	-4.200000	27.000000	254.200000	297.500000
75%	26.000000	-2.900000	28.000000	285.600000	320.100000
max	28.300000	-0.800000	31.000000	568.800000	350.300000

Table 3: Summary statistics of the weather dataset that includes extreme values (i.e., max or min) of the historic data of the stations.



	max tmax (degC)	min tmin (degC)	max af (days)	max rain (mm)	max sun (hours)
aberporth	22.4	-3.5	26	232.3	342.1
armagh	23.8	-4.2	27	226.4	256
ballypatrick	20	-1.8	22	313	279.3
bradford	24.8	-4.9	28	266.8	300.1
braemar	22.1	-8.6	31	316.8	265.2

Table 4: The first five rows of the weather dataset with the extreme values of the stations.

When looking at the pair plot of the dataset with extreme values, three pairwise relationships particularly stood out to me (Figure 2). These are (tmax, tmin), (tmax, sun), and (tmax, rain). For example, the (tmax, tmin) plot could have two clusters. The first cluster consists of stations that have observed lower maximum temperatures (say, below 22 °C) and higher minimum temperatures (say, above -5 °C). The second cluster consists of stations that have observed higher maximum temperatures (say, above 22 °C) and lower minimum temperatures (say, below -5 °C). So, the second cluster consists of areas that have observed greater variation in weather compared to the areas in the other cluster.

In the (tmax, sun) plot, there could be 2 or 3 clusters. These would be clusters around how high the temperature can get and how long of sunshine the stations can observe. One cluster could be of stations with lower maximum temperatures and lower sunshine durations, while the other cluster could be of stations with higher maximum temperatures and higher sunshine durations. This plot also indicates a positive correlation between maximum temperature and sunshine duration. It makes intuitive sense as we tend to see more sunshine in hotter places.

The third noteworthy plot is that of (tmax, rain). We can see that there could be 2 or more clusters. There could be a cluster of stations that have observed higher maximum temperatures and lower rainfalls, whereas the other cluster could be of stations with lower maximum temperatures and higher rainfalls. This plot indicates a negative correlation between maximum temperature and maximum rainfall. It makes sense as we can expect to see lesser rainfall in hotter places, except for tropical places of the globe that can be very hot and can have very high rainfalls. Interestingly, extreme value analysis reveals that warmer atmosphere can hold more moisture, so there is a higher potential for more extreme rainfall events in future climates[4]. These events are probably less frequent because it takes longer for the moisture to build back up in the atmosphere. To analyse extreme weather events, one needs to examine the distributions of weather variables[5]. This is beyond the scope of this project as I choose to keep the analysis simple.

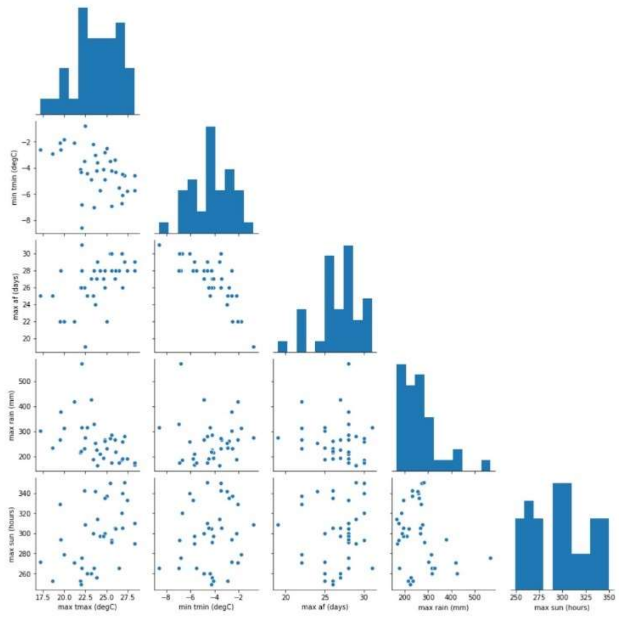
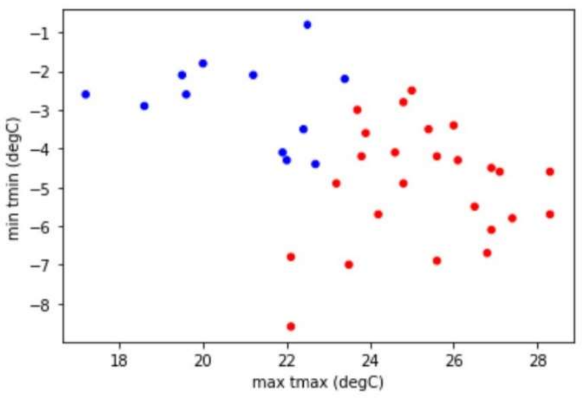


Figure 2: Pair plot of the weather stations dataset with extreme values.

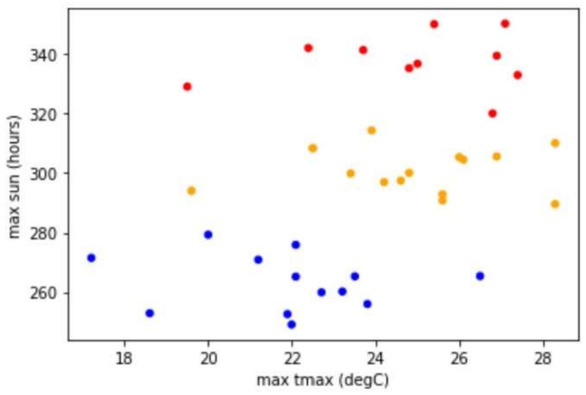
I applied the k-means algorithm to cluster the weather stations. I chose k-means as it is relatively simple to implement and easy to understand. The algorithm guarantees convergence and easily adapts to new examples[6]. However, one disadvantage of k-means is that it can be sensitive to outliers as the centroids can be dragged by outliers, or the outliers might get their own cluster. In this dataset, this should not be much of a problem as we do not have many outliers. To implement the algorithm, I used the KMeans class from scikit-learn (a machine learning library for Python)[7].

After experimenting with different numbers of clusters and visually inspecting the results, I chose the numbers of clusters (i.e., k parameters) of the three pairwise relationships (Figure 3). I chose two clusters for the k-means algorithm of tmax and tmin features (Figure 3a). I noticed that the stations can be clustered neatly around how hot and how cold they can get. The red dots represent a cluster of stations that have observed higher maximum temperatures and lower minimum temperatures with a centre of around (25.3 °C, -5.0 °C). The blue dots represent a cluster of stations with lower maximum temperatures and higher minimum temperatures with a centre of around (21.0 °C, -2.8 °C).

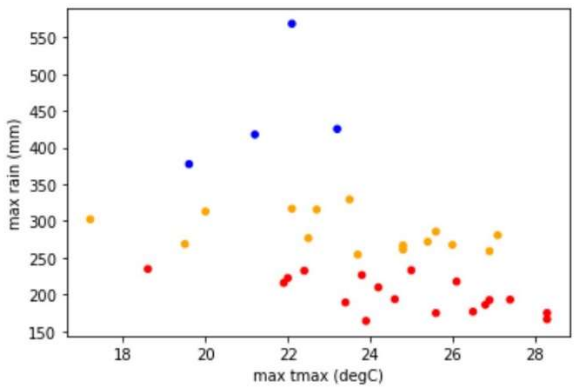
I chose three clusters (i.e., k = 3) for the clustering algorithm of tmax and sun features (Figure 3b). The blue cluster represents stations with relatively low maximum temperature and low sunshine duration (21.9 °C, 263 hours), the orange cluster represents stations with relatively high maximum temperature and medium sunshine duration (25.0 °C, 301 hours), and the third cluster represents stations with relatively high maximum temperature and high sunshine duration (25.0 °C, 338 hours).



(a) 2 clusters of stations using (tmax, tmin) variables



(b) 3 clusters of stations using (tmax, sun) variables



(c) 3 clusters of stations using (tmax, rain) variables

Figure 3: Plots of clusters of stations using extreme values of weather variables.

I also went with three clusters for the algorithm of tmax and rain features (Figure 3c). The blue cluster represents stations with relatively low maximum temperature and high rainfall (21.5 °C, 447 mm). The orange cluster represents stations with relatively medium maximum temperature and medium rainfall (23.5 °C, 285 mm). The red cluster represents stations with relatively high maximum temperature and low rainfall (24.8 °C, 200 mm). In this figure, we see one limitation of the k-means algorithm as the centroid for the blue cluster is pulled up in the rainfall direction (i.e., 447 mm) due to the outlier station that received the highest rainfall (568 mm), whereas other stations in the same cluster received maximum rainfalls around 400 mm.

So, the question is what could be forming these different clusters? My first guess would be that the geographic locations of the stations are forming these different clusters. In Figure 1, we observed that the stations are well spread out throughout the UK, and different regions in the country could observe different weather patterns. However, we cannot confirm this yet by running clustering algorithms. Let us proceed to the next section to find out more. For all the code and visualisations in this section, please refer to the 'perform_clustering_weather_data.ipynb' file in the supplementary folder.

3. Classification

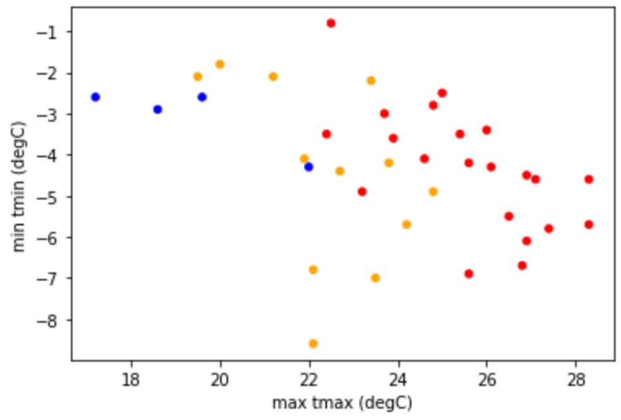
In this section, I discuss classification models built from the weather dataset. Classification is a type of supervised learning where the algorithm finds classes in which to place its inputs. A classification algorithm uses labelled dataset to make predictions. This contrasts with clustering, where the algorithm does not use a labelled dataset to look for patterns in the data.

I divided the weather stations into three regions based on their latitudes. The UK's most northern point has latitude of 60.9°, and the most southern point has latitude of 49.9°. The latitudes have been divided into three bands, and each station has been assigned to a region based on which band of latitudes the station falls into (Table 5). The latitude datapoint of each station is in the original text data file of the station. I fetched all the latitude data from the files and joined them into the dataframe containing extreme values. I again consider the dataset with extreme values of the stations because I assume that the classification models will be more accurate if there is more dispersion in the dataset.

Region	Latitude Range
Southern	49.900° - 53.566°
Central	53.567° - 57.232°
Northern	57.233° - 60.900°

Table 5: Bands of latitudes used to divide the UK weather stations into three regions.

I focus on the features that stood out to me in Section 2's exploratory data analysis: max tmax, min tmin, max sun, and max rain. For instance, we can visualise the regional class labels of the stations with different colours on a plot of max tmax and min tmin (Figure 4). For the plots of other sets of variables, please refer to the file 'perform_classification_weather_data.ipynb'. The cluster (in Section 2) of stations with lower tmax and higher tmin happens to be a combination of northern and central stations. The cluster of stations with higher tmax and lower tmin happens to be mostly southern stations. This is informative in the sense that the geographic regions do see different weather characteristics at the tails of distributions, with the southern region observing more variation in weather compared to that observed by the northern region.



[blue: northern, orange: central, red: southern]

Figure 4: Plot of the weather data using (tmax, tmin) variables, colour coded for different regions.

I built three classification models using the k-nearest neighbours (kNN) algorithm. I chose the kNN algorithm for its simplicity and speed. The algorithm is simple to implement, easy to understand, and fast[8]. The first kNN model is based on (max tmax, min tmin) features. The second model is based on (max tmax, max sun) features, and the third model is based on all features of interest, i.e. (max tmax, min tmin, max sun, max rain). For the implementation of the models, I used the KNeighborsClassifier class from scikit-learn[9]. To see the implementation, please refer to 'perform_classification_weather_data.ipynb' in the supplementary folder.

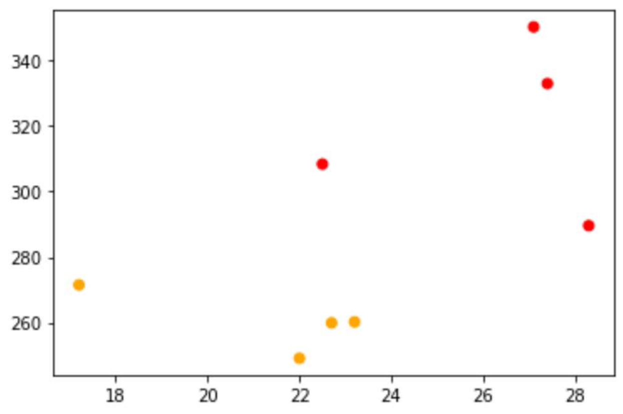
I used an 80-20 split between training and test sets. By varying the parameters of k and distance metric, I ran several experiments running each model multiple times. Table 6 shows results from an experiment where each model was run 10 times. I noticed that both model 1 and model 2 achieve quite close average accuracies, ranging from 0.500 to 0.875. Model 3 has the lowest average accuracy, indicating possible overfitting due to having more features in the model. I experimented with varying numbers of k (from 2-4) and distance metrics (e.g., Euclidean and Minkowski), even attempting to keep the parameters constant across all models. These parameters do not seem to make a difference in the accuracies of the models. I observed similar results with different choices of k and distance metric.

model	features	k	distance metric	average accuracy	min accuracy	max accuracy
1	max tmax, min tmin	3	euclidean	0.700	0.500	0.875
2	max tmax, max sun	2	minkowski	0.688	0.500	0.875
3	max tmax, min tmin, max sun, max rain	2	euclidean	0.438	0.250	0.750

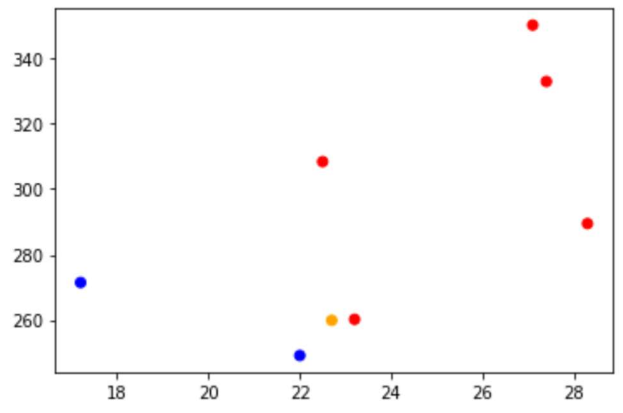
Table 6: Model evaluation of the three classification models.

Overall, I noticed that the models perform well when the test set includes mostly central and southern stations. The models perform poorer when there are northern stations in the test set and seem to confuse those with central stations due to proximity in their weather data. The poorer performance is due to the training set having only a few northern stations (i.e., the lowest among the three regions). So, even with a small number of k, the northern stations get voted out by the central stations in the classification algorithms. Figure 5 shows the plots from a run of model 2 where it achieved an accuracy score of 0.625. To view the plots of other models and to experiment with the models and their parameters, please see 'perform_classification_weather_data.ipynb'.

I have two recommendations for anyone who is interested in expanding on this work. Firstly, I recommend using a Receiver operating characteristic (ROC) curve for selecting optimal parameters of the classification algorithms. I tuned the parameters heuristically by running the algorithms several times, and visually inspecting the results. Perhaps, one may find more accurate results by using a more rigorous method such as plotting the ROC curve. Secondly, I recommend applying other models such as Convolutional Neural Networks (CNNs) that may achieve higher accuracies in classifying the regions of weather stations.



(a) predicted class labels.



(b) actual class labels.

[x: max tmax (degC), y: max sun (hours)]

[blue: northern, orange: central, red: southern]

Figure 5: Plots of predicted class labels and actual class labels from a run of the kNN algorithm.

4. Weather and Happiness

In this section, I investigate whether happiness is affected by weather. The happiness dataset has data at granular levels such as the local government level, and the higher levels include region. There is not a one-to-one mapping of a weather station in the weather dataset to a census area in the happiness dataset. But we can use the regional level happiness rating in the happiness dataset as a proxy for the happiness rating of the area around a weather station.

I did the mapping by assigning a region of the happiness dataset to a weather station that is closest from the station. Latitude and longitude data are available for both the weather stations (in the original text data files) and the regions of the happiness dataset (in a file called 'regions.txt' in the supplementary 'code' folder). I computed the Euclidean distances from a station to all regions in the dataset and assigned the region that has the minimum distance to the station. There are possibly better methods to do the mapping, but I found that the heuristic works well by inspecting the mapping visually in Google Maps.

For the features, I select certain percentiles of the weather variables this time. Except for tmin (for which I select the 10th percentile of historic records of the stations), the other features are the 90th percentiles of the historic records of tmax, af, rain, and sun of the stations. For both clustering and classification of the weather stations, the extreme values were sensible as the dissimilar weather of the stations at their extreme ends are easier for the models to learn. But they are indeed the upper bounds and lower bounds of the weather (at least for now), and the probabilities of their events are naturally very low. As we are interested in studying the relationship between weather and happiness, it is sensible to consider weather that is more common. For instance, the 90th percentile tmax is a maximum temperature that an area can expect to see more often in the summer. There can be hotter days than this, but the 90th percentile tmax is probably a typical maximum temperature in hottest months of the year. Similarly, a 10th percentile tmin is a typical minimum temperature in the coldest months of the year.

By examining the pair plot (Figure 6), I observed some interesting features. It seems that except for a few stations with high happiness ratings, happiness rating has a positive correlation with tmax, tmin, and sun. Although weak, happiness rating seems to have a negative correlation with air frost and rainfall. But the pairwise correlation matrix do not capture these insights as the stations with high happiness ratings bias the correlations. For this pairwise correlation matrix, please refer to 'perform_regression_weather_happiness_datasets.ipynb'.

As I am interested in analysing the impact of weather on happiness, these datapoints with high happiness ratings are not particularly helpful. It could be that these stations have other factors that influence their happiness ratings that are beyond the scope of this project. So, I excluded these datapoints from the dataset, and recalculated the pairwise correlation matrix (Table 7). These outlier stations with high happiness ratings happen to be in Northern Ireland. The updated correlation matrix below captures the insights that we discussed using the pair plot. We can see that the 90th percentile tmax, for instance, has a positive correlation with average rating.

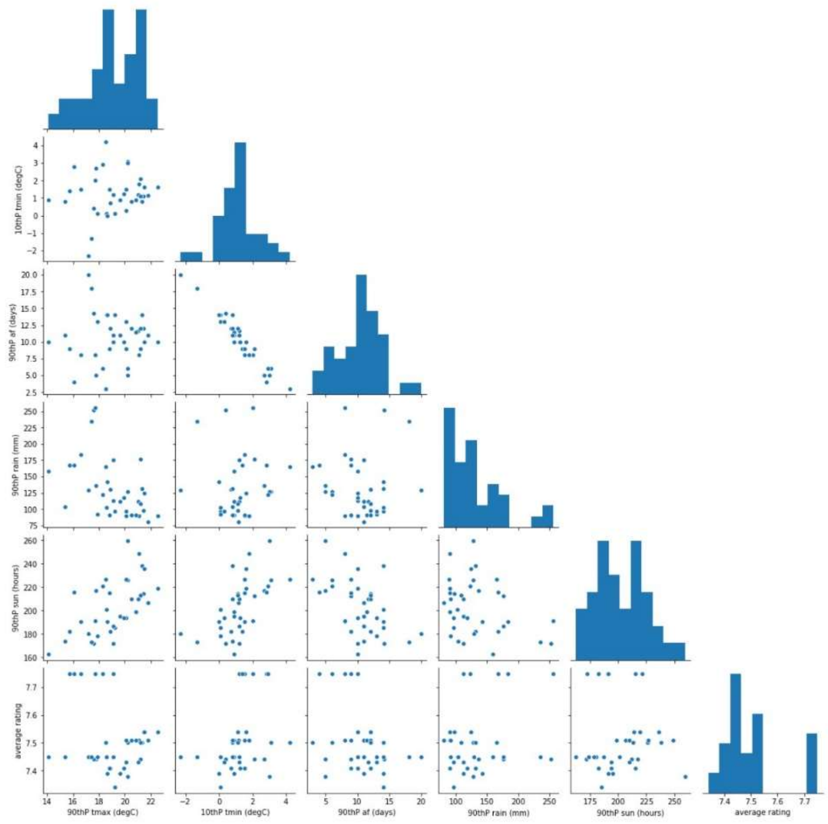


Figure 6: Pair plot of variables from the joined weather and happiness datasets.

	90thP tmax (degC)	10thP tmin (degC)	90thP af (days)	90thP rain (mm)	90thP sun (hours)	lat	long	average rating
90thP tmax (degC)	1.000000	0.303590	-0.178186	-0.405350	0.673889	-0.799217	0.405075	0.421474
10thP tmin (degC)	0.303590	1.000000	-0.970925	-0.085737	0.658919	-0.591129	0.028014	0.221393
90thP af (days)	-0.178186	-0.970925	1.000000	0.148071	-0.600738	0.456633	-0.084762	-0.144911
90thP rain (mm)	-0.405350	-0.085737	0.148071	1.000000	-0.327574	0.059175	-0.526147	-0.186299
90thP sun (hours)	0.673889	0.658919	-0.600738	-0.327574	1.000000	-0.779446	0.401738	0.380702
lat	-0.799217	-0.591129	0.456633	0.059175	-0.779446	1.000000	-0.172923	-0.357342
long	0.405075	0.028014	-0.084762	-0.526147	0.401738	-0.172923	1.000000	0.198961
average rating	0.421474	0.221393	-0.144911	-0.186299	0.380702	-0.357342	0.198961	1.000000

Table 7: Updated pairwise correlation matrix after removing the outliers.

Using the results from the pair plot and the pairwise correlation matrix, I selected three features to analyse further. The features are 90th percentile maximum temperature, 90th percentile sunshine duration, and 10th percentile minimum temperature. However, it is to be noted that I am interested in exposing a possible linear relationship between weather and happiness. For a linear relationship, considering correlation in feature selection is perhaps reasonable as correlation is designed to measure the degree of linear relationship between two variables. But correlation does not work in the presence of nonlinearities and should not be considered in that case[10].

I used linear regression to determine if happiness is affected by weather. I applied linear regression as the algorithm is simple to implement, and it is easy to interpret the results. However, the downside of linear regression is that it assumes a linear relationship between dependent and independent variables, which can possibly over-simplify of the model.

I used the LinearRegression class from scikit-learn and built three linear regression models from the selected features[11]. The first model uses only the 90th percentile maximum temperature feature. The second model uses only the 90th percentile sunshine duration feature. The third model uses 90th percentile maximum temperature and 10th percentile minimum temperature features. I avoided building models with features that are quite correlated (e.g., 90th percentile maximum temperature and 90th percentile sunshine duration) to avoid multicollinearity as statistical inferences from a model with multicollinearity may not be reliable[12]. Table 8 shows the summary of the models.

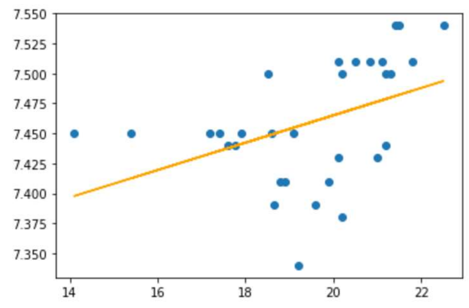
The first model has an R-squared of around 0.178. This means that around 17.8% of the variance in average rating is explained by variance in the 90th percentile tmax. The coefficient of 90th percentile tmax is around 0.011. This means that for each °C increase in 90th percentile tmax, average rating increases by around 0.01. So, for a 10 °C increase in 90th percentile tmax, average rating is expected to increase by around 0.11.

model	features	coefficient(s)	intercept	mean squared error	R-squared
1	90thP tmax	0.011416	7.236788	0.002119	0.177640
2	90thP sun	0.000822	7.290785	0.002203	0.144934
3	90thP tmax, 10thP tmin	0.010570, 0.004321	7.248623	0.002094	0.187257

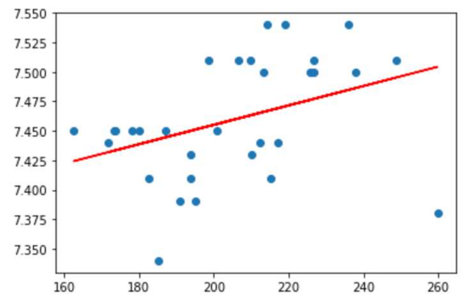
Table 8: Model evaluation of the three linear regression models.

The second model has resulted in slightly lower R-squared and slightly higher mean squared error compared to those of the first model. The coefficient of 0.000822 means that for every hour increase in the 90th percentile sunshine duration, happiness rating increases by around 0.0008. So, if an area receives 100 hours more of sunshine, happiness rating is expected to increase by 0.08.

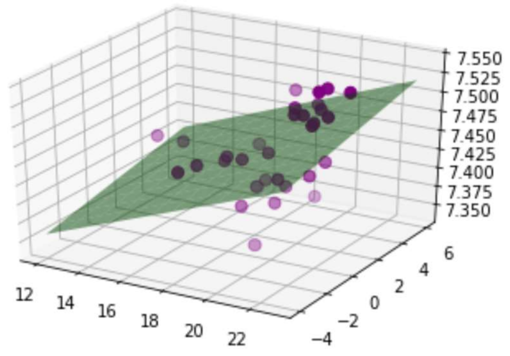
For the third model, I added the 10th percentile tmin feature to the first model. We see that the evaluation metrics improve marginally, suggesting that tmin is not a good predictor of happiness rating. The lower coefficient of 10th percentile tmin compared to that of the 90th percentile tmax also indicates the same. The positive coefficient of tmin suggests that warmer temperatures increase happiness.



(a) scatterplot and linear fit of (tmax, average rating)



(b) scatterplot and linear fit of (sun, average rating)



(c) surface plot and plane of best fit of (tmax, tmin, average rating)

Figure 7: Plots of the weather and happiness data and the linear regression models.

Overall, I conclude that weather does not have a significant linear effect on happiness. As we have seen in Table 7, the coefficients of all models are very low. The stations in our dataset (after removing the outliers) have happiness ratings ranging from around 7.34 to 7.54 (Figure 7). So, the average rating does not vary much between these stations anyway. Also, the intercepts of all models start at high figures (e.g., 7.24 in model 1). So, there are other factors that have stronger effects on happiness, which are not explained by these models. Moreover, the slight associations observed between weather and happiness could very well be random. The happiness data is only from 2014-15. So, there could be factors around that time influencing happiness.

To perform a more rigorous investigation of weather’s impact on happiness, a time series analysis can be conducted mapping weather data to happiness data over a long period of time (say, 1000 years). Of course, this kind of study would not be feasible currently as we do not have the data available for both weather and happiness rating dating back several years. Also, I attempted to expose a linear relationship between weather and happiness in this project. I recommend investigating whether there is a nonlinear relationship between weather and happiness. To see all the code, graphs, and model statistics of this section, please refer to ‘perform_regression_weather_happiness_datasets.ipynb’ in the ‘code’ folder.

5. Automation

In this section, I describe the end-to-end automation of this data science project. The entire workflow from downloading data (Section 1) to running regression models (Section 4) has been automated using a combination of Python and Bash. The whole process could be rerun easily with opportunities to change parameters at certain places. As I have taken a data science pipeline approach in this project, some intermediate datasets that can be reused in later stages of the pipeline are saved to disk for reducing computational cost.

Figure 8 illustrates the entire project’s workflow and the relationships of the Jupyter notebooks with the intermediate files created in the process. Here are the steps below to run the process end-to-end.

(1) Open the ‘code’ folder, and open the folder inside named ‘test_automation’. The folder has five Jupyter notebooks and two text files.

(2) Download the weather data files by running the 'download_weather_data.ipynb' notebook. The program fetches data from the alterable 'stations.txt' file to determine the subset of stations it needs to download the data for. The program will create a new folder called 'weather_data' and will store the original data files in there. After all the files have been downloaded, the script creates new text files where the data from the original files are copied over without the metadata at the top and at the bottom of the original files.

(3) Download the happiness data by running 'download_happiness_data.ipynb' notebook. This will create a new folder named 'happiness_data' and will store the Excel file of personal well-being estimates dataset in the folder.

(4) Perform clustering of weather data by running 'perform_clustering_weather_data.ipynb' notebook. This script fetches data from the 'stations.txt' file, and from the text data files (with removed metadata) in the 'weather_data' folder. The program cleans the data and stores the cleaned pandas dataframes in a dictionary, which is saved to disk ('stations_dict.pkl') in the 'weather_data' folder for reuse in the regression stage. The dataset with extreme weather values is also saved in the 'weather_data' folder for reuse in the classification stage. The later parts of the program visualise data and run clustering algorithms.

(5) Perform classification by running 'perform_classification_weather_data.ipynb' notebook. The program loads the extreme values dataset that was saved to disk from the previous stage (i.e., 'stations_aggregates.xlsx'). The latitude data is fetched from the original text data files of the weather stations. The rest of the program visualises data, runs classification algorithms, and prints results.

(6) Perform regression of weather and happiness datasets by running 'perform_regression_weather_happiness_datasets.ipynb' notebook. This notebook loads the census regions from the 'regions.txt' file, happiness ratings data from the 'happiness_data' folder, cleaned weather stations data from the 'stations_dict.pkl' file (created in the clustering stage), and latitude and longitude data from the original text data files of stations. The data from all these files are joined to create the final dataset to be fed into the regression models. The rest of the program visualises data, computes summary statistics, runs linear regressions, and prints model evaluation metrics.

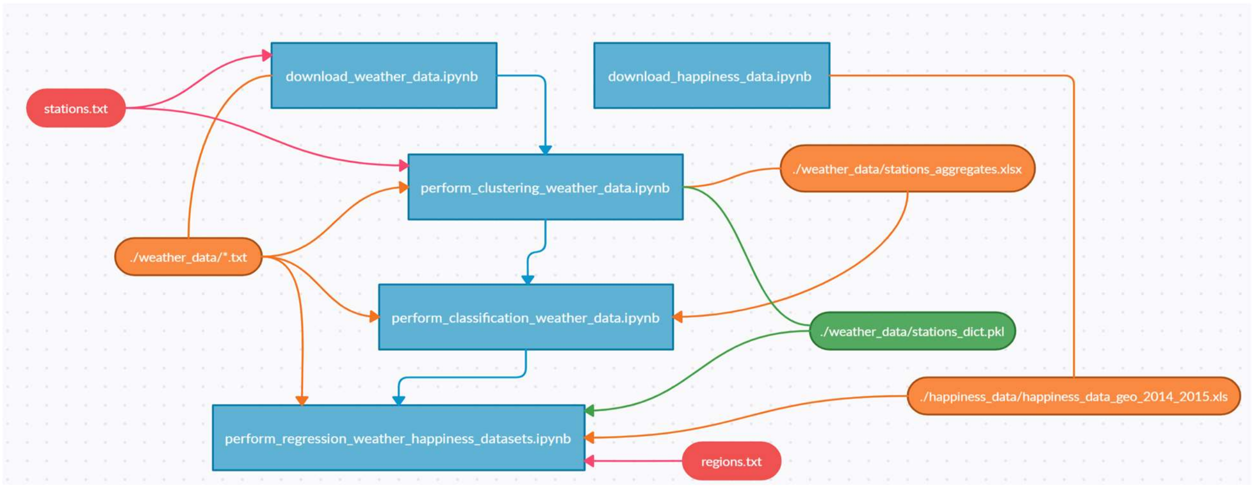

stations.txt
download_weather_data.ipynb
download_happiness_data.ipynb
perform_clustering_weather_data.ipynb
./weather_data/stations_aggregates.xlsx
./weather_data/*.txt
perform_classification_weather_data.ipynb
./weather_data/stations_dict.pkl
./happiness_data/happiness_data_geo_2014_2015.xls
perform_regression_weather_happiness_datasets.ipynb
regions.txt

Figure 8: End-to-end project workflow including relationships of notebooks with intermediate files.

6. Conclusion

The goal of this paper was to analyse the weather in the UK. The UK weather stations have observed different characteristics at the tails of the distributions of weather variables. This enabled us to form some clusters of stations with similar weather patterns, for instance, a cluster of stations that have observed more variation in temperatures compared to another cluster of stations that have seen lesser variation in temperatures. We have seen that one aspect of the formation of natural clusters is around the regions of the UK, which observed different weather patterns. This regional variation in weather enabled us to build classification models to predict regions of stations with decent accuracies. Another aspect of the paper was to determine if weather affects happiness in the UK. We have seen that weather has a negligible effect on happiness when considering a linear relationship.

For anyone interested in expanding on this work, all the code and data are made available in the supplementary 'code' folder. The entire workflow is automated end-to-end for enabling anyone to easily rerun the code with updated weather data or happiness data, or to change certain parameters to test the effects.

References

[1] Met Office, 2021. Historic station data. [online] Met Office. Available at: <https://www.metoffice.gov.uk/research/climate/maps-and-data/historic-station-data> [Accessed 18 April 2021].

[2] Office for National Statistics, "Dataset: Personal well-being estimates geographical breakdown", *ons.gov.uk*, 2015. [Online]. Available: https://www.ons.gov.uk/peoplepopulationandcommunity/wellbeing/datasets/personalwellbeingestimatesgeographicalbreakdown. [Accessed: 18- Apr- 2021].

[3] pandas, "pandas.DataFrame", *pandas.pydata.org*. [Online]. Available: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html. [Accessed: 18- Apr- 2021].

[4] Met Office, n.d. *Extreme Value Analysis*. [online] Met Office. Available at: <https://www.metoffice.gov.uk/services/research-consulting/weather-climate-consultancy/extreme-value-analysis> [Accessed 18 April 2021].

[5] NOAA PSL, "Distributions of Daily Meteorological Variables: Background", *psl.noaa.gov*, 2021. [Online]. Available: https://psl.noaa.gov/data/writ/distributions/background/index.html. [Accessed: 18- Apr- 2021].

[6] Google Developers, "k-Means Advantages and Disadvantages", *Google Developers*. [Online]. Available: https://developers.google.com/machine-learning/clustering/algorithm/advantages-disadvantages. [Accessed: 18- Apr- 2021].

[7] scikit-learn, "sklearn.cluster.KMeans", *scikit-learn*, 2021. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html. [Accessed: 18- Apr- 2021].

[8] Elsevier B.V., "K Nearest Neighbor", *ScienceDirect*. [Online]. Available: https://www.sciencedirect.com/topics/biochemistry-genetics-and-molecular-biology/k-nearest-neighbor. [Accessed: 18- Apr- 2021].

[9] scikit-learn, "sklearn.neighbors.KNeighborsClassifier", *scikit-learn*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html. [Accessed: 18- Apr- 2021].

[10] N. Taleb, "Fooled by Correlation: Common Misinterpretations in Social Science", *Academia*, 2019. [Online]. Available: https://www.academia.edu/39797871/Fooled_by_Correlation_Common_Misinterpretations_in_Social_Science_. [Accessed: 18- Apr- 2021].

[11] scikit-learn, "sklearn.linear_model.LinearRegression", *scikit-learn*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html. [Accessed: 18- Apr- 2021].

[12] A. Hayes, "Multicollinearity", *Investopedia*, 2021. [Online]. Available: https://www.investopedia.com/terms/m/multicollinearity.asp. [Accessed: 18- Apr- 2021].