Weather Analysis and Forecasting using Machine Learning

20.04.2021

Report by:
Milind Soni and Sanskar Tewatia

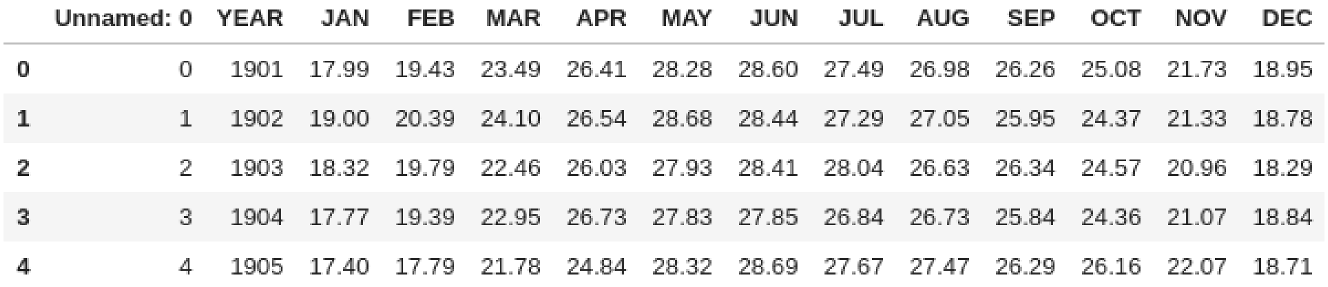
Contents

Abstract

Weather forecasting is a product of science that impacts the lives of many people. We have been recording the temperatures daily and it is important that we analyse it efficiently and see the trend to forecast the future weather conditions. Due to the applicability of machine learning in a variety of fields, it is of interest to study whether an artificial neural network can be a good candidate for prediction of weather conditions in combination with large data sets. The availability of meteorological data from multiple online sources is an advantage.

INTRODUCTION

We have chosen the publicly available dataset on https://data.gov.in/ which contains the mean monthly temperature in india from the year 1901 to 2017.



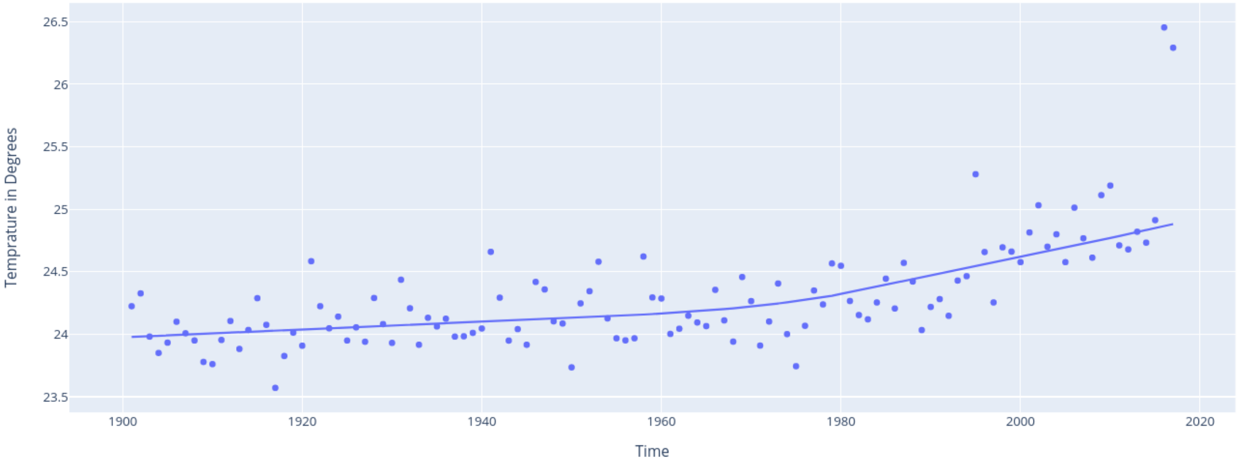
	Unnamed: 0	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
0	0	1901	17.99	19.43	23.49	26.41	28.28	28.60	27.49	26.98	26.26	25.08	21.73	18.95
1	1	1902	19.00	20.39	24.10	26.54	28.68	28.44	27.29	27.05	25.95	24.37	21.33	18.78
2	2	1903	18.32	19.79	22.46	26.03	27.93	28.41	28.04	26.63	26.34	24.57	20.96	18.29
3	3	1904	17.77	19.39	22.95	26.73	27.83	27.85	26.84	26.73	25.84	24.36	21.07	18.84
4	4	1905	17.40	17.79	21.78	24.84	28.32	28.69	27.67	27.47	26.29	26.16	22.07	18.71

In this project we will be :

1. Analysing the indian weather data which has been recorded for a hundred years.
2. Finding out trends in the data by exploratory data analysis and seeing the effect of global warming.
3. Future forecasting of data by using machine learning models.

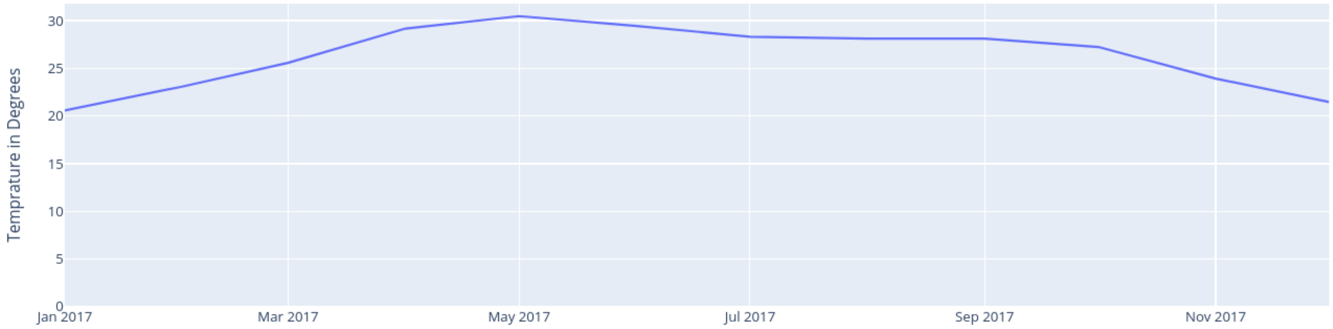
Exploratory Data Analysis

We will now start with the analysis and visualisation of the data given to us.

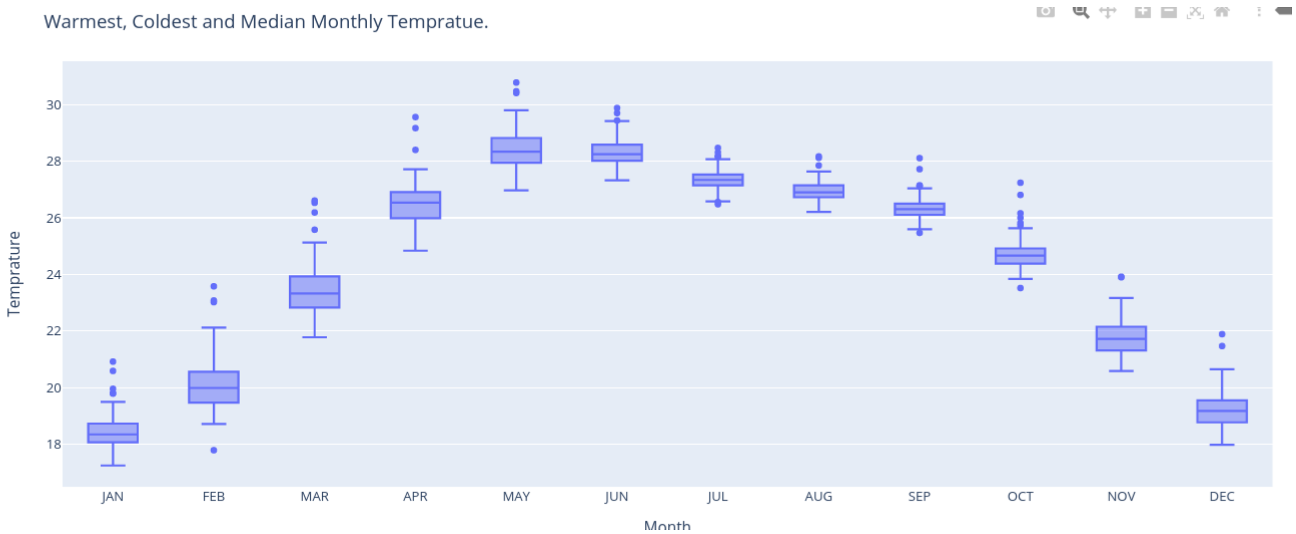


We can clearly see the trendline over the years and we can conclude that:

- The yearly mean temperature was not increasing much till 1980. It was only after 1979 that we can see the sudden increase in yearly mean temperature.
- We can very well see the effects of global warming and the resulting increasing temperature.
- After 2015, yearly temperature has increased alarmingly.



We can see the fluctuation in the mean temperatures throughout the year and shortly we will be establishing the effects of global warming using EDA.



Here we have shown the Warmest, Coldest and Median monthly Temperatures using boxplots.

Boxplot is a method for graphically depicting groups of numerical data through their quartiles.

In these boxplots we can compare the range and distribution of the mean temperatures, throughout the years.

We can conclude that -

1. January has the coldest Days in a Year.
2. May has the hottest days in an Year.
3. July is the month with least Standard Deviation.

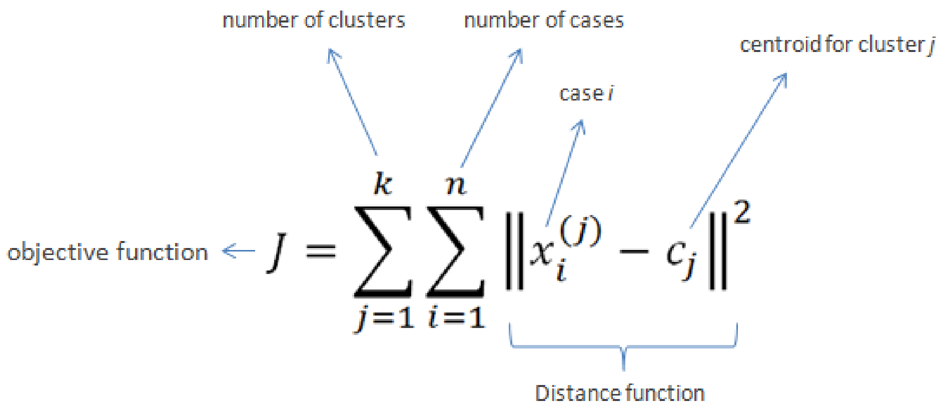
METHODOLOGY

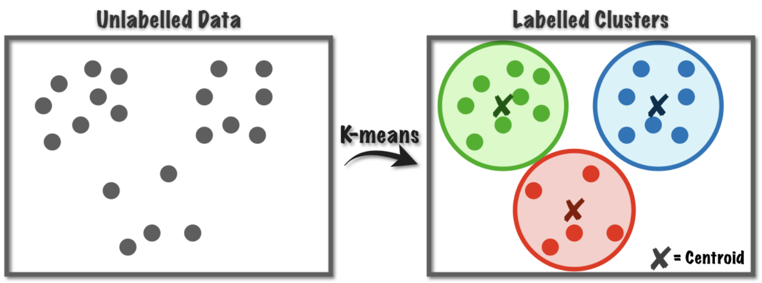
Initial Algorithm - K Means Clustering

“A cluster refers to a collection of data points aggregated together because of certain similarities.”

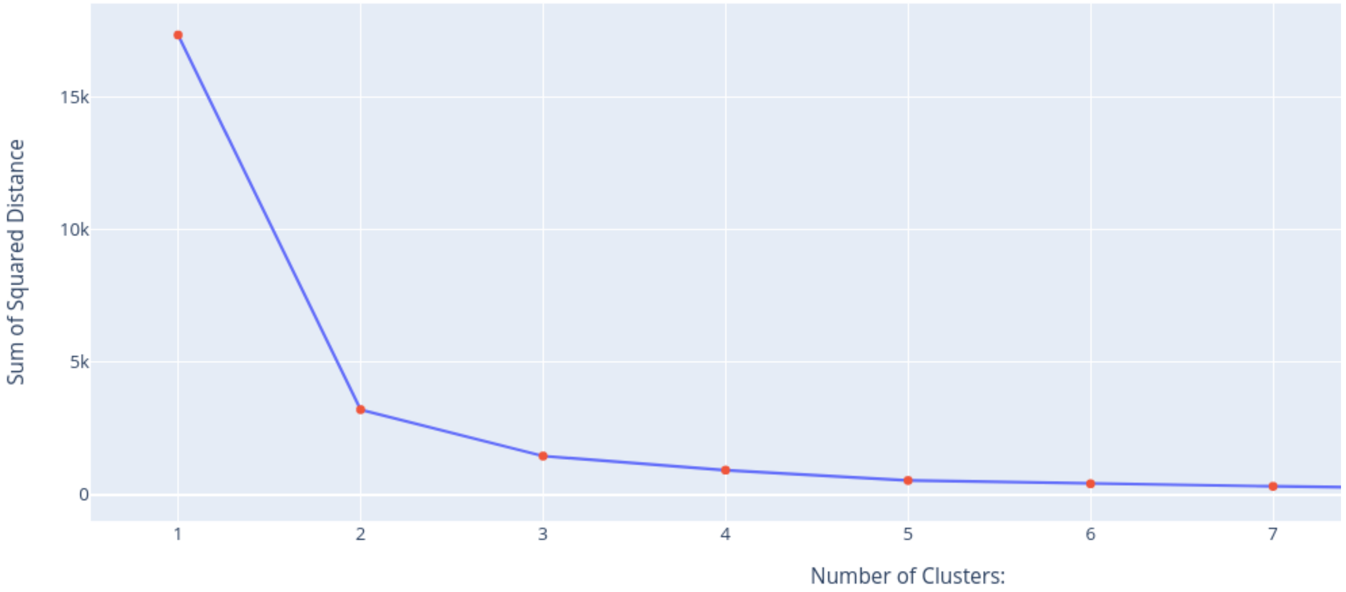
We will be using K-means clustering to visualise the seasons according to the mean temperatures in the dataset.

To process the learning data, the K-means algorithm in data analysis starts with a first group of randomly selected centroids, which are used as the beginning points for every cluster, and then performs iterative calculations, to optimize the positions of the centroids.

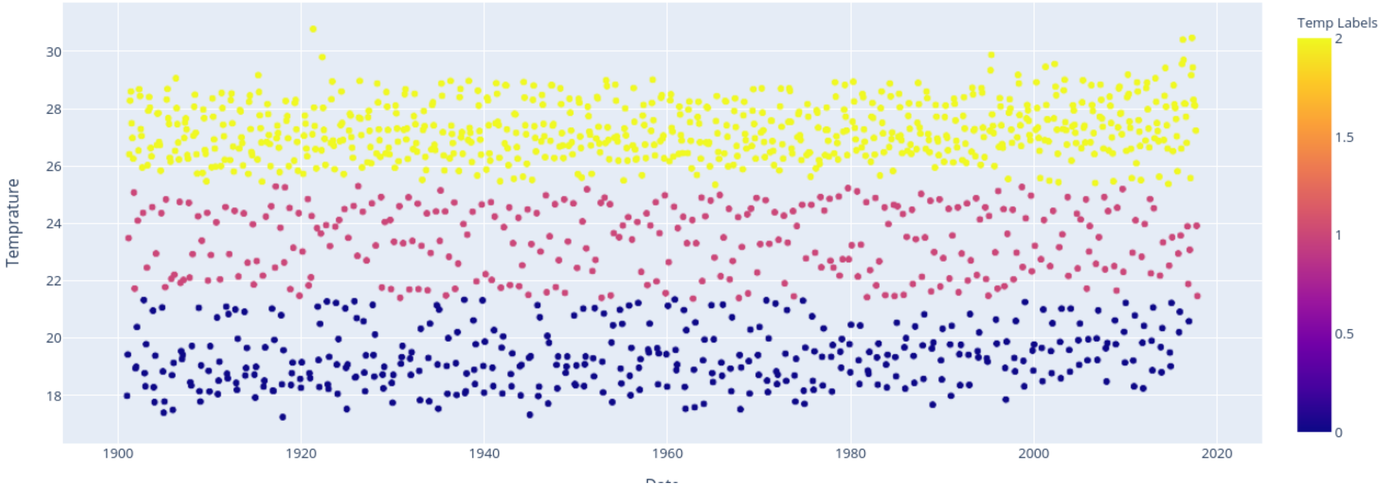




On evaluating the number of clusters suitable for this dataset, we find that **3** would be a suitable number of clusters.



We will be using the libraries to perform this function for us.



In the plot we are clearly able to see the 3 main clusters based on the temperatures.

Forecasting using Machine Learning Algorithms

We will be moving on to some machine learning regression algorithms which would help us to forecast the average mean temperatures based on the dataset.

1) Linear Regression

It is a commonly used algorithm and can be imported from the Linear Regression class. A single input variable(the significant one) is used to predict one or more output variables, assuming that the input variable isn't correlated with each other. It is represented as :

$$Y_i = \alpha + \beta_i \times x_i + \varepsilon_i$$

Here α,β_i are the coefficients/weights or parameters of the regression and ε_i is the error term. The values of these coefficients are found using Ordinary Least Squares method. This means that given a regression line through the data, one calculates the distance from each data point to the regression line, squares it, and sums all of the squared errors together. This is the quantity that ordinary least squares seeks to minimize, also known as the cost function -

$$\sum_{i=1}^{M}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{M}\left(y_i - \sum_{j=0}^{p} w_j \times x_{ij}\right)^2$$

Cost function for simple linear model

$$\hat{\alpha} = \min_{\alpha}\sum_{i=1}^{n}(y_i - \alpha - \beta x_i)^2 = \min_{\alpha}\sum_{i=1}^{n}\varepsilon_i^2$$

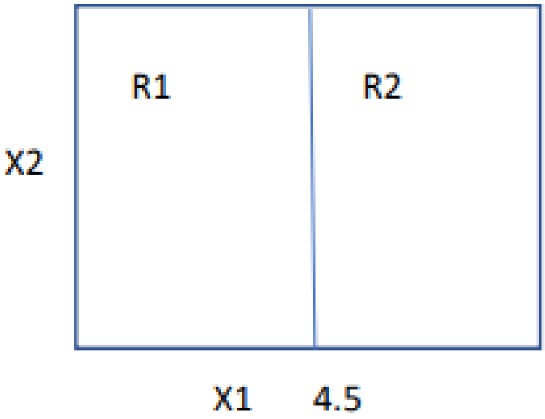
$$\hat{\beta} = \min_{\beta}\sum_{i=1}^{n}(y_i - \alpha - \beta x_i)^2 = \min_{\beta}\sum_{i=1}^{n}\varepsilon_i^2$$

The most optimal values are taken when the error term is minimized.

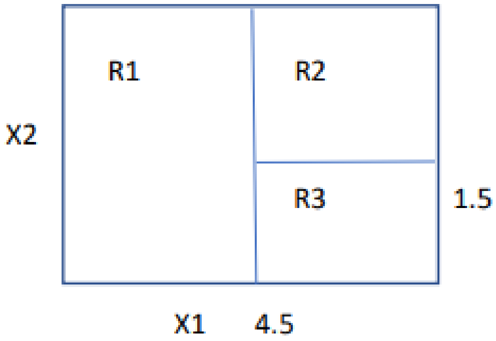
2) Decision Tree Algorithm

The Decision-tree algorithm falls under the category of supervised learning algorithms, since it requires well-labelled classes. It consists of decision nodes, interior and leaf nodes.

Let $X1$, $X2$,… , Xp be the predictors and Y be the real valued response. A decision tree is built by partitioning the feature space generated by $X1$, $X2$,… , Xp into rectangles (or boxes) $R1$, $R2$, … , RJ .



The descriptions of the boxes are as follows: $R1 = \{(x1, x2): x1 \leq 4.5\}$ and $R2 = \{(x1, x2): x1 > 4.5\}$



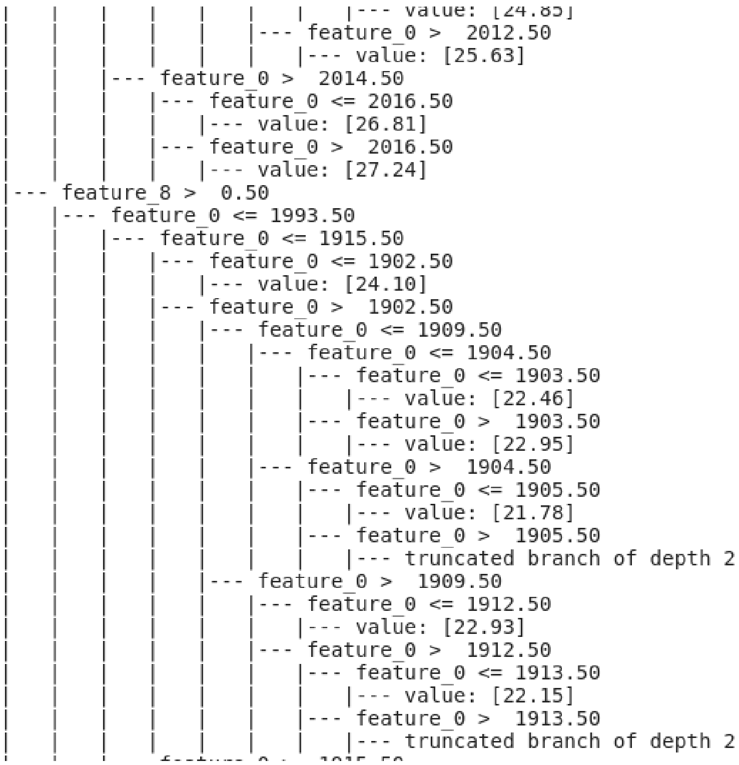
The above partitioning forces the underlying tree to give a fixed value *ŷRj* for all observations satisfying the cut-off conditions. We now look at the strategy for choosing these cut-off points.

We work with finitely many options for X1, X2, …., Xp. Then find the RSS corresponding to each cut (*Xi*, *si*):

$$\sum_{X_i \le s_i} \left(y_i - \hat{y}_{R_j}\right)^2 + \sum_{X_i > s_i} \left(y_i - \hat{y}_{R_j}\right)^2$$

We then repeat this process for each individual box.

In our dataset, we will be training a decision tree to produce an output, since this is a regression problem.



```
|   |   |   |   |   |   |   |--- value: [24.85]
|   |   |   |   |   |   |--- feature_0 >  2012.50
|   |   |   |   |   |   |   |--- value: [25.63]
|   |   |--- feature_0 >  2014.50
|   |   |   |--- feature_0 <= 2016.50
|   |   |   |   |--- value: [26.81]
|   |   |   |--- feature_0 >  2016.50
|   |   |   |   |--- value: [27.24]
|--- feature_8 >  0.50
|   |--- feature_0 <= 1993.50
|   |   |--- feature_0 <= 1915.50
|   |   |   |--- feature_0 <= 1902.50
|   |   |   |   |--- value: [24.10]
|   |   |   |--- feature_0 >  1902.50
|   |   |   |   |--- feature_0 <= 1909.50
|   |   |   |   |   |--- feature_0 <= 1904.50
|   |   |   |   |   |   |--- feature_0 <= 1903.50
|   |   |   |   |   |   |   |--- value: [22.46]
|   |   |   |   |   |   |--- feature_0 >  1903.50
|   |   |   |   |   |   |   |--- value: [22.95]
|   |   |   |   |   |--- feature_0 >  1904.50
|   |   |   |   |   |   |--- feature_0 <= 1905.50
|   |   |   |   |   |   |   |--- value: [21.78]
|   |   |   |   |   |   |--- feature_0 >  1905.50
|   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |--- feature_0 >  1909.50
|   |   |   |   |   |--- feature_0 <= 1912.50
|   |   |   |   |   |   |--- value: [22.93]
|   |   |   |   |   |--- feature_0 >  1912.50
|   |   |   |   |   |   |--- feature_0 <= 1913.50
|   |   |   |   |   |   |   |--- value: [22.15]
|   |   |   |   |   |   |--- feature_0 >  1913.50
|   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |--- feature_0 >  1915.50
```

A glimpse of the decision tree on our dataset

3) Ridge Regression

This is also an extension of linear regression but it performs L2 regularization, which adds a penalty to the cost function. That is, the cost function is altered by adding a penalty equivalent to square of the magnitude of the coefficients.

$$\sum_{i=1}^{M}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{M}\left(y_i - \sum_{j=0}^{p} w_j \times x_{ij}\right)^2 + \lambda \sum_{j=0}^{p} w_j^2$$

Cost function for ridge regression

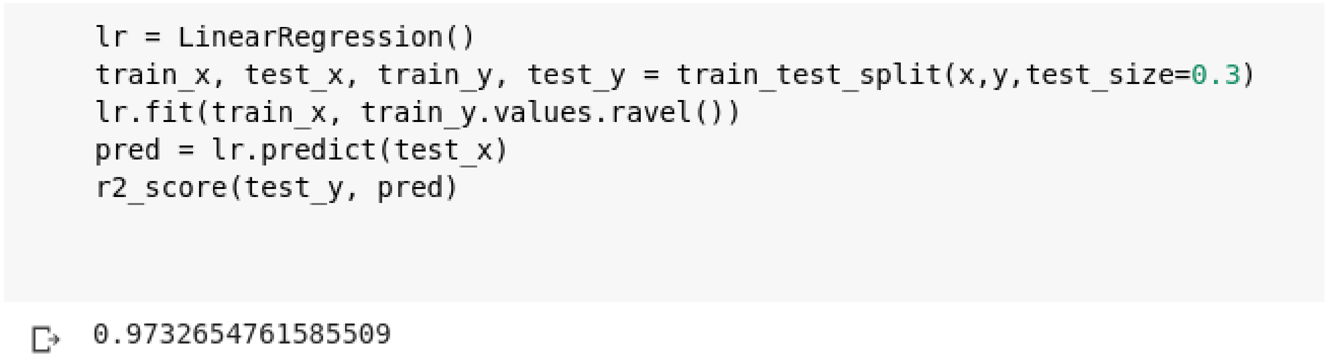
As the value of this variable Lambda is reduced, it tries to resemble more like the Linear regression algorithm. Ridge regression shrinks the coefficients and it helps to reduce the model complexity.

But for very large λ, the minimisation would force the coefficient of λ to shrink towards 0. This means that each w shrinks towards 0. So, if some parameters become 0, then the corresponding terms get dropped from the model which basically means variable selection.

Cross Validation is a very useful technique for assessing the effectiveness of your model, particularly in cases where you need to mitigate overfitting. It is also of use in determining the hyper parameters of your model, in the sense that which parameters will result in the lowest test error.

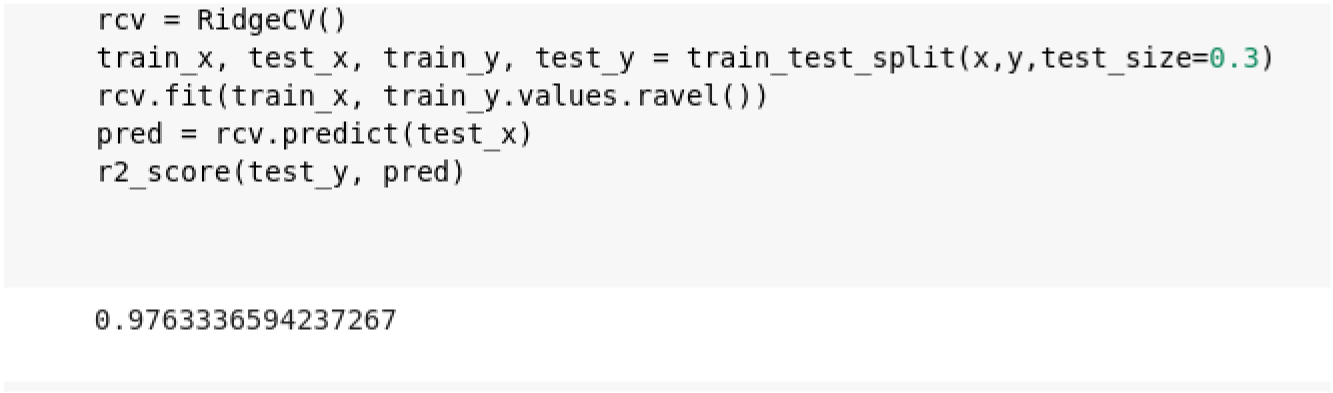
We will be using **RidgeCV** model which is ridge regression built in with cross validation.

RESULTS AND CONCLUSIONS



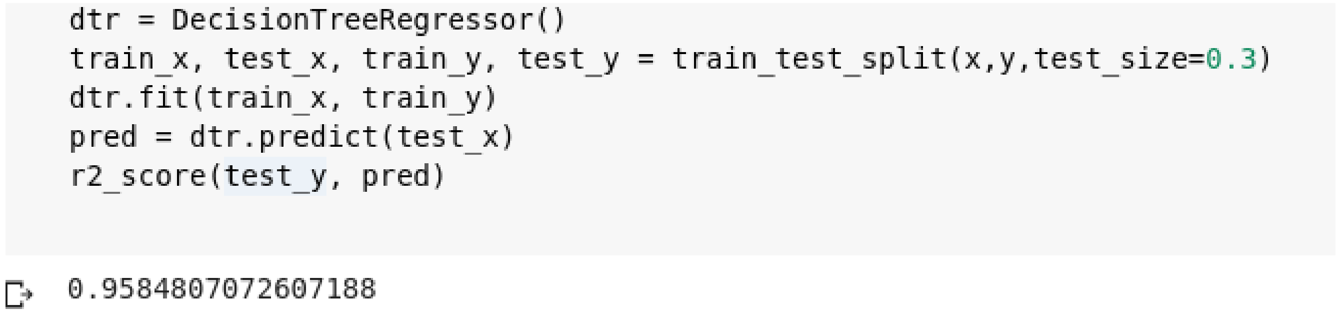
```
lr = LinearRegression()
train_x, test_x, train_y, test_y = train_test_split(x,y,test_size=0.3)
lr.fit(train_x, train_y.values.ravel())
pred = lr.predict(test_x)
r2_score(test_y, pred)
```

```
0.9732654761585509
```



```
rcv = RidgeCV()
train_x, test_x, train_y, test_y = train_test_split(x,y,test_size=0.3)
rcv.fit(train_x, train_y.values.ravel())
pred = rcv.predict(test_x)
r2_score(test_y, pred)
```

```
0.9763336594237267
```

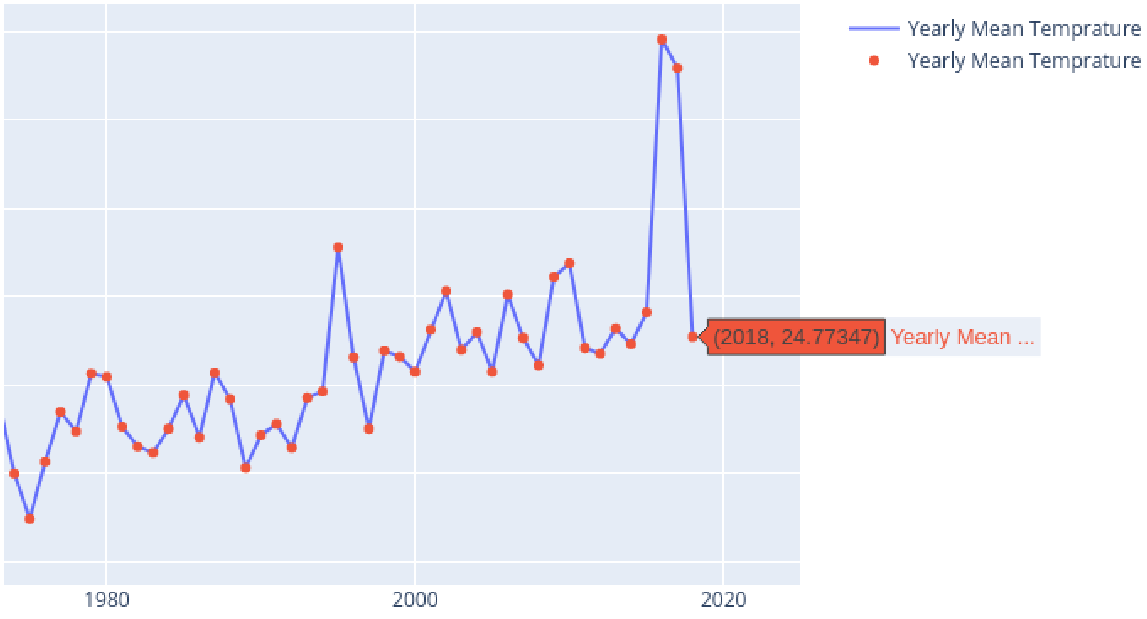


```
dtr = DecisionTreeRegressor()
train_x, test_x, train_y, test_y = train_test_split(x,y,test_size=0.3)
dtr.fit(train_x, train_y)
pred = dtr.predict(test_x)
r2_score(test_y, pred)
```

```
0.9584807072607188
```

From the analysis of past temperature data, it can be concluded that the average temperature has been increasing since the 1980s. This situation may not appear very threatening at the moment but over time, it can become disastrous. Proper steps must be taken by the government, as well as each and every single person in order to prevent further global warming, through the use of renewable sources of energy instead of the current methods of energy generation.

In this project we compared the performance of two very popular but different machine learning regression algorithms - linear and ridge regression, and decision trees. The performance of these was compared and it can be concluded that linear and ridge regression techniques performed better than the decision trees regressor at least for this problem, because the R2 score was better. Computationally, the decision trees model was able to train faster than the linear and ridge regression models, in our case.



Predictions for 2018 temperatures using Linear Regression model

REFERENCES

- https://www.python-course.eu/Decision_Trees.php
- https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html
- https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html
- https://xgboost.readthedocs.io/en/latest/parameter.html
- https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
- https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html
- https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html
- https://en.wikipedia.org/wiki/Coefficient_of_determination