**WORLD HAPPINESS REPORT DATASET**

* **Introduction**

The World Happiness Report is a landmark survey of the state of global happiness. The first report was published in 2012, the second in 2013, the third in 2015, and the fourth in 2016. The World Happiness Report-2017, which ranks 158 countries by their happiness levels, was released at the United Nations at an event celebrating International Day of Happiness on March 20th. The report continues to gain global recognition as governments, organizations and civil society increasingly use happiness indicators to inform their policy-making decisions. Leading experts across fields – economics, psychology, survey analysis, national statistics, health, public policy and more – describe how measurements of well-being can be used effectively to assess the progress of nations. The reports review the state of happiness in the world today and show how the new science of happiness explains personal and national variations in happiness.

The happiness scores and rankings use data from the Gallup World Poll. The scores are based on answers to the main life evaluation question asked in the poll. This question, known as the Cantril ladder, asks respondents to think of a ladder with the best possible life for them being a 10 and the worst possible life being a 0 and to rate their own current lives on that scale. The scores are from nationally representative samples for the years 2013-2016 and use the Gallup weights to make the estimates representative. The columns following the happiness score estimate the extent to which each of six factors – economic production, social support, life expectancy, freedom, absence of corruption, and generosity – contribute to making life evaluations higher in each country than they are in Dystopia, a hypothetical country that has values equal to the world’s lowest national averages for each of the six factors. They have no impact on the total score reported for each country, but they do explain why some countries rank higher than others.

Dystopia is an imaginary country that has the world’s least-happy people. The purpose in establishing Dystopia is to have a benchmark against which all countries can be favourably compared (no country performs more poorly than Dystopia) in terms of each of the six key variables, thus allowing each sub-bar to be of positive width. The lowest scores observed for the six key variables, therefore, characterize Dystopia. Since life would be very unpleasant in a country with the world’s lowest incomes, lowest life expectancy, lowest generosity, most corruption, least freedom and least social support, it is referred to as “Dystopia,” in contrast to Utopia.

And the following columns: GDP per Capita, Family, Life Expectancy, Freedom, Generosity, Trust Government Corruption describe the extent to which these factors contribute in evaluating the happiness in each country.

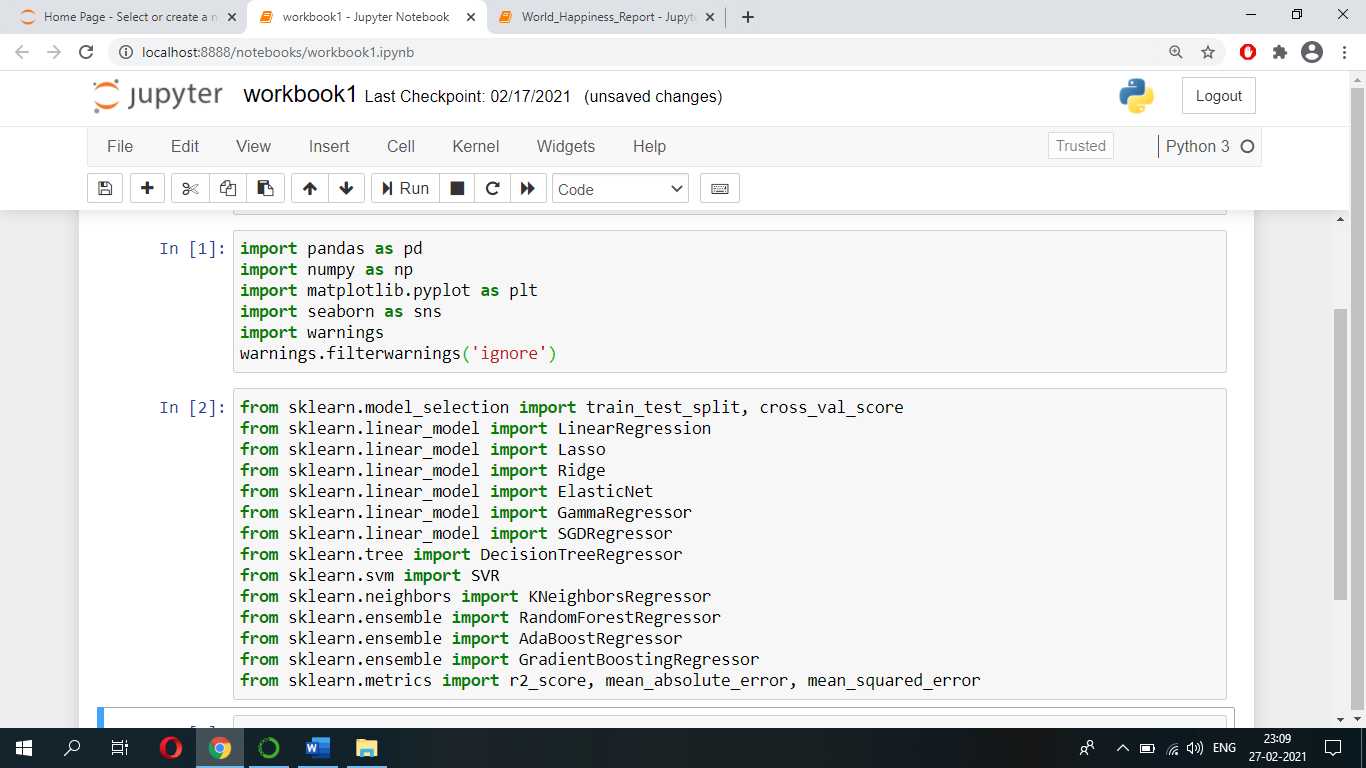
The Dystopia Residual metric actually is the Dystopia Happiness Score (1.85) + the Residual value or the unexplained value for each country.

If you add all these factors up, you get the happiness score.

* **Problem Statement**

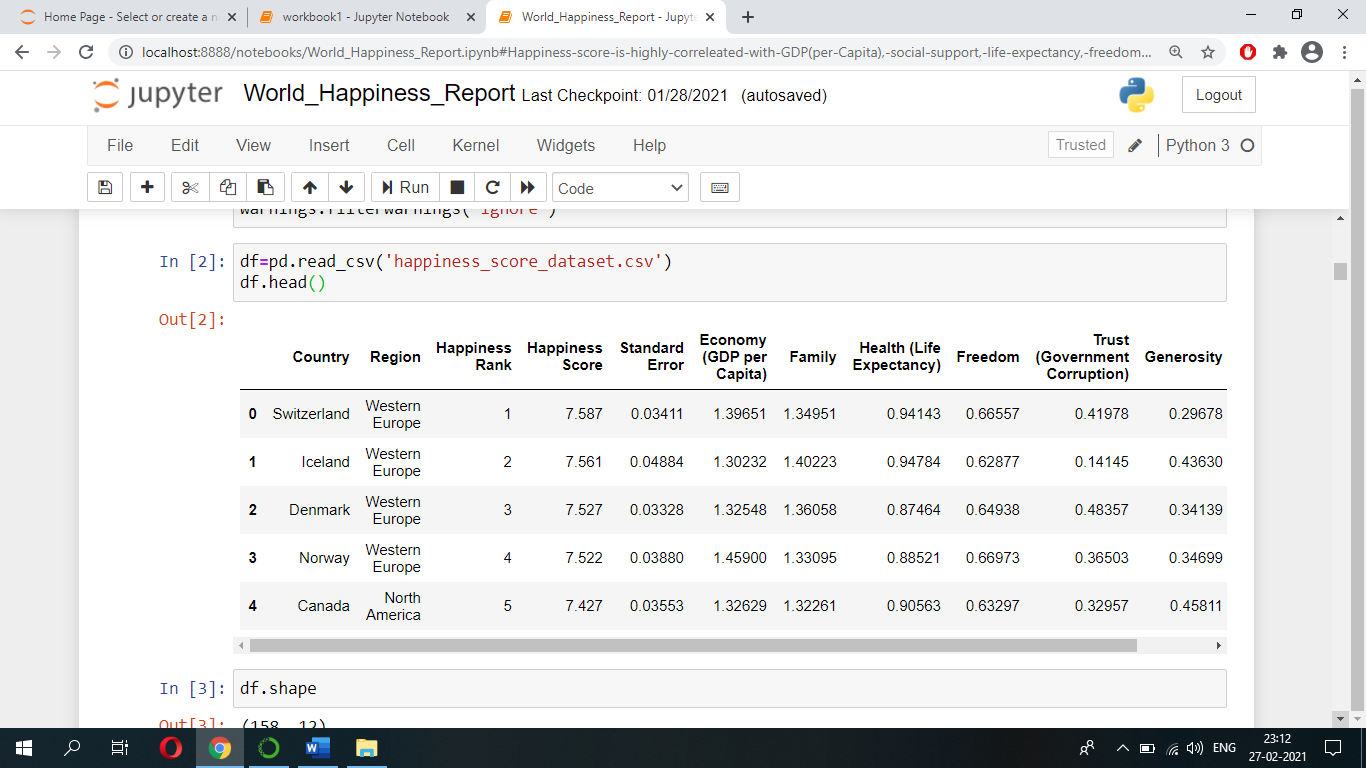
To train a model/algorithm which can predict the Happiness scores of countries (Regression problem).

* **Importing Libraries**

First of all, we will import ‘pandas’ to read our file from a csv file and manipulate it for further use. We will use ‘numpy’ to convert our data into a format suitable to feed our models. We will use ‘matplotlib’ and ‘seaborn’ for visualizations. We will then import models and its necessary metrics from ‘sklearn’. Then, we will import ‘joblib’ available in ‘sklearn’ to save our model for future use.

* **Loading Data**

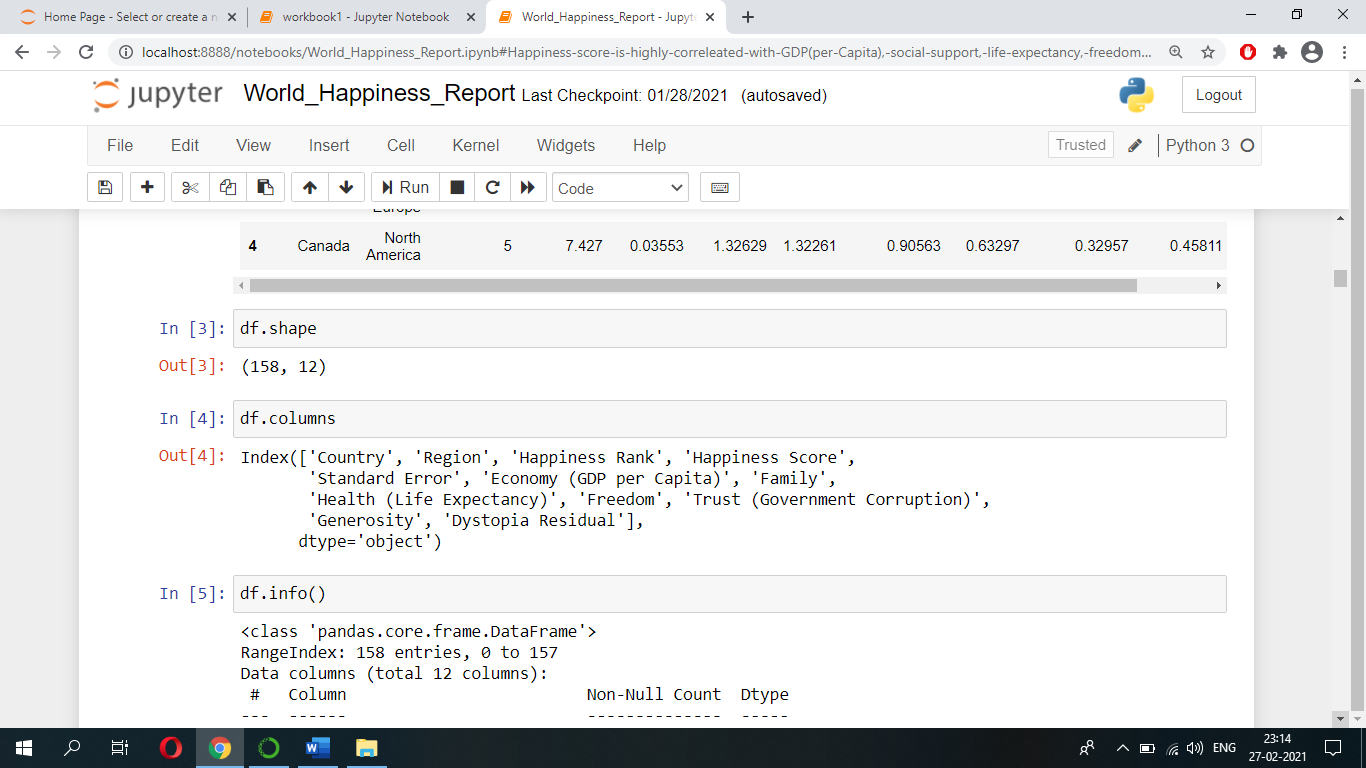
We have our data saved in an csv file named ‘happiness\_score\_dataset.csv’. We first read our dataset into a pandas dataframe called ‘df’, and then use the head() function to show the first five records of our dataset.



* **Exploratory Data Analysis (EDA)**

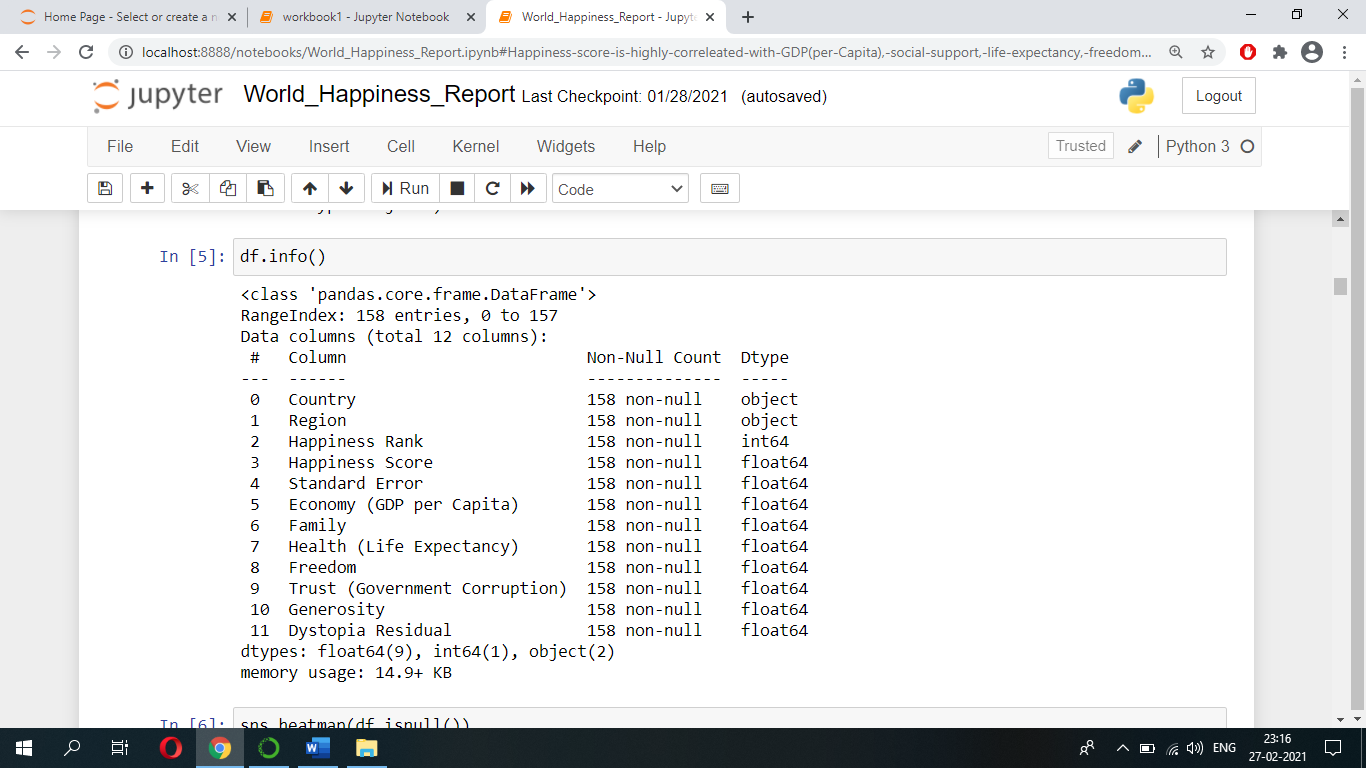
Let us now explore our dataset to get a feel of what it looks like and get some insights about it.

Let’s start by finding how many instances/rows we got in our dataset and which are the attributes/columns present in our dataset using shape and columns function.

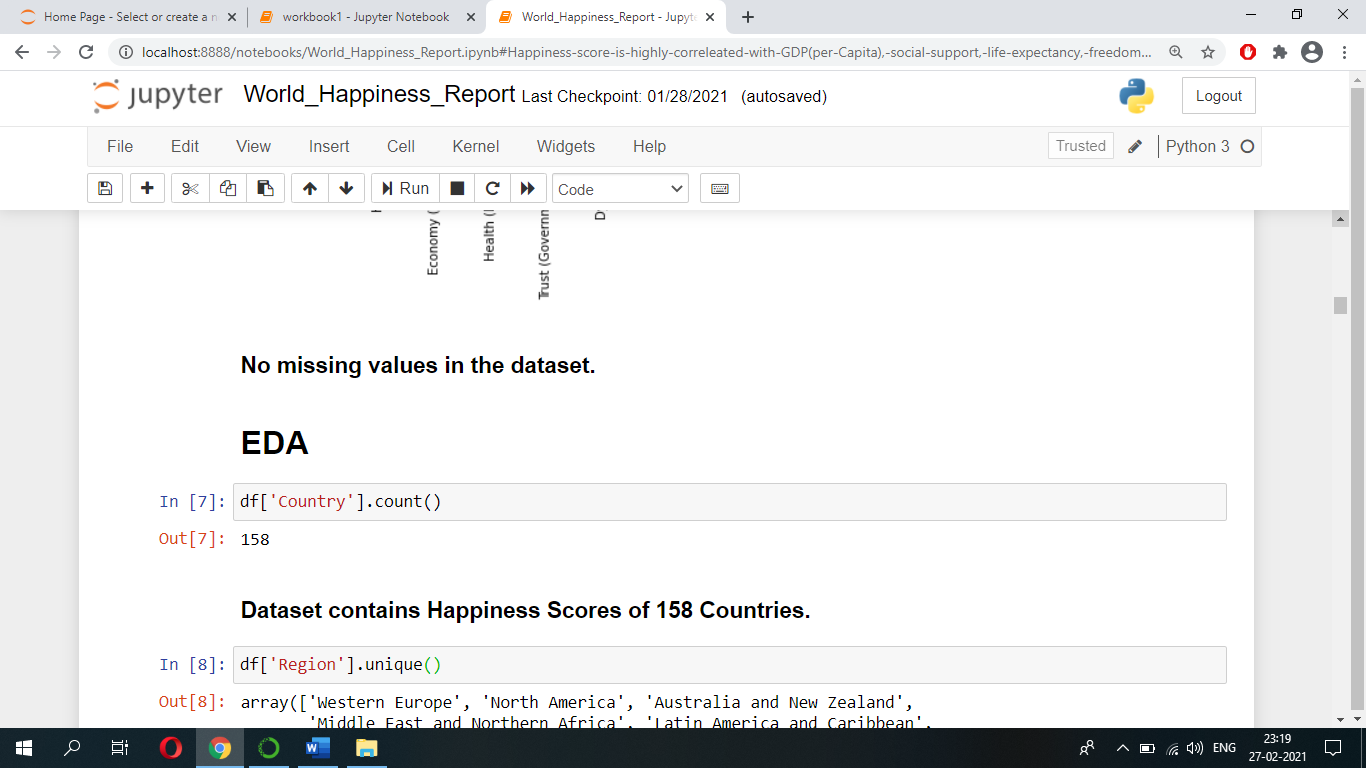


From above we can see that we got 158 instances/rows and 12 attributes/columns. And we can see which are the attributes present in our dataset.

Let’s see we got any missing values in our dataset and see what are the data-types of attributes by using info() function.

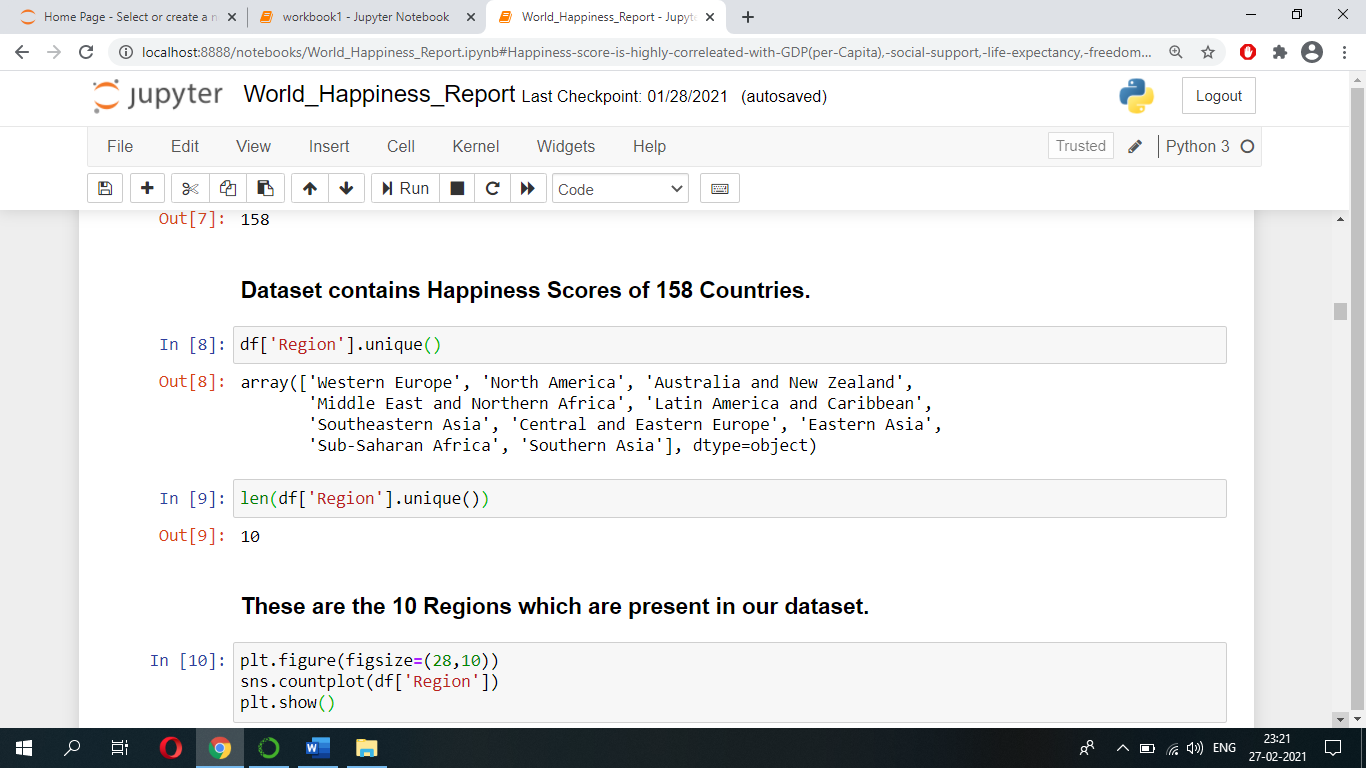


From above we can see that our dataset contains no missing values in it (we can also validate this using heatmap function) and attributes such as Country, Region is in ‘object’ data-types and rest of the attributes are in ‘int’ and ‘float’ data-types.

Let’s do some univariate analysis, first let’s find out how many countries Happiness scores are there in our dataset using count() function.

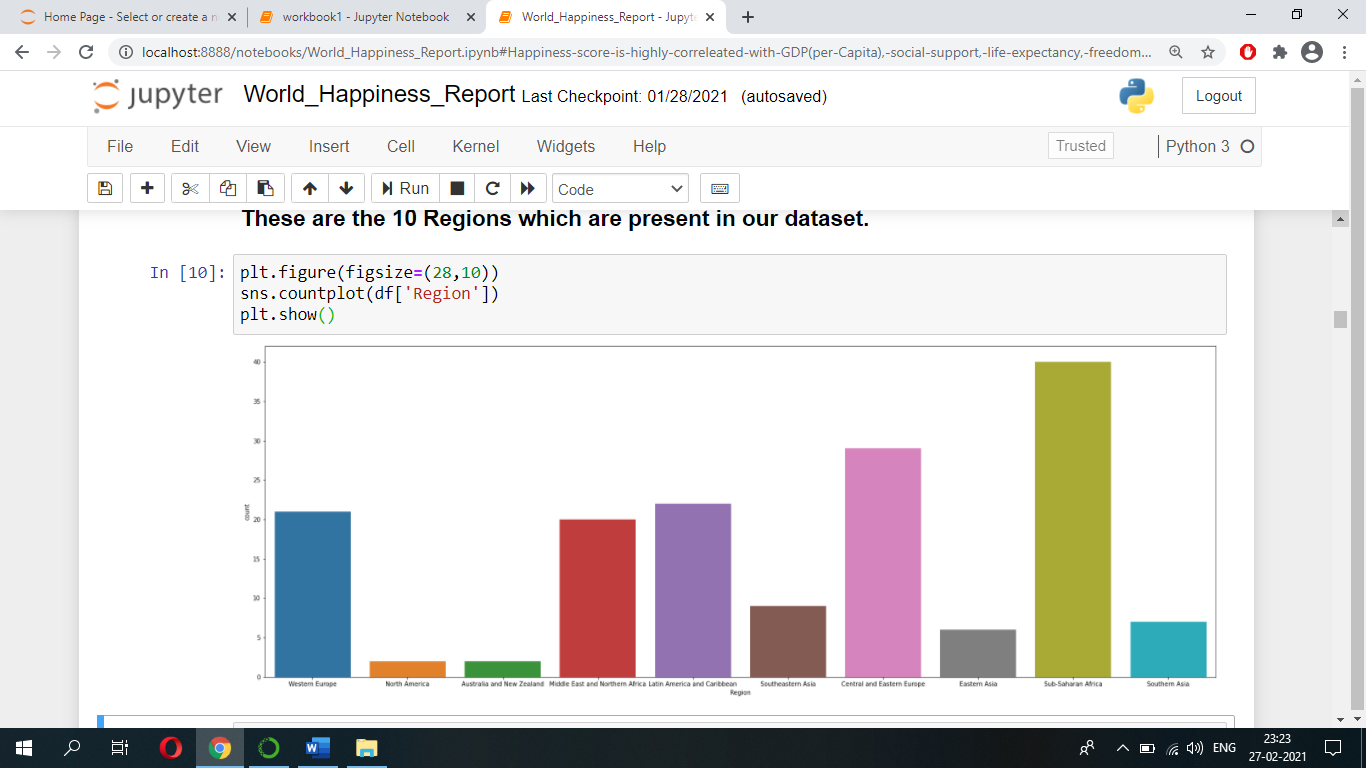
From above we can see that dataset contains Happiness Scores of 158 Countries.

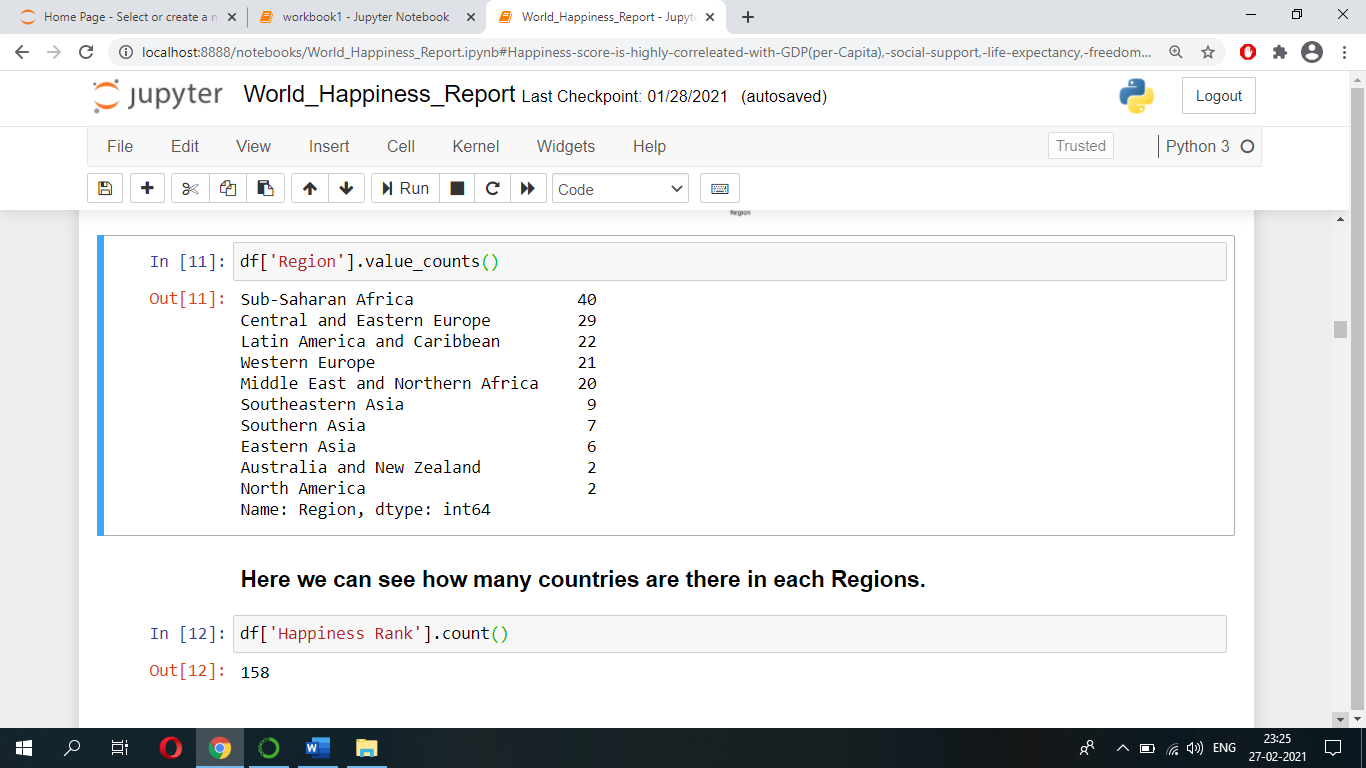
Let’s see which are the Regions that are present in our dataset using unique() function.



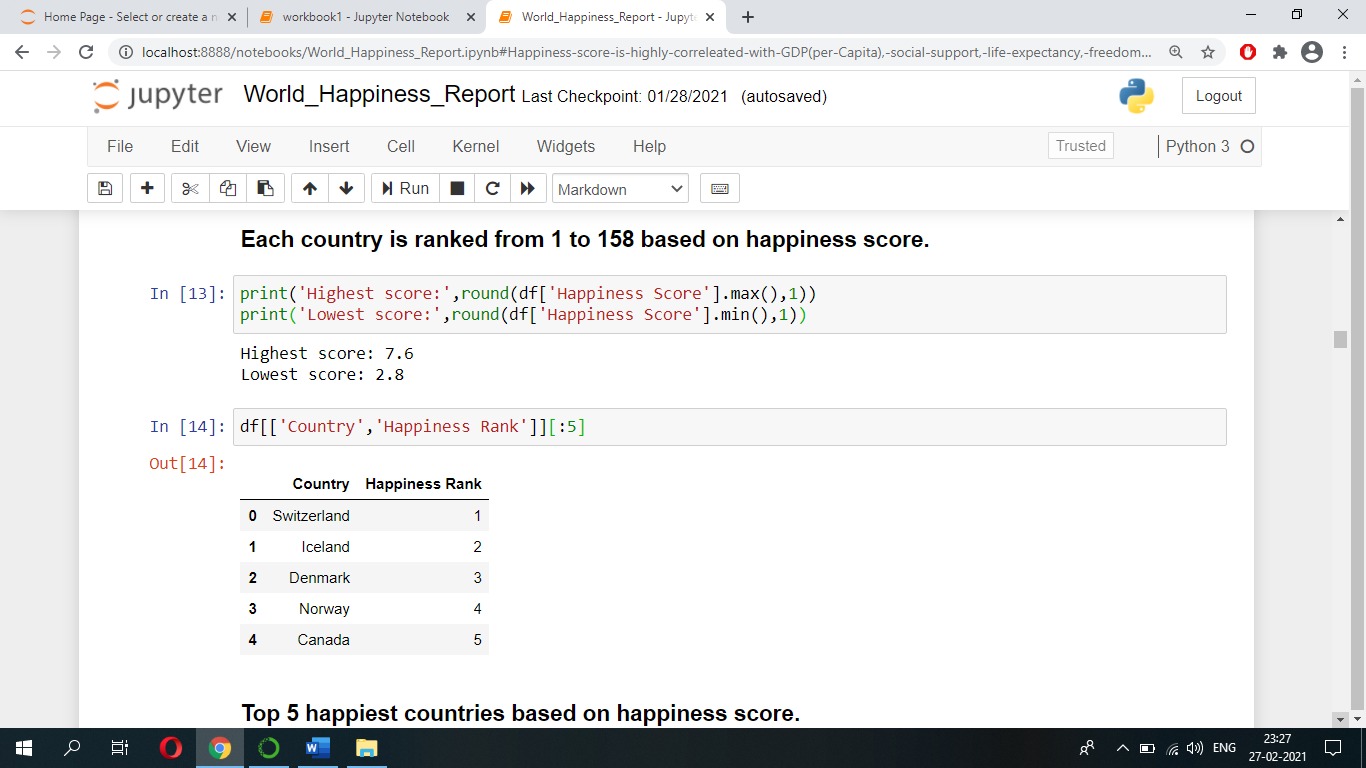
These are the 10 Regions which are present in our dataset.

Let’s see how many countries are there in each of these Regions using countplot and value\_counts() function.



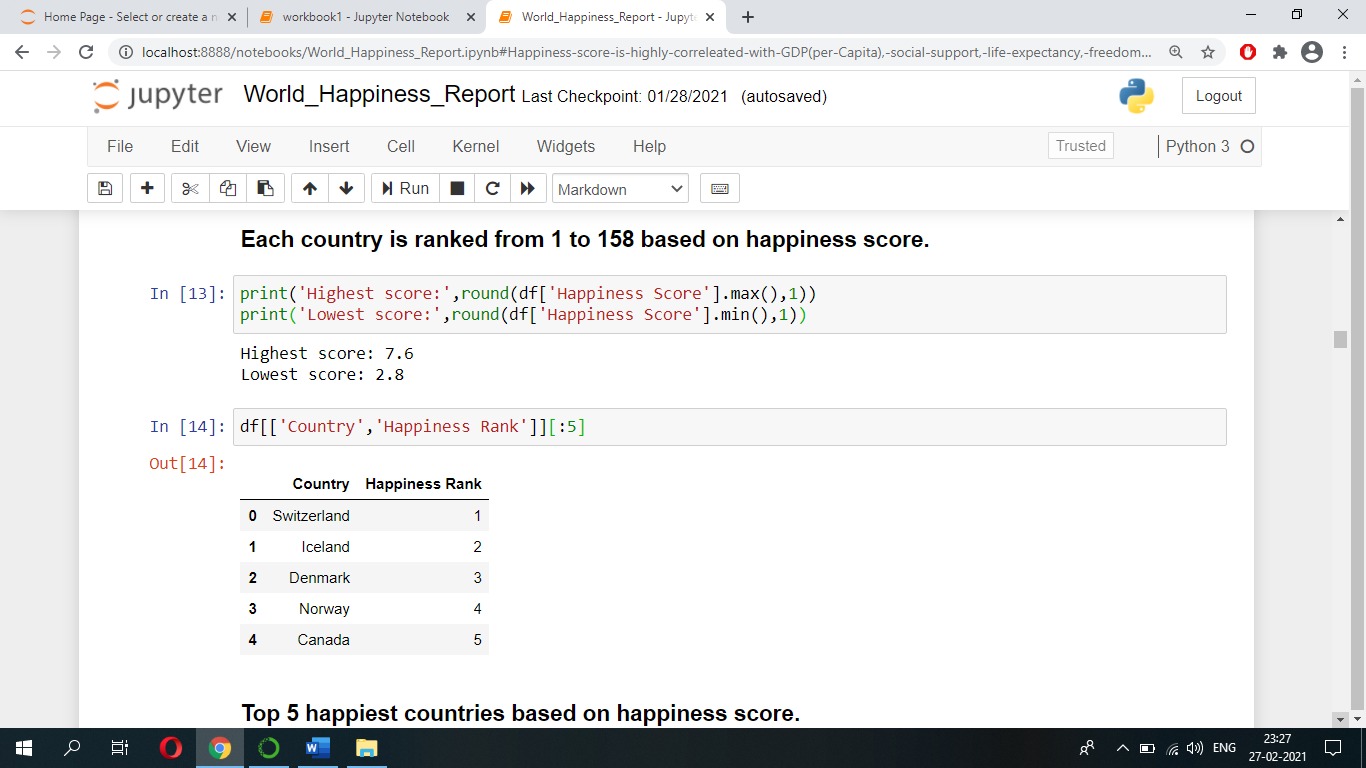


Let’s see what is the highest and the lowest Happiness score a country got in our dataset.

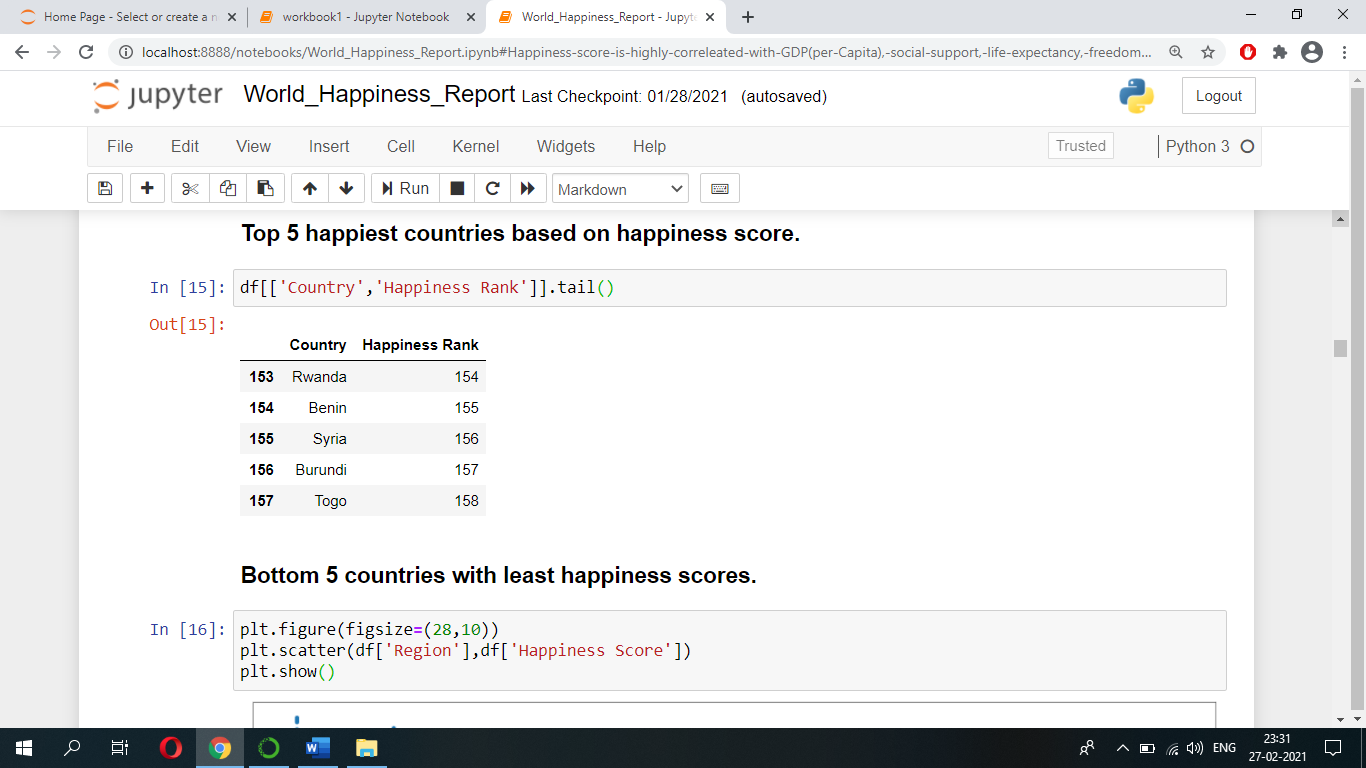


Let’s do some bivariate analysis because it will help us to understand the relations between different attributes and the target variable.

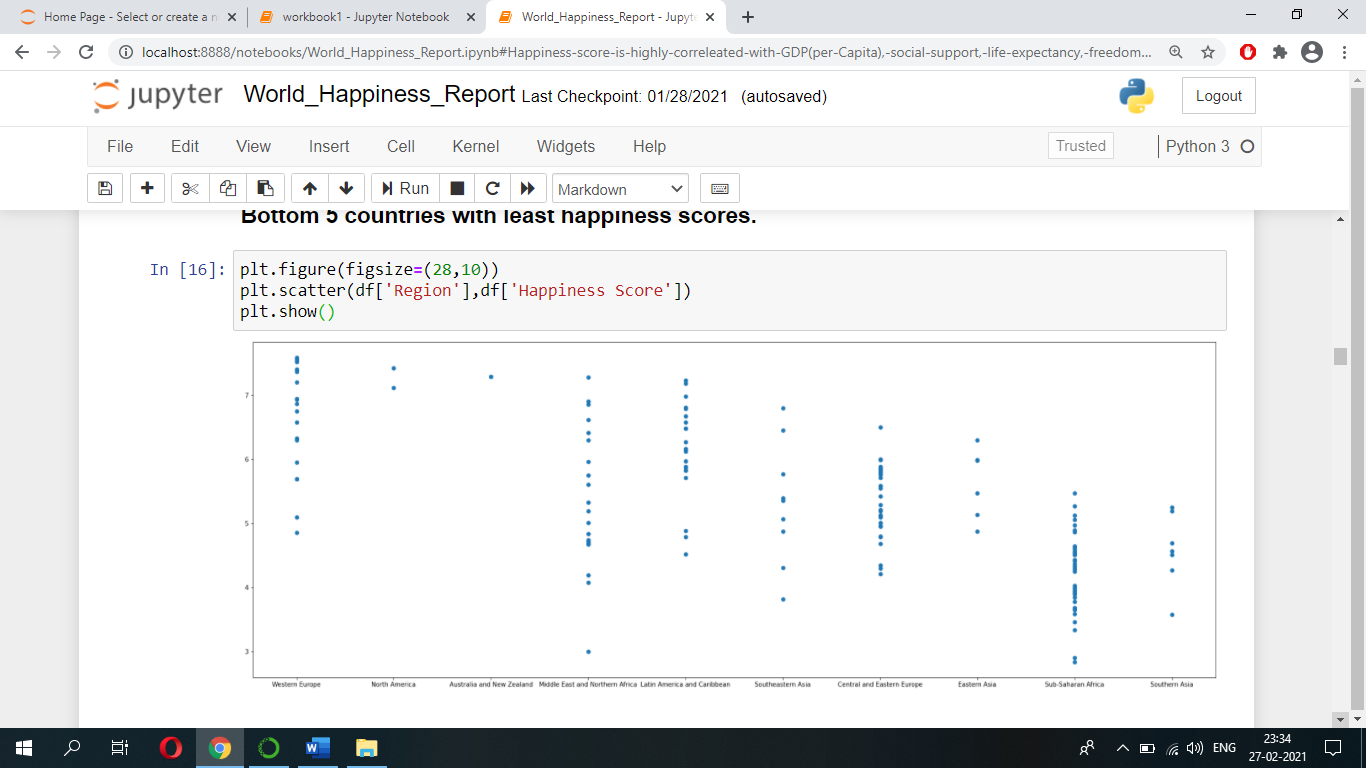
Let’s see some Top ranked countries which have high Happiness scores.



Let’s see some Bottom ranked countries which have low Happiness scores.

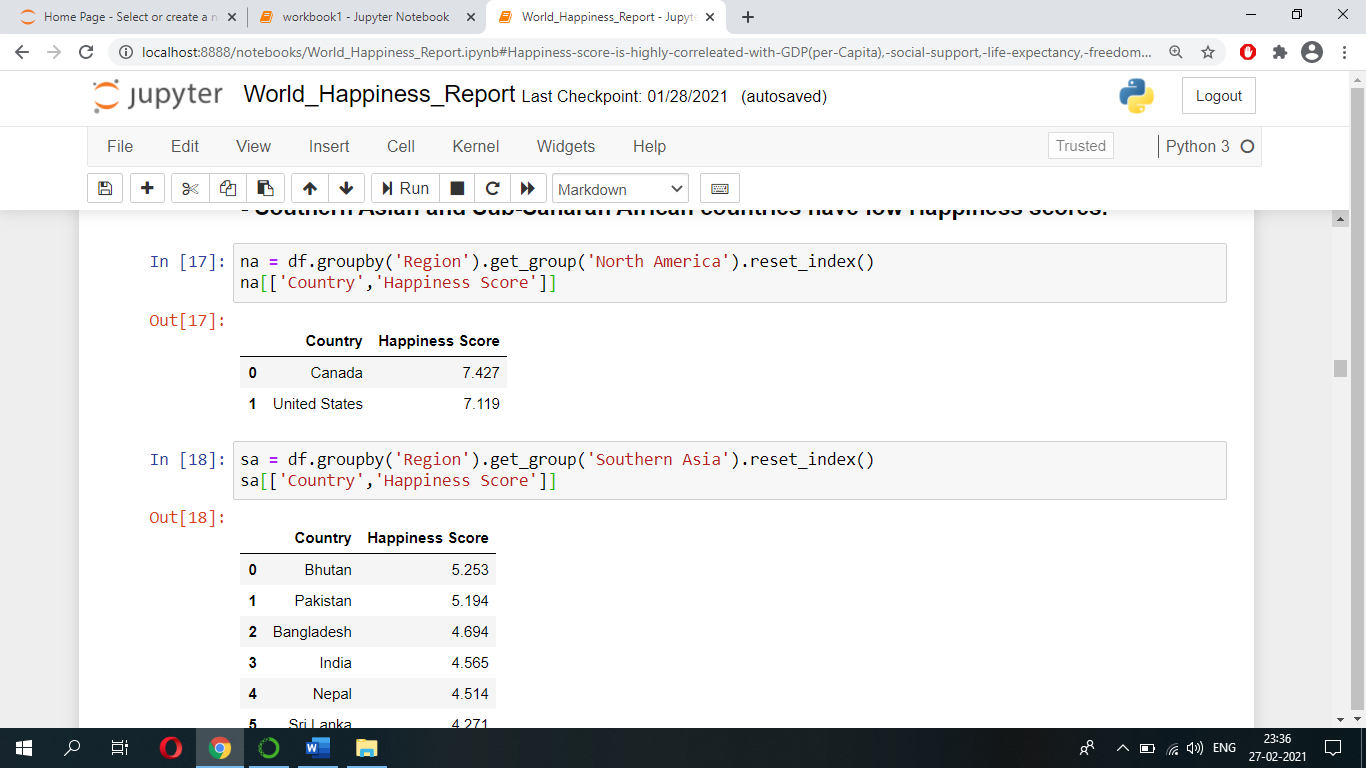


Let’s see countries belonging to which Regions have high and low Happiness scores.

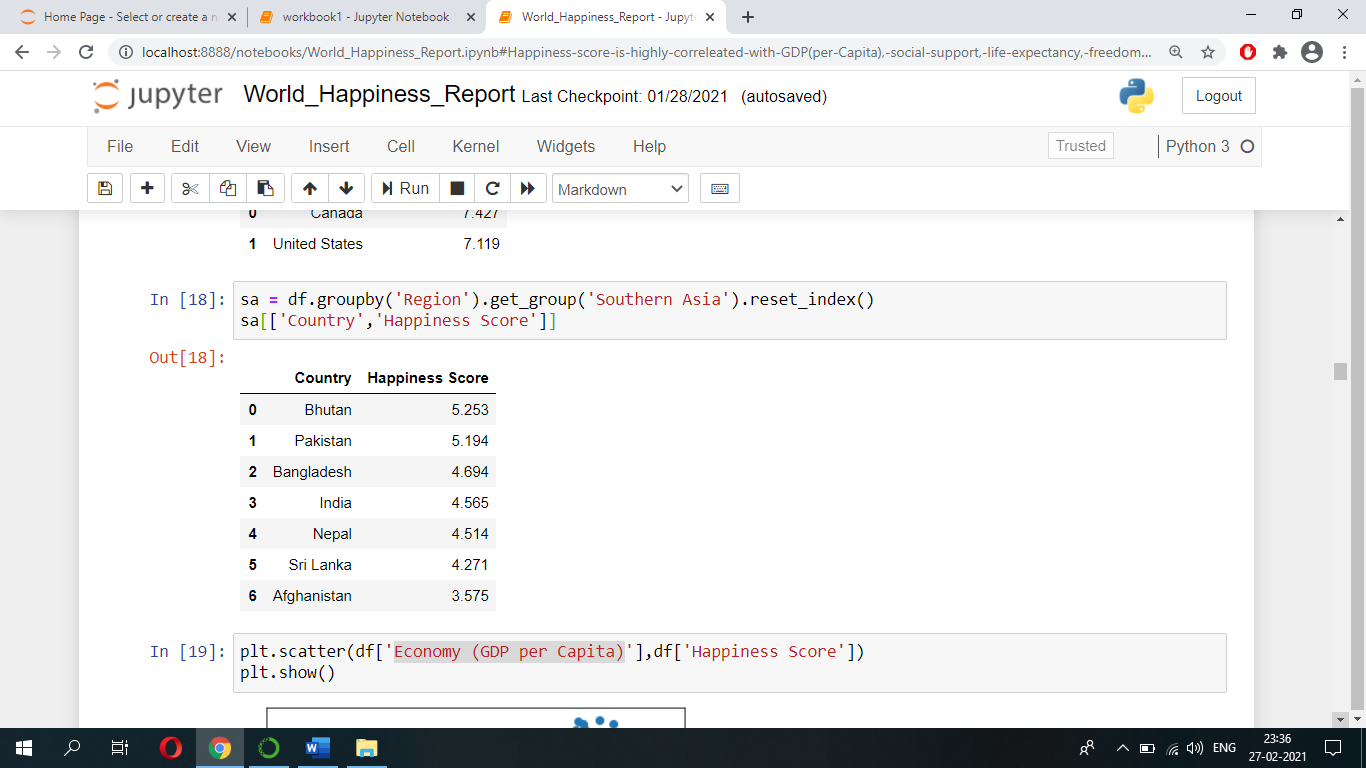


From above it is clear that Western Europe and North American countries have high Happiness scores and Sub-Saharan African and Southern Asian countries have low Happiness scores.

Let’s see North American countries Happiness scores.

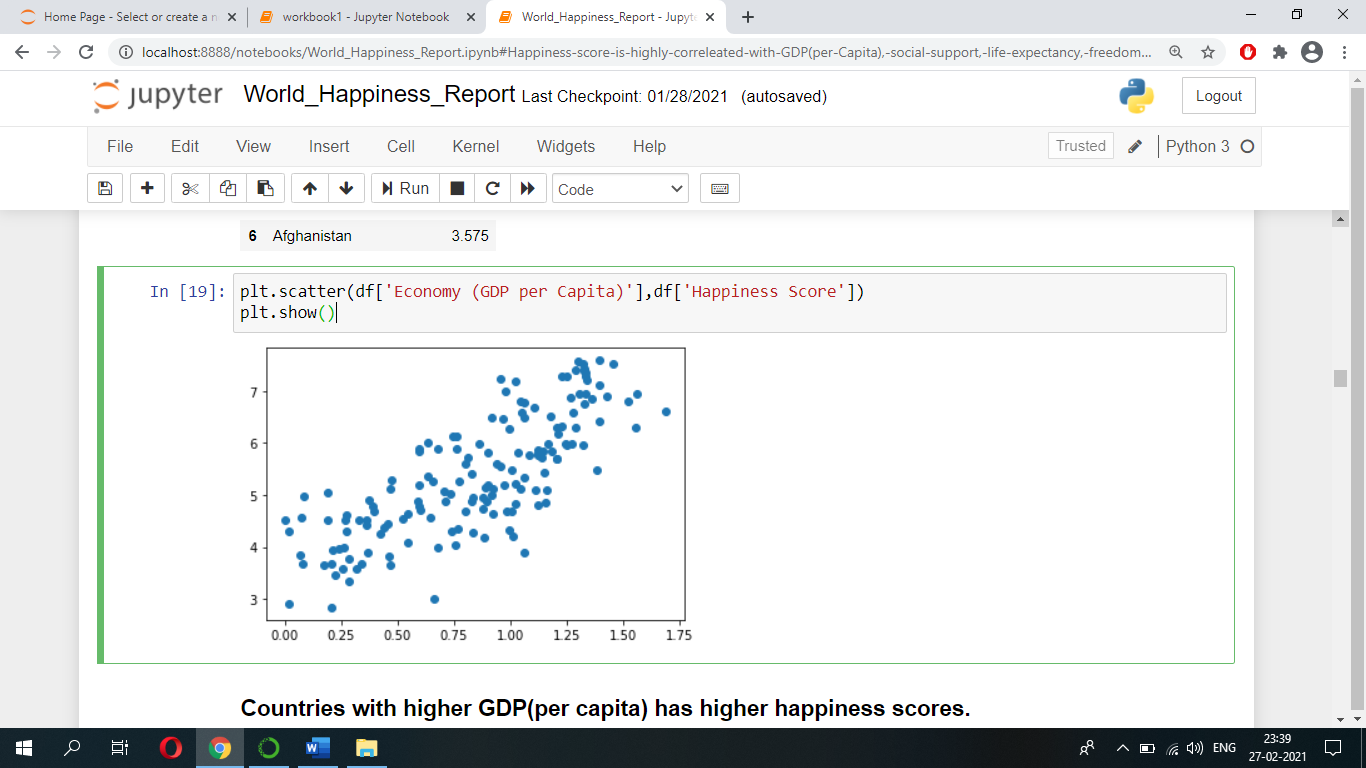


Let’s see Southern Asian countries Happiness scores.

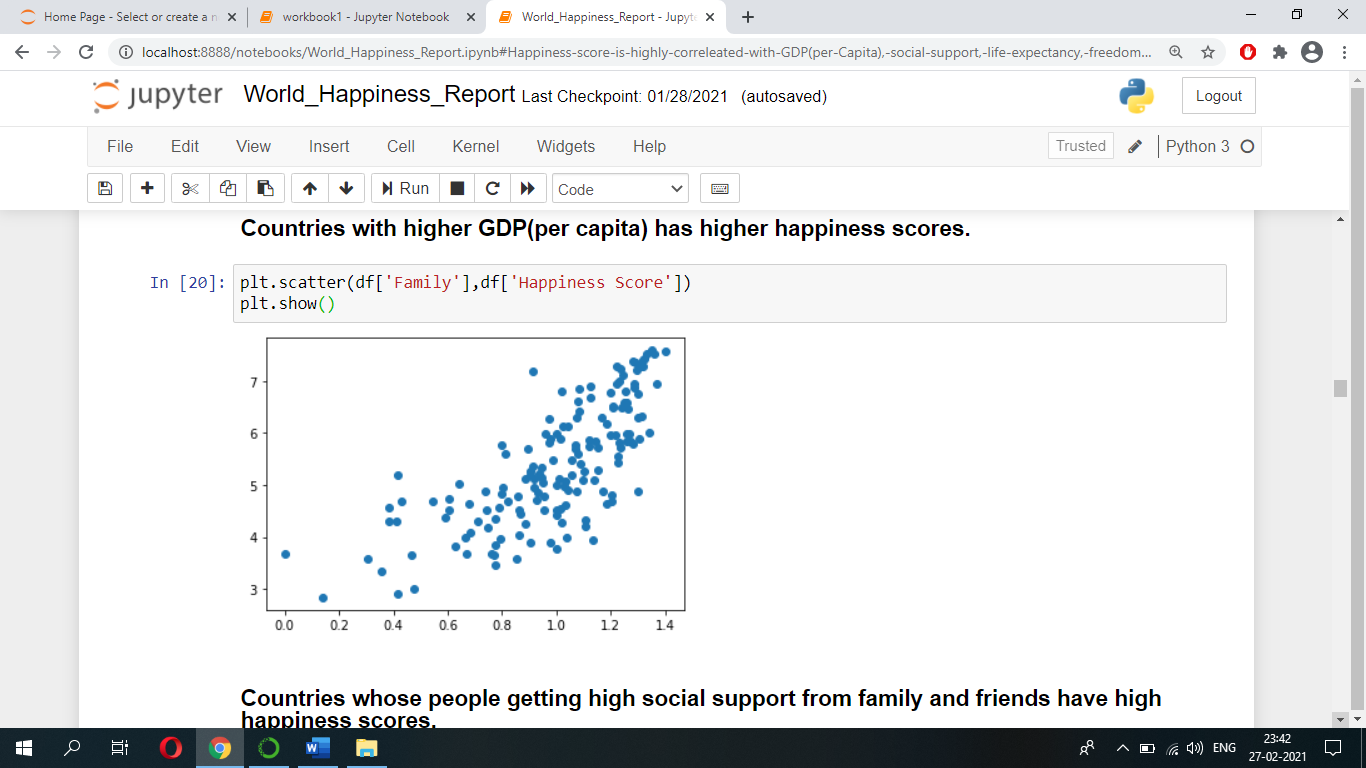


We can see India’s Happiness score is 4.56

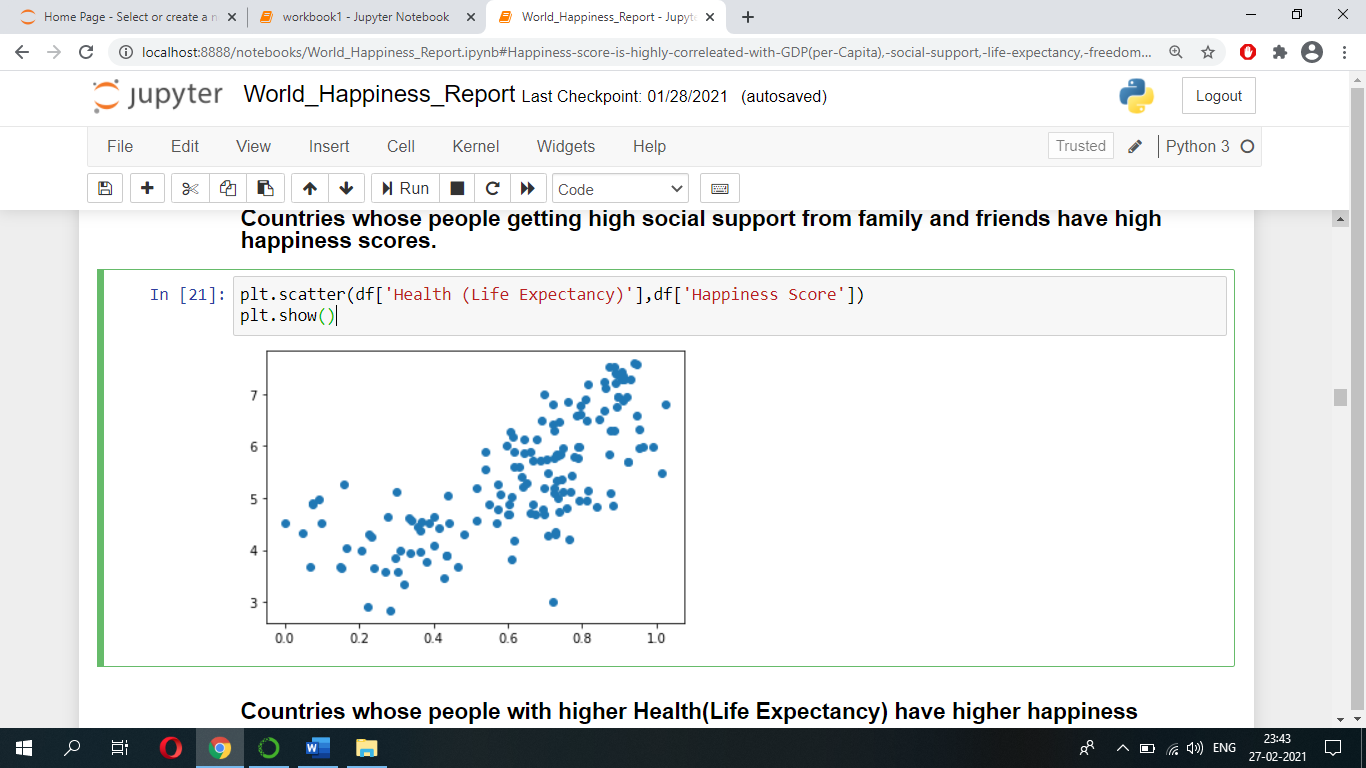
Let’s see how Happiness score varies when Economy (GDP per Capita) of a country varies.



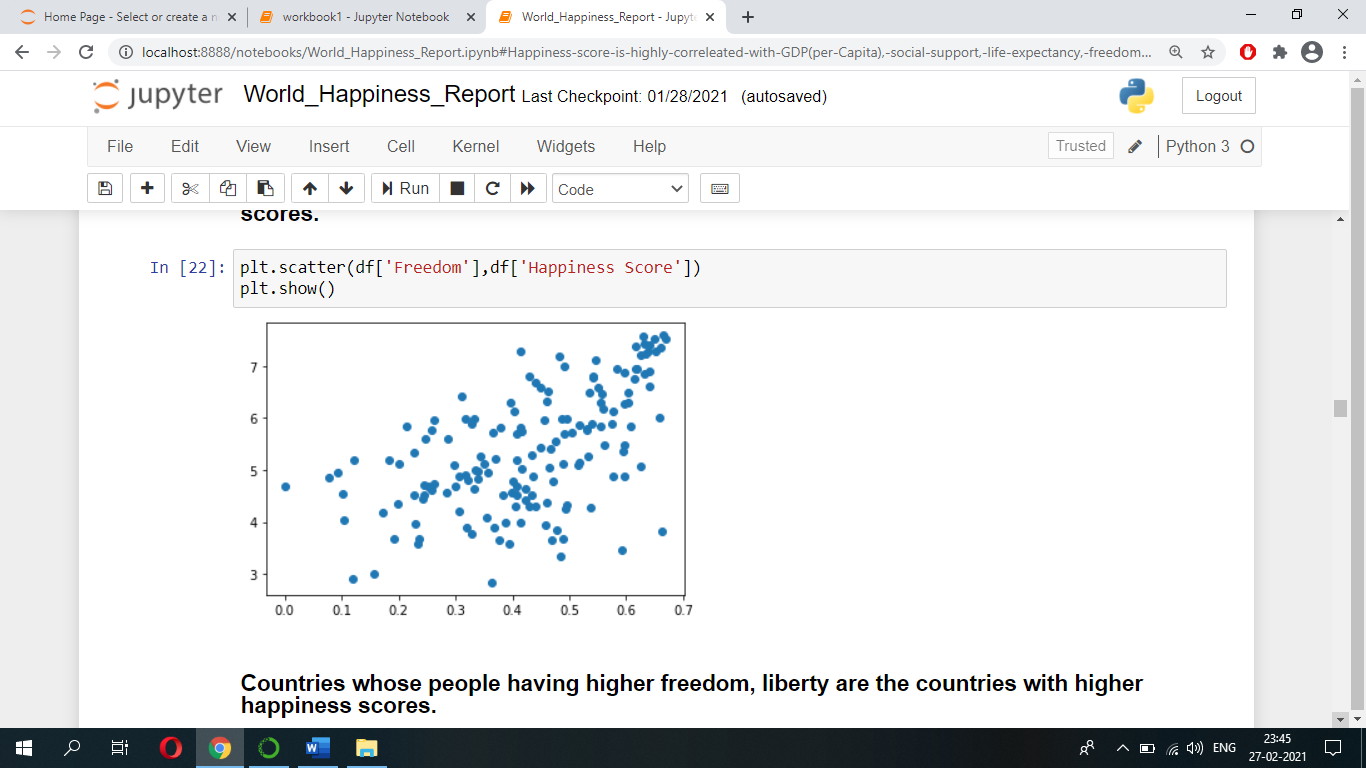
We can see countries with higher GDP (per capita) has higher Happiness scores.

Let’s see how Happiness score varies when people’s social support from family and friends varies.

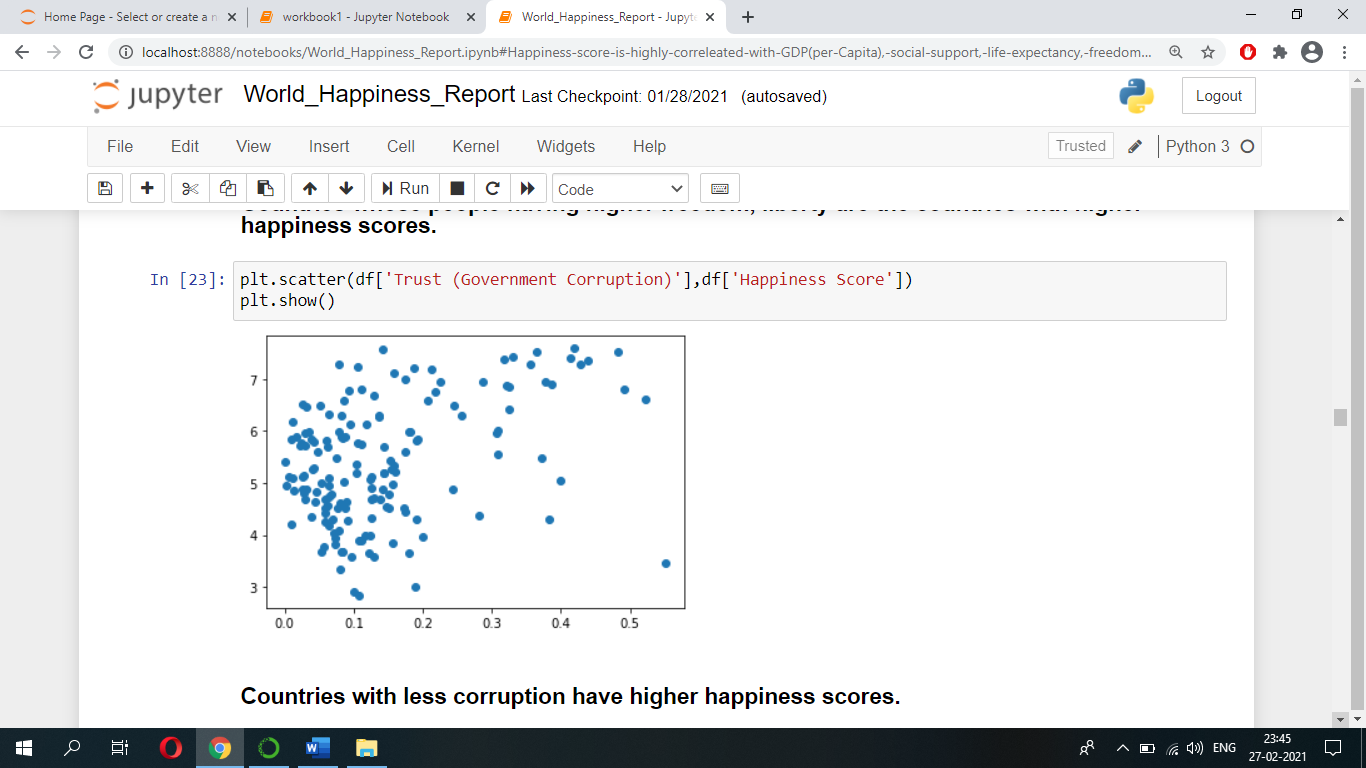
We can see countries whose people getting high social support from family and friends have high Happiness scores.

Let’s see how Happiness score varies when people’s Health (Life Expectancy) varies.

We can see countries whose people with higher Health (Life Expectancy) have higher Happiness scores.

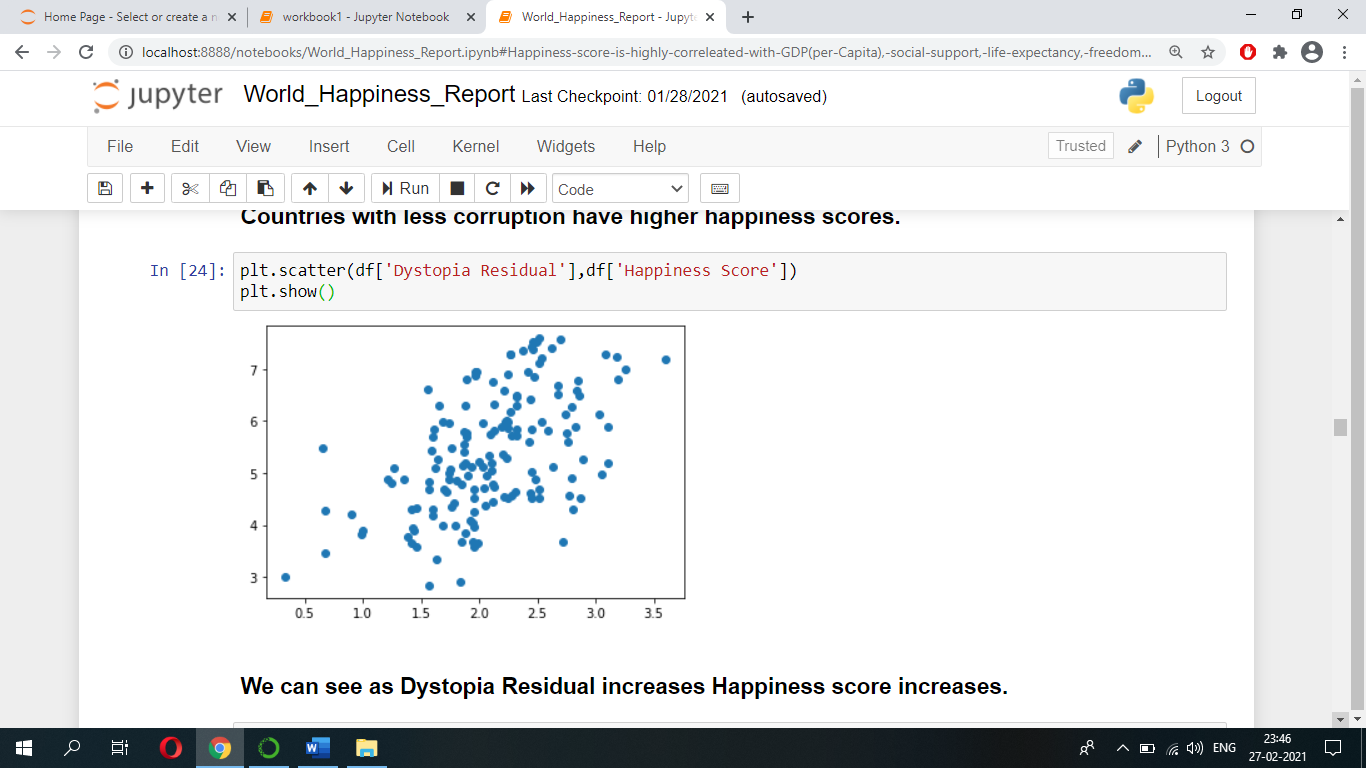
Let’s see how Happiness score varies when people’s Freedom in their country varies.

We can see countries whose citizen having higher freedom, liberty are the countries with higher Happiness scores.

Let’s see how Happiness score varies when countries Corruption level varies.

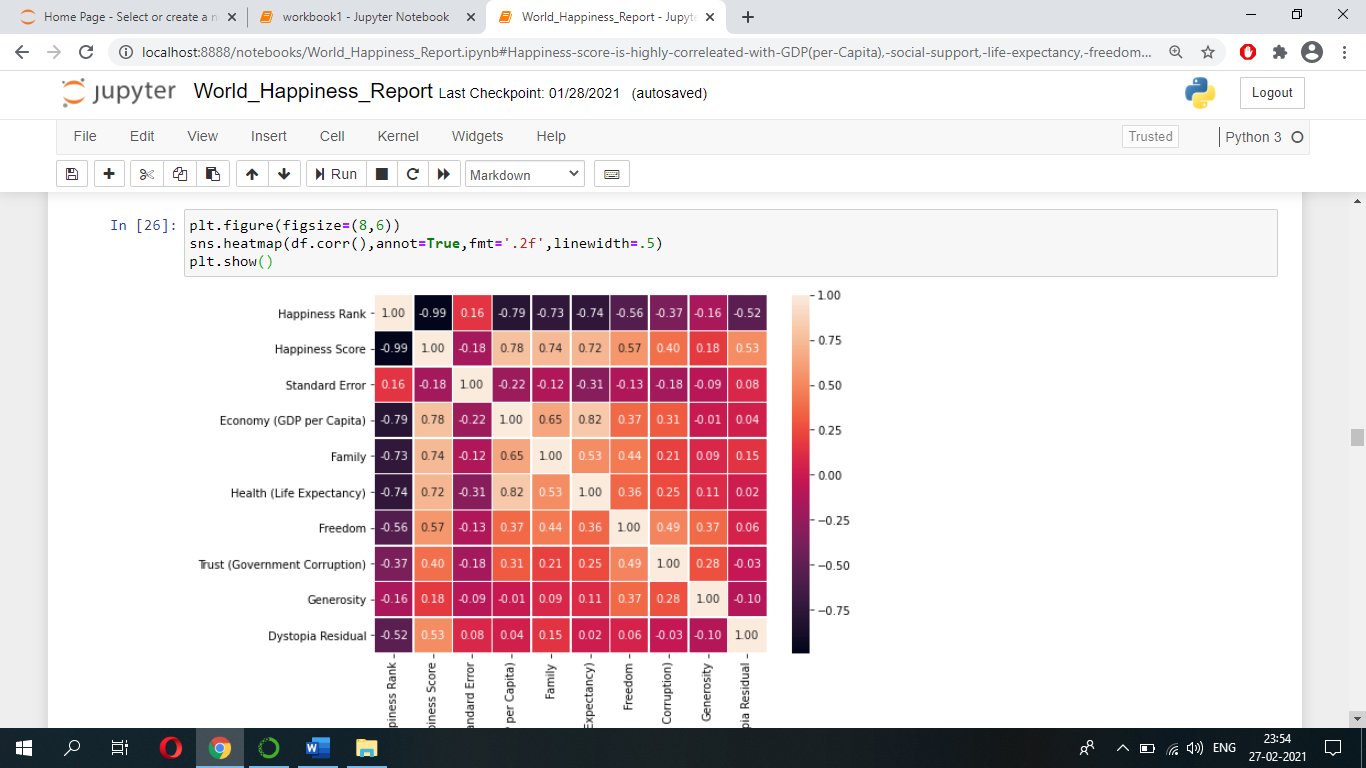
We can see majority of countries with less corruption have better Happiness score.

Let’s see how Happiness score varies when Dystopia Residual varies.



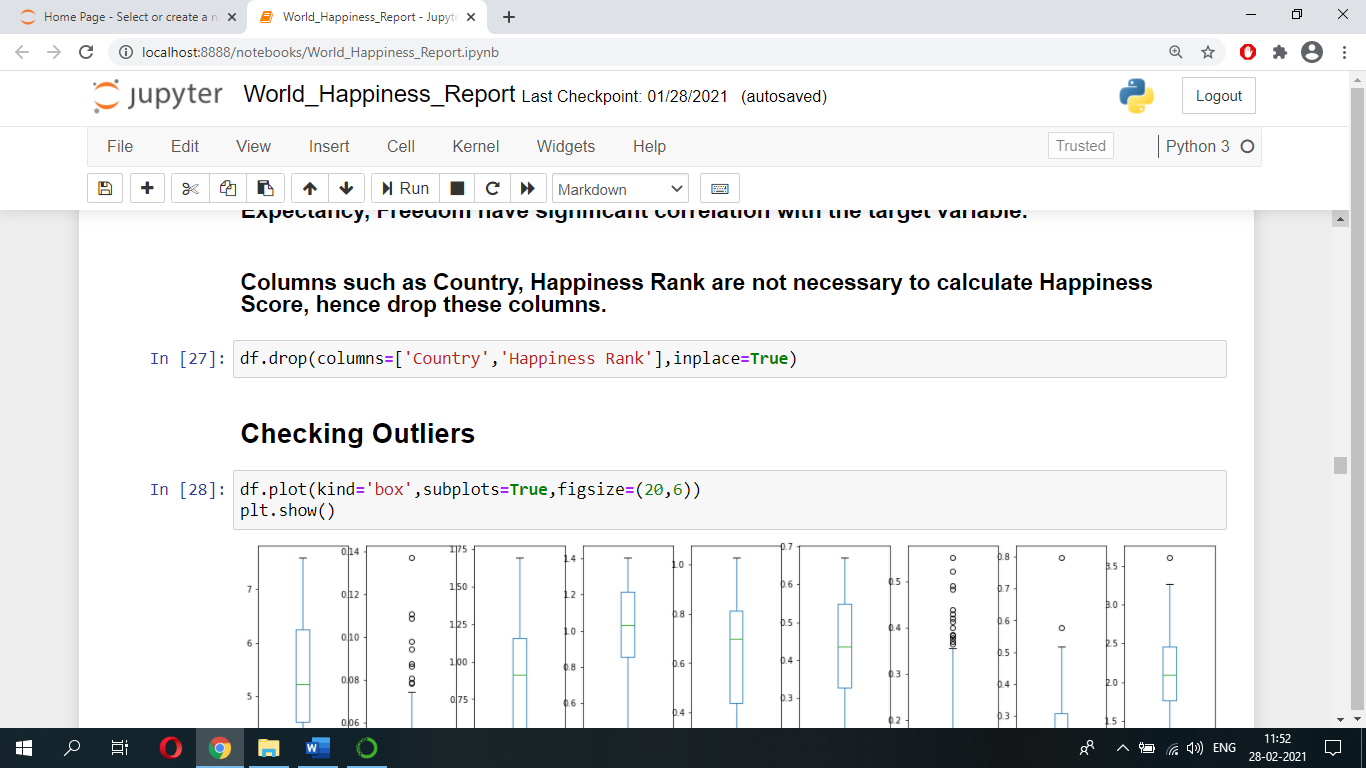
We can see as Dystopia Residual increases Happiness score increases.

Now Let’s find correlation of every pair of features (and the target variable), and visualize the correlations using a heatmap.

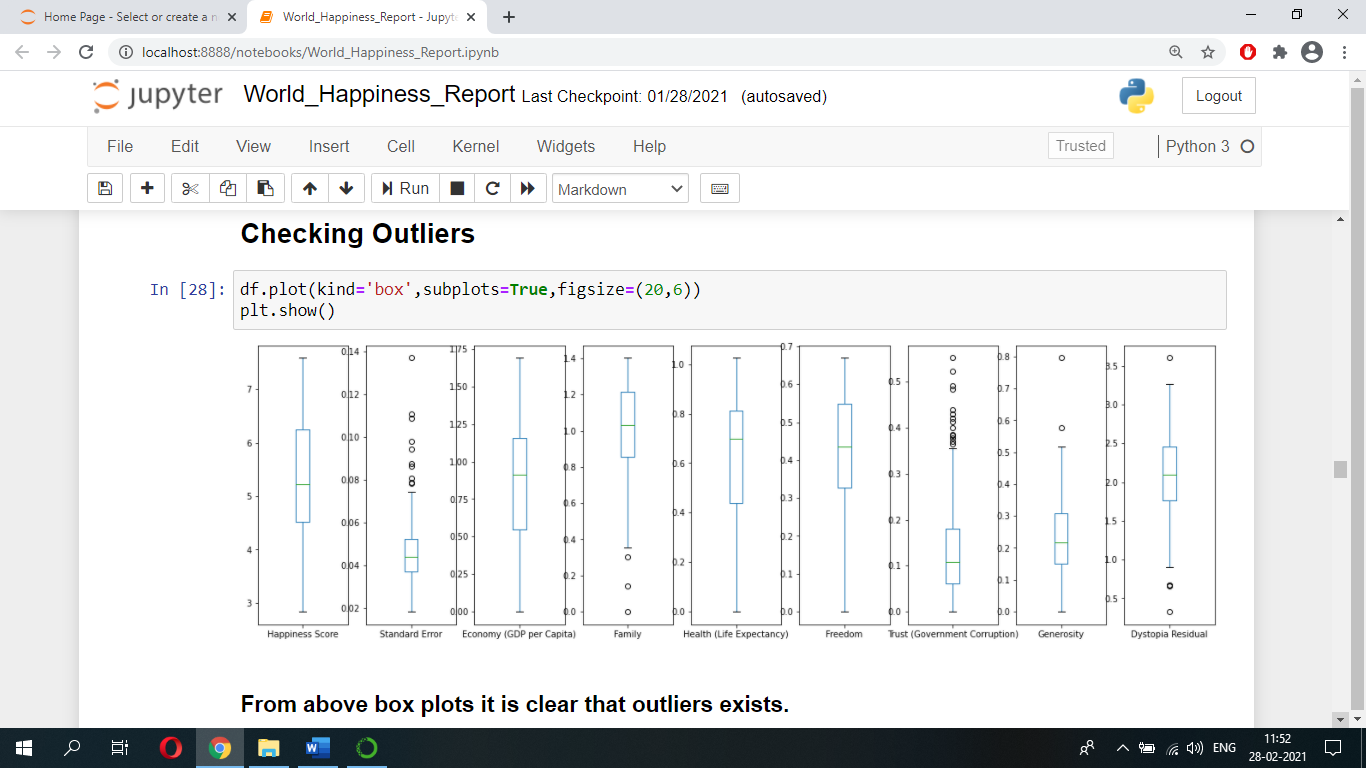


In the above heatmap, lighter colours indicate more correlation. As we can see from the table and the heatmap, attributes such as Economy (GDP per Capita), Family, Health (Life Expectancy, Freedom have significant correlation with the target variable.

* **Data Pre-processing**
* **Dropping columns**

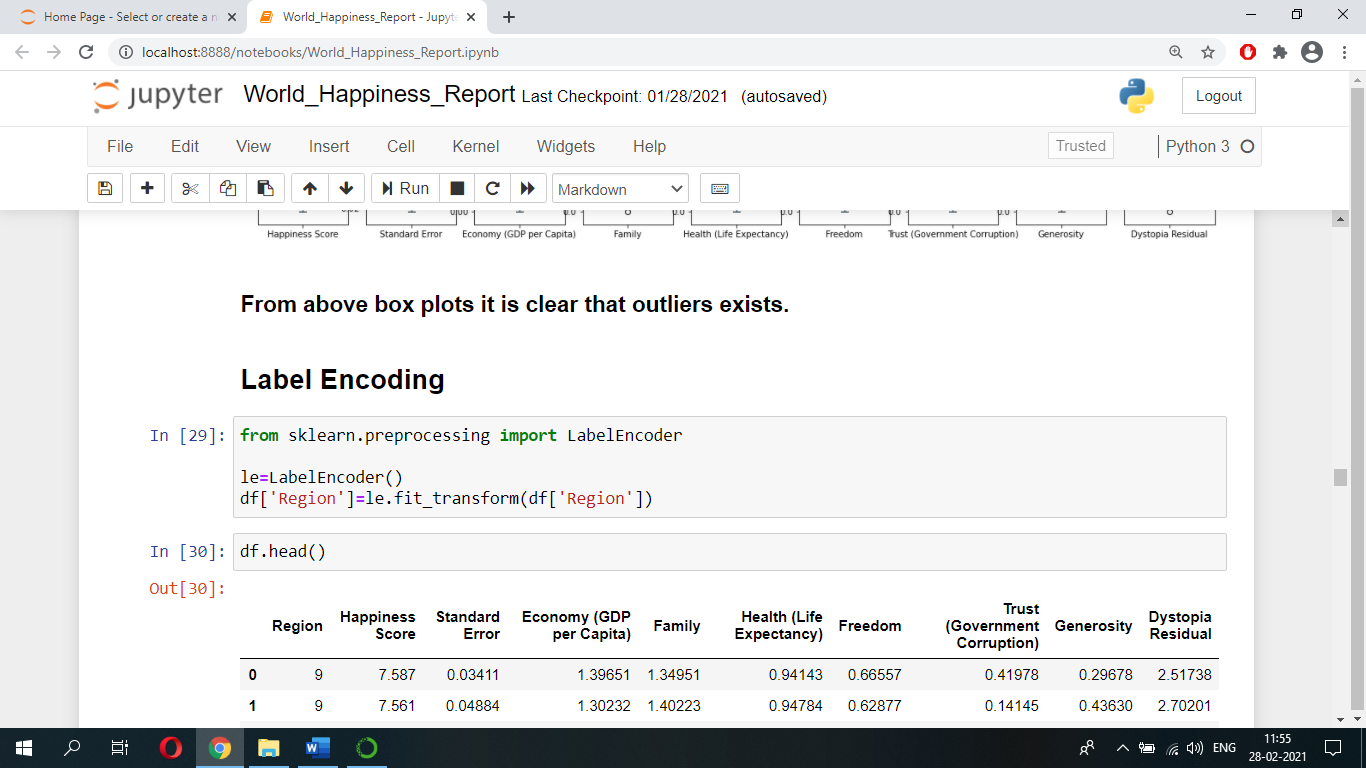
Columns such as Country, Happiness Rank are not necessary to calculate Happiness Score, hence we can drop these columns.

* **Checking Outliers**

Let’s check whether the dataset has any outliers by plotting boxplots. Below is the plot:

From above boxplots it is evident that dataset contains outliers.

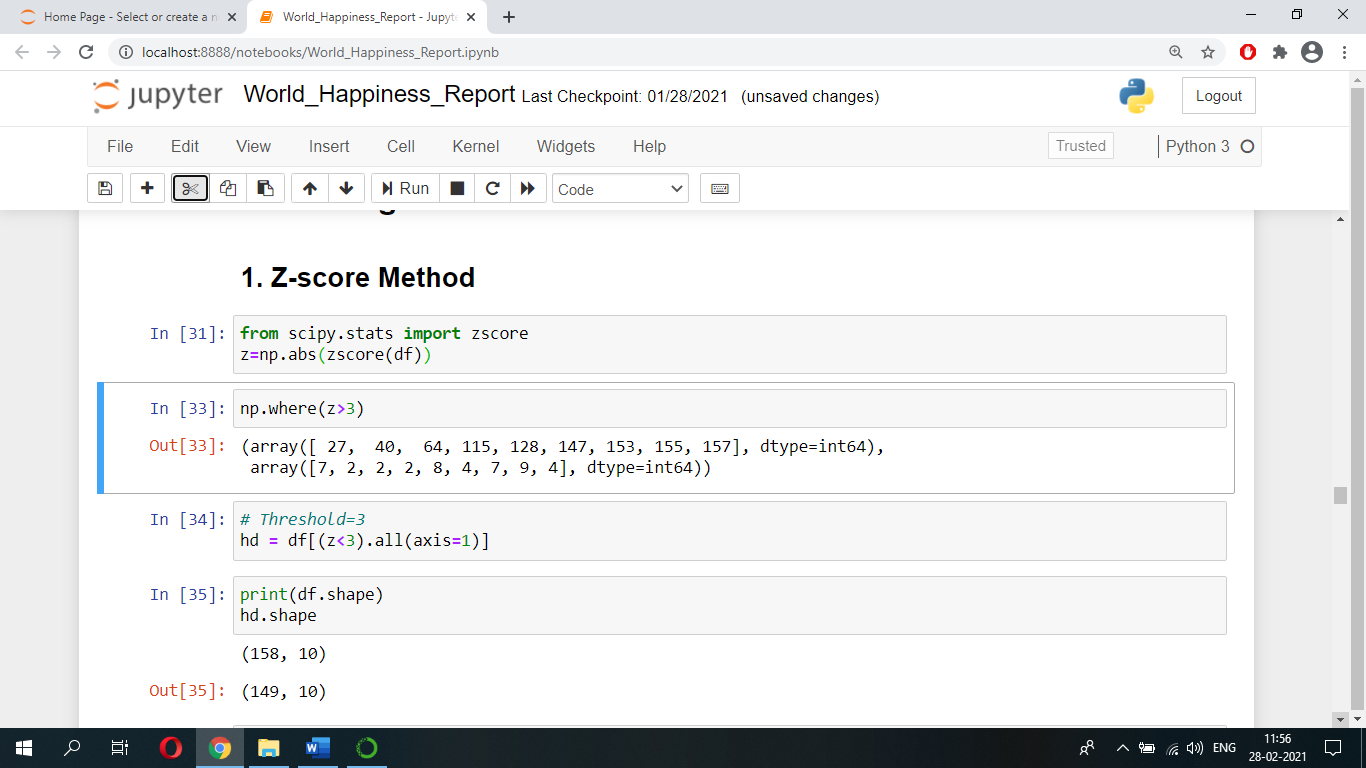
* **Label Encoding**

Before removing outliers, we should convert all columns values into numerical values. Since Region column is in ‘object’ data-type i.e., its values are of categorical type, hence we should convert these into numerical values. Therefore, we are using Label Encoder to convert these categorical values into numerical values.

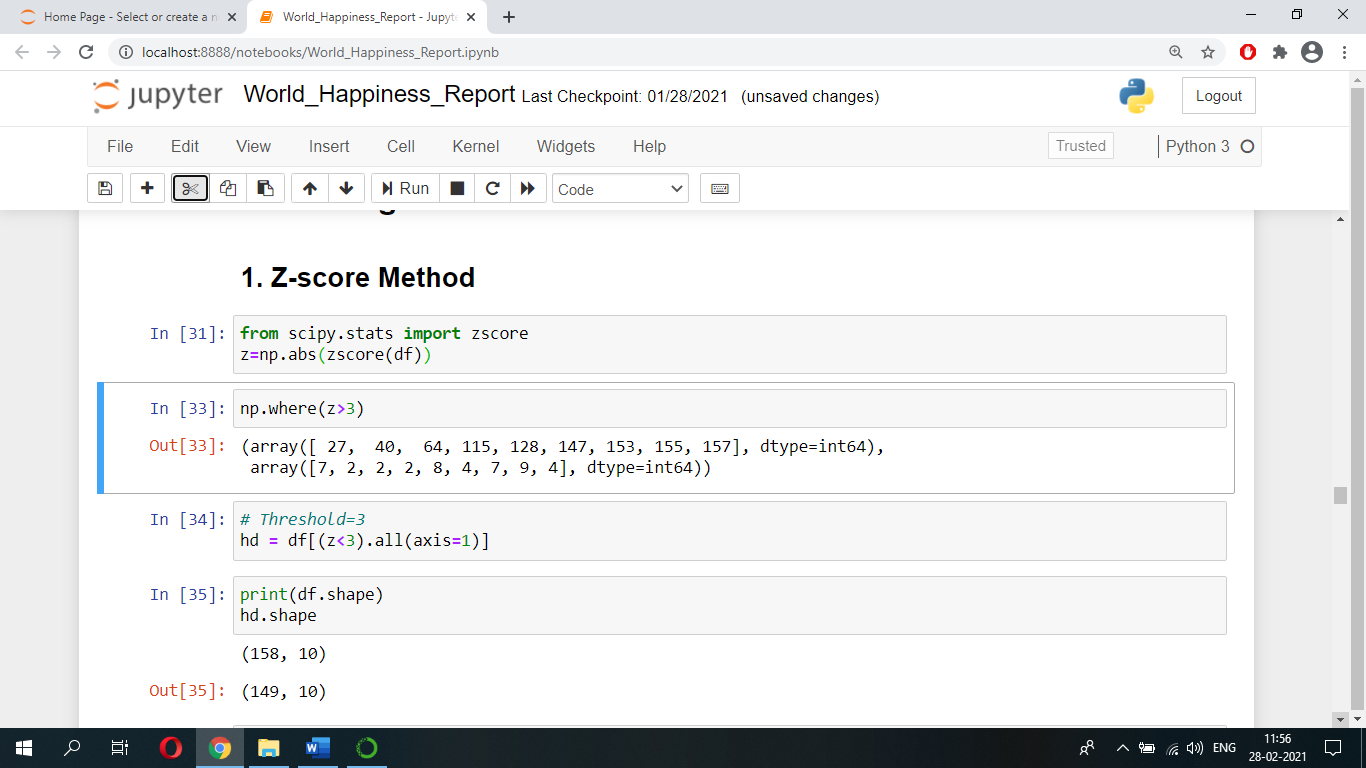
* **Removing Outliers**

We have several methods to remove outliers, but the two widely used methods to remove outliers are, Z-score method and IQR method. Let’s select Z-score method to remove outliers.

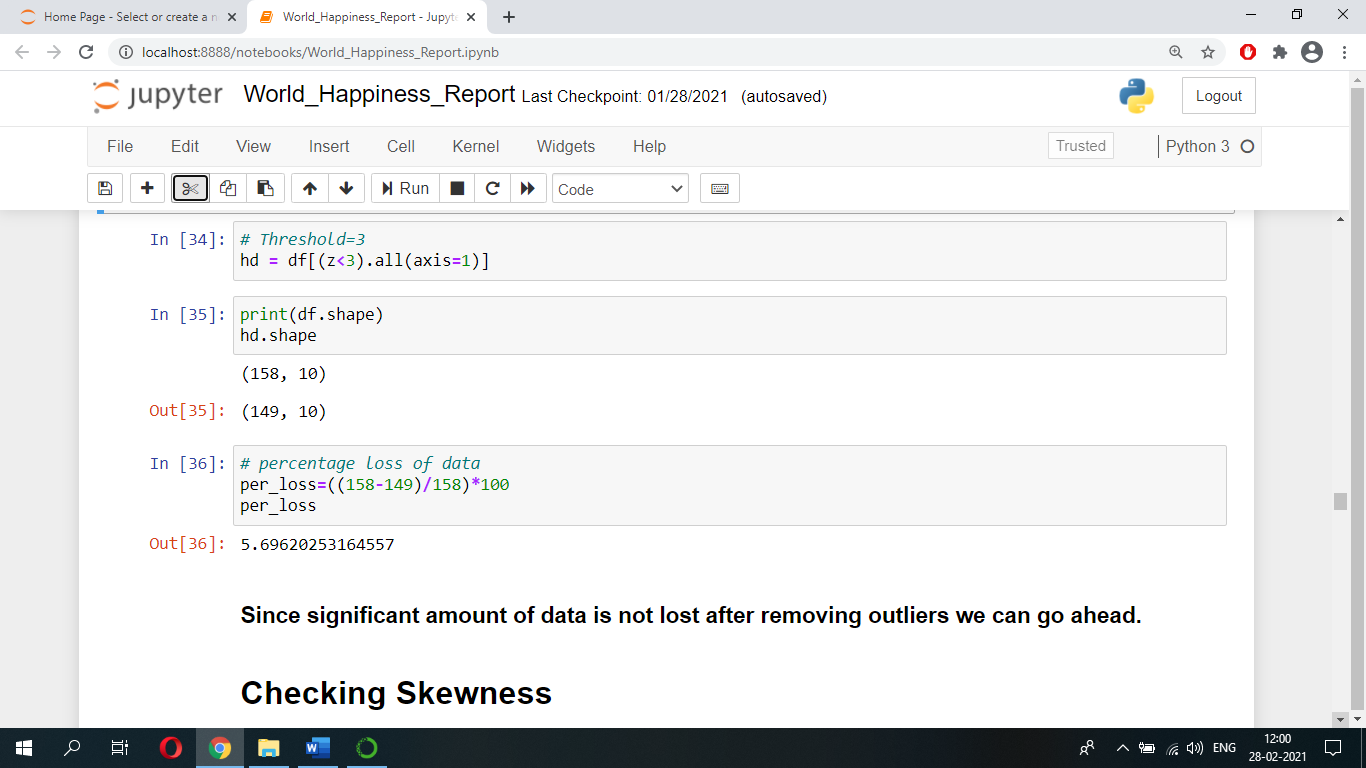
Let’s import ‘zscore’ function from ‘scipy.stats’ and find out the zscore value of columns values.



Let’s consider, any value beyond +/-3 Standard Deviation i.e., let’s take Threshold=3, as outliers. And if any such values exist, then let’s remove it. After removing outliers save the data into ‘hd’.



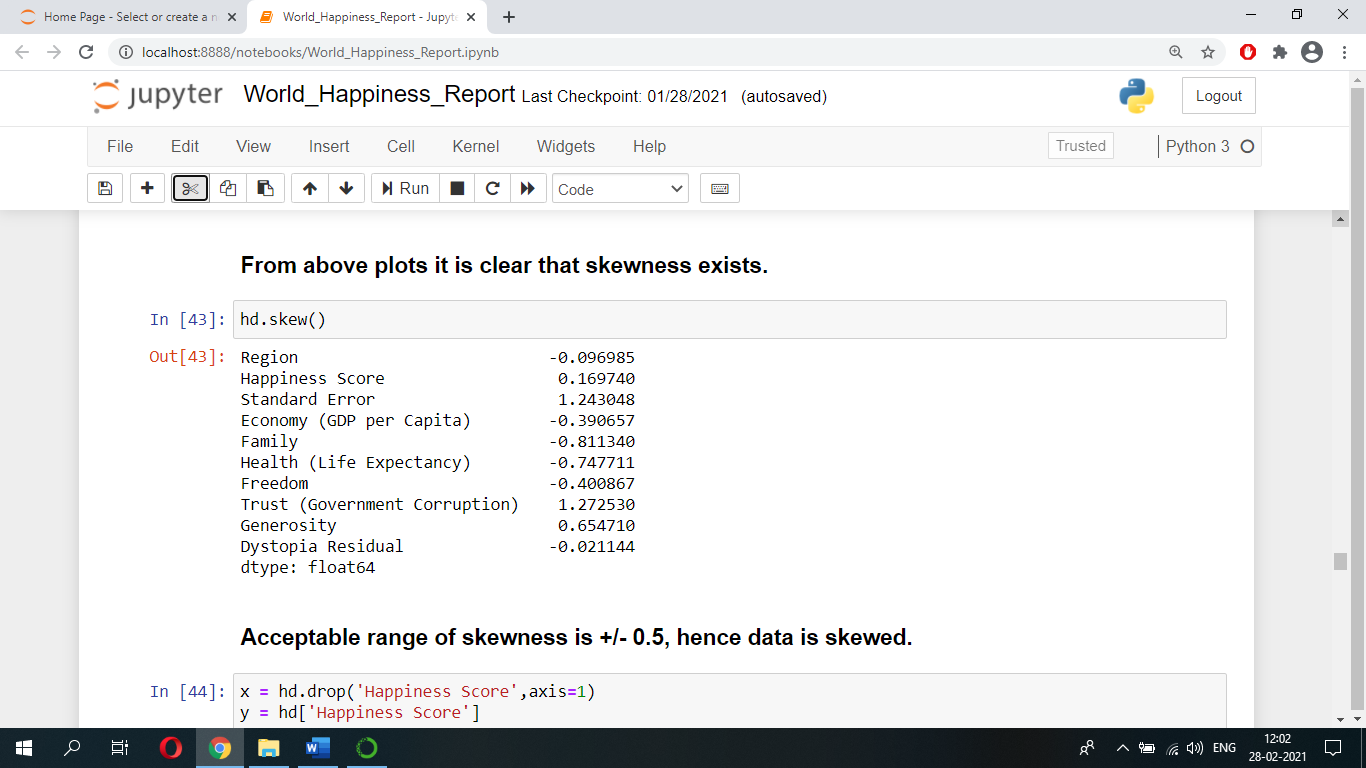
Now let’s check what is the percentage loss of data.



Since significant amount of data is not lost after removing outliers we can go ahead.

* **Checking Skewness**

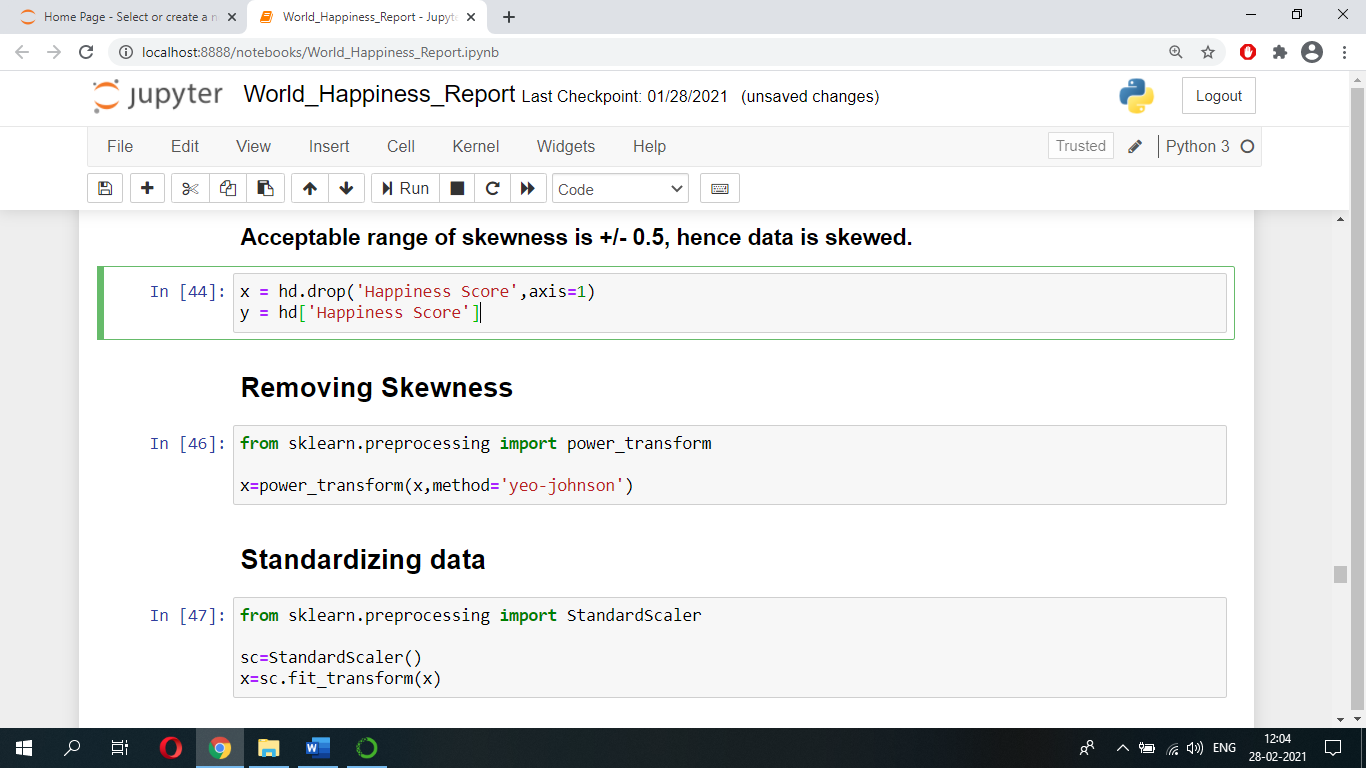
We can check whether data is skewed or not using histograms or we can use skew() function to check the same.



Acceptable range of skewness is +/- 0.5, hence from above it is clear that our data is skewed.

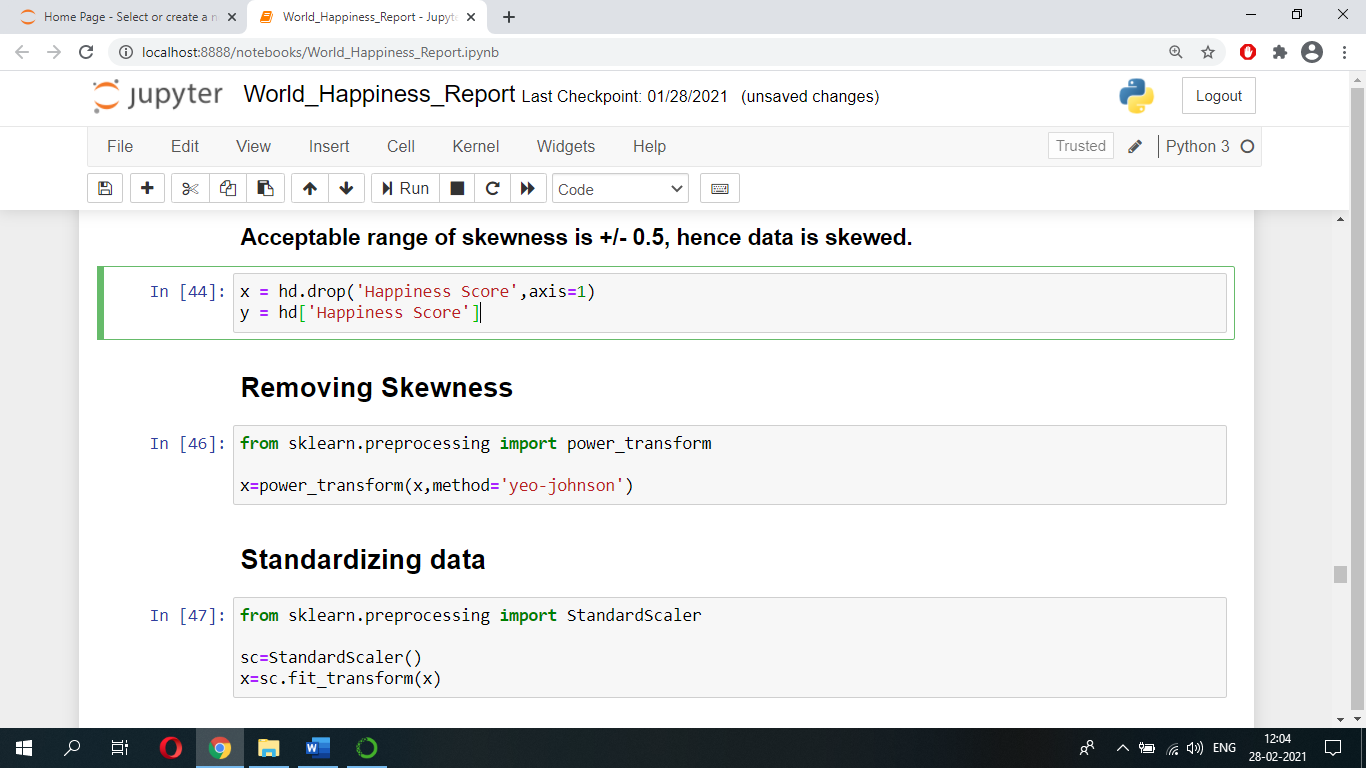
* **Splitting Data**

Let’s separate the label and features i.e., in order to predict ‘Happiness Score’ (Regression Problem) we need to separate ‘Happiness Score’ from rest of the features.



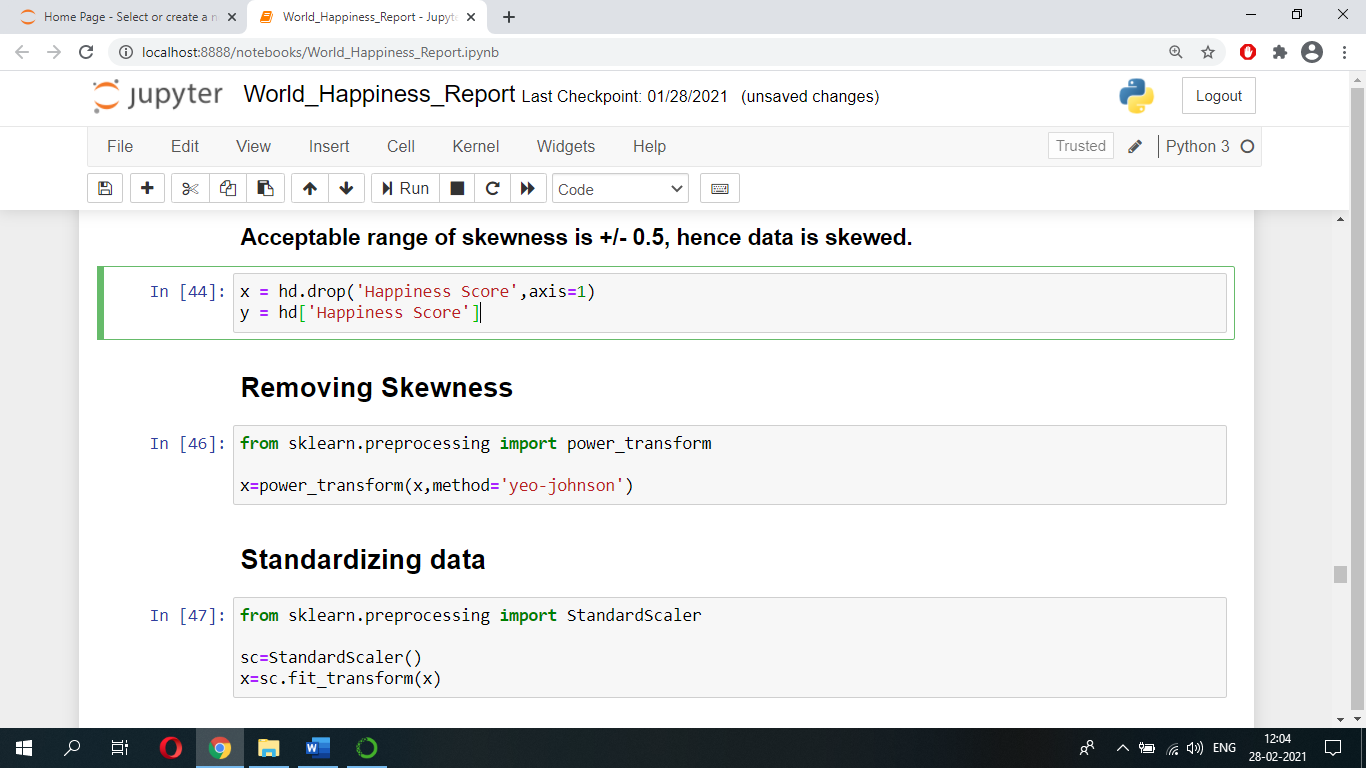
* **Removing Skewness**

Now let’s remove skewness by using ‘power transform’ function. We can import ‘power transform’ function from ‘sklearn.preprocessing’.



* **Standardizing Data**

Let’s standardize our data using’ StandardScaler’ function. We can import ‘StandardScaler’ function from ‘sklearn.preprocessing’.

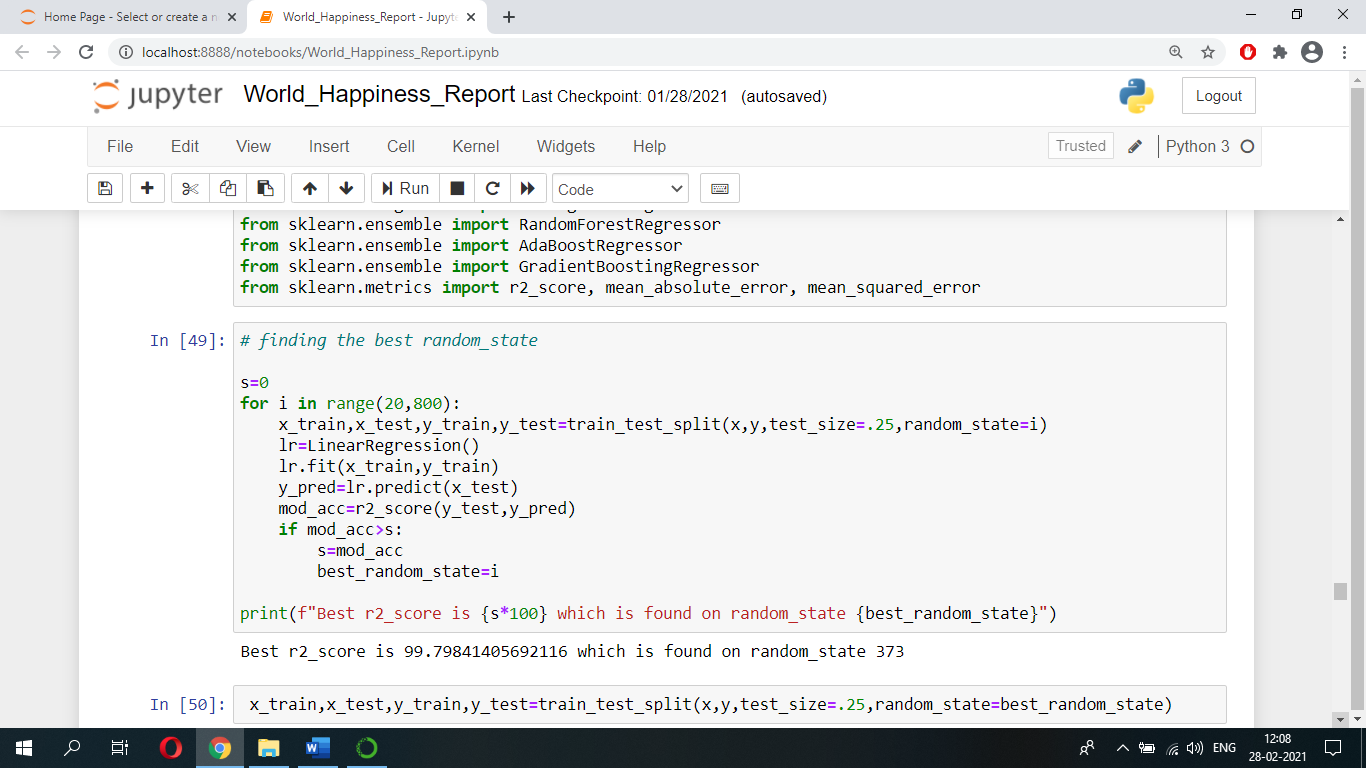


* **Model Training And Testing**
* **Let’s Predict ‘Happiness Score’ using Regression Models.**

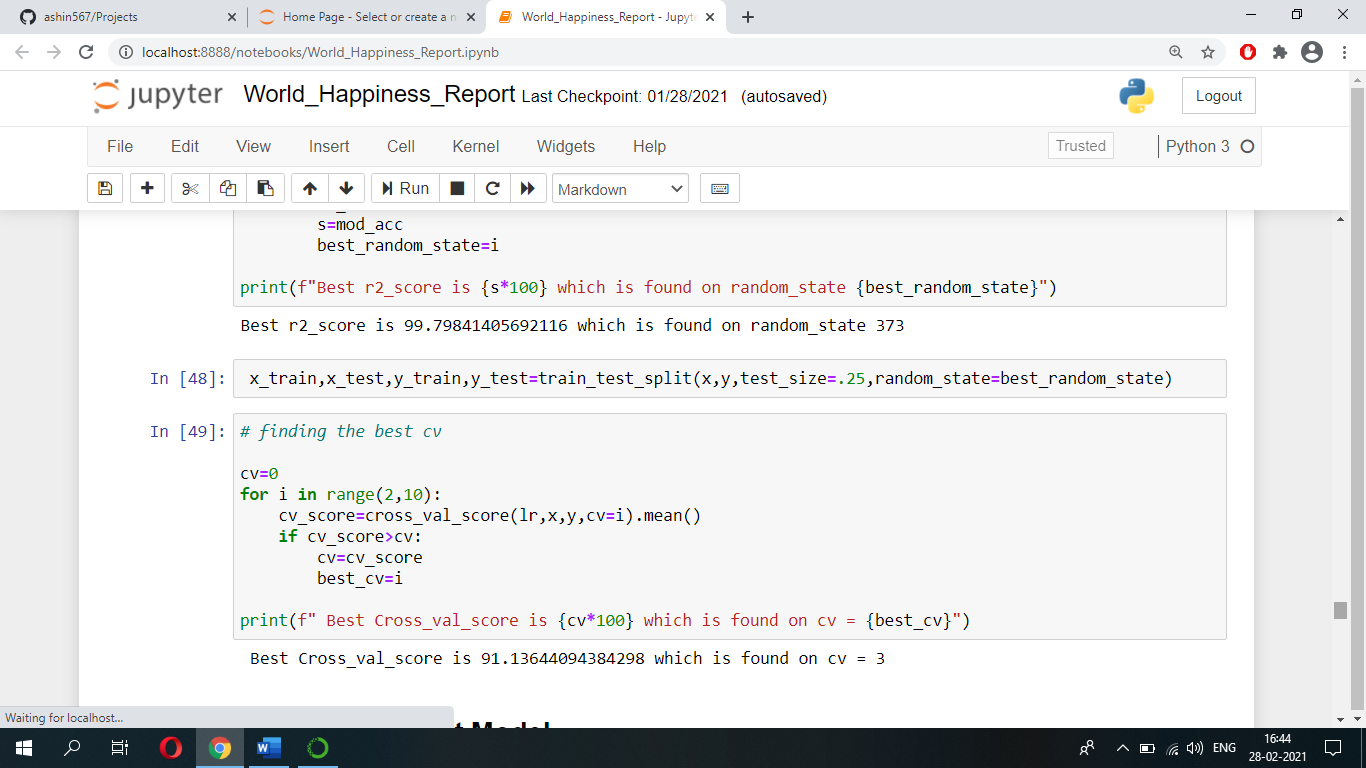
When using machine learning algorithms/models we should always split our data into a training set and testing set.

Let’s split our data into train set and test set using ‘train\_test\_split’. Let the test size be 25% and remaining 75% be train set.

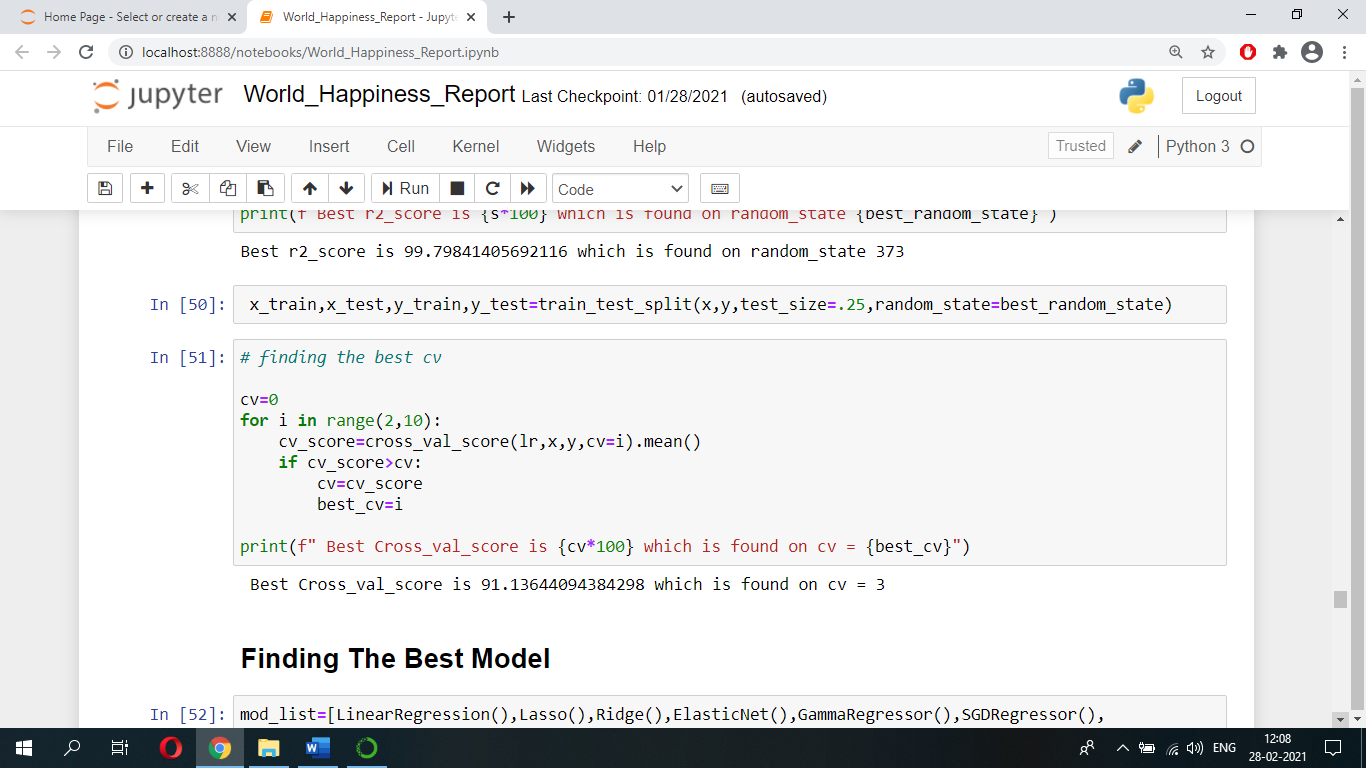
Below code has been used to find the best random state.



Below code has been used to find the best cv.



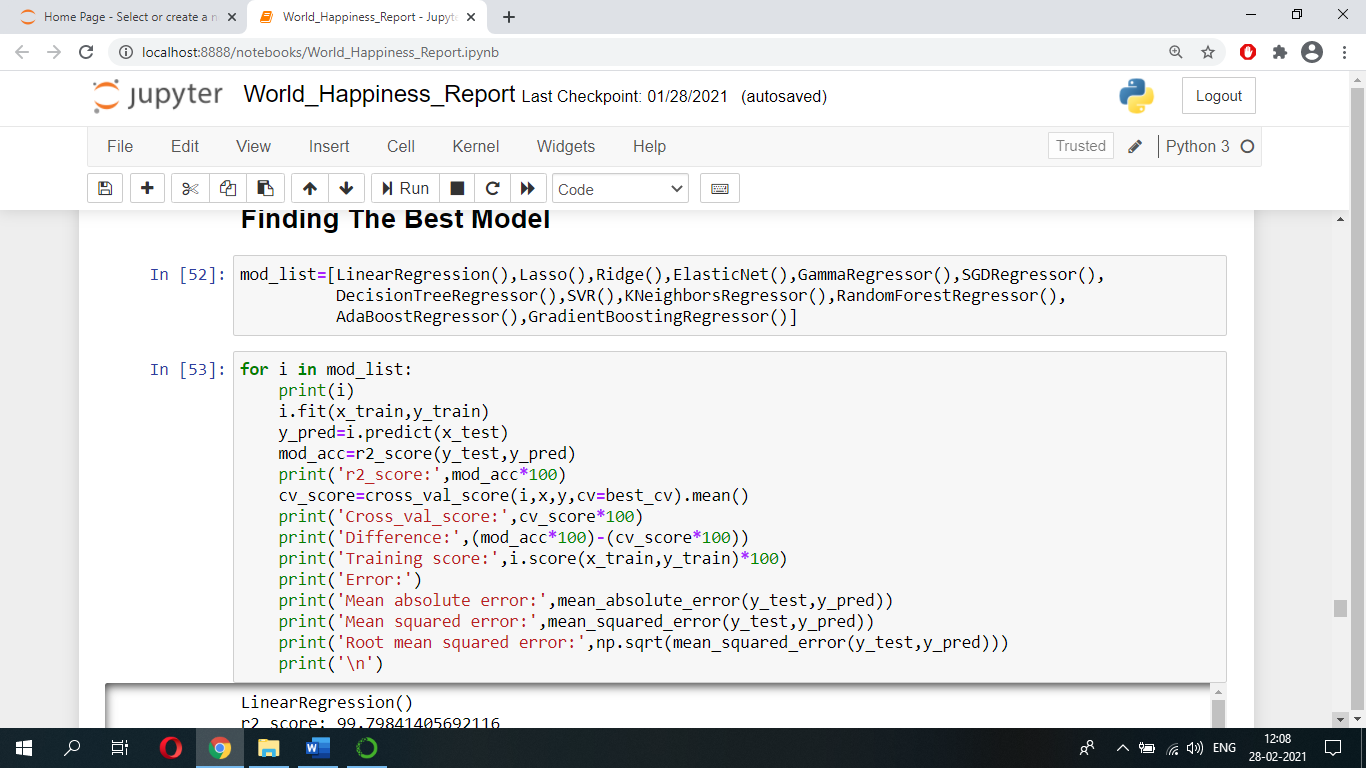
Then the best random state number will be used while splitting our data into train set and test set.



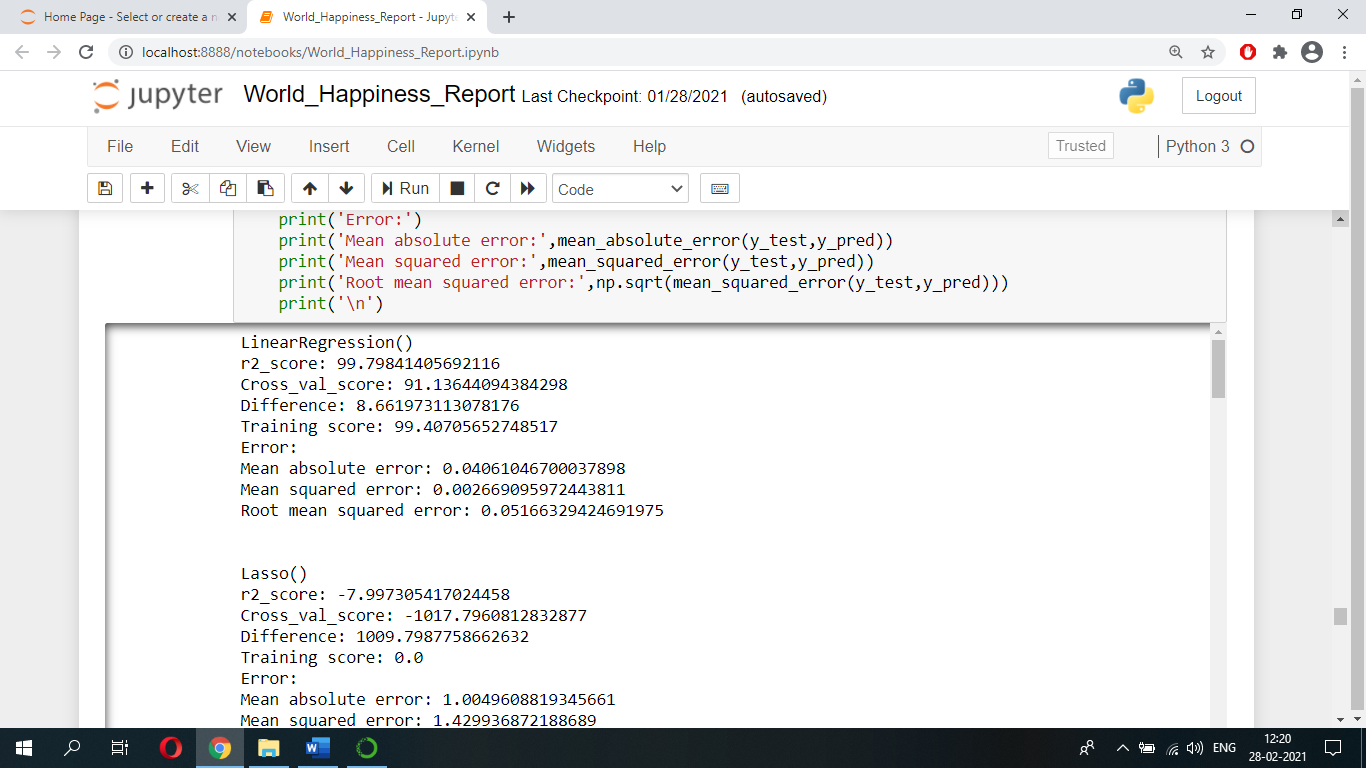
* **Finding The Best Model**

We can now train our model. We use the fit function to train the model.

Let’s run the code shown below and see which model gives us high ‘r2\_score’ and less Root mean squared error because model with high r2\_score and less Root mean squared error is a better performing model.

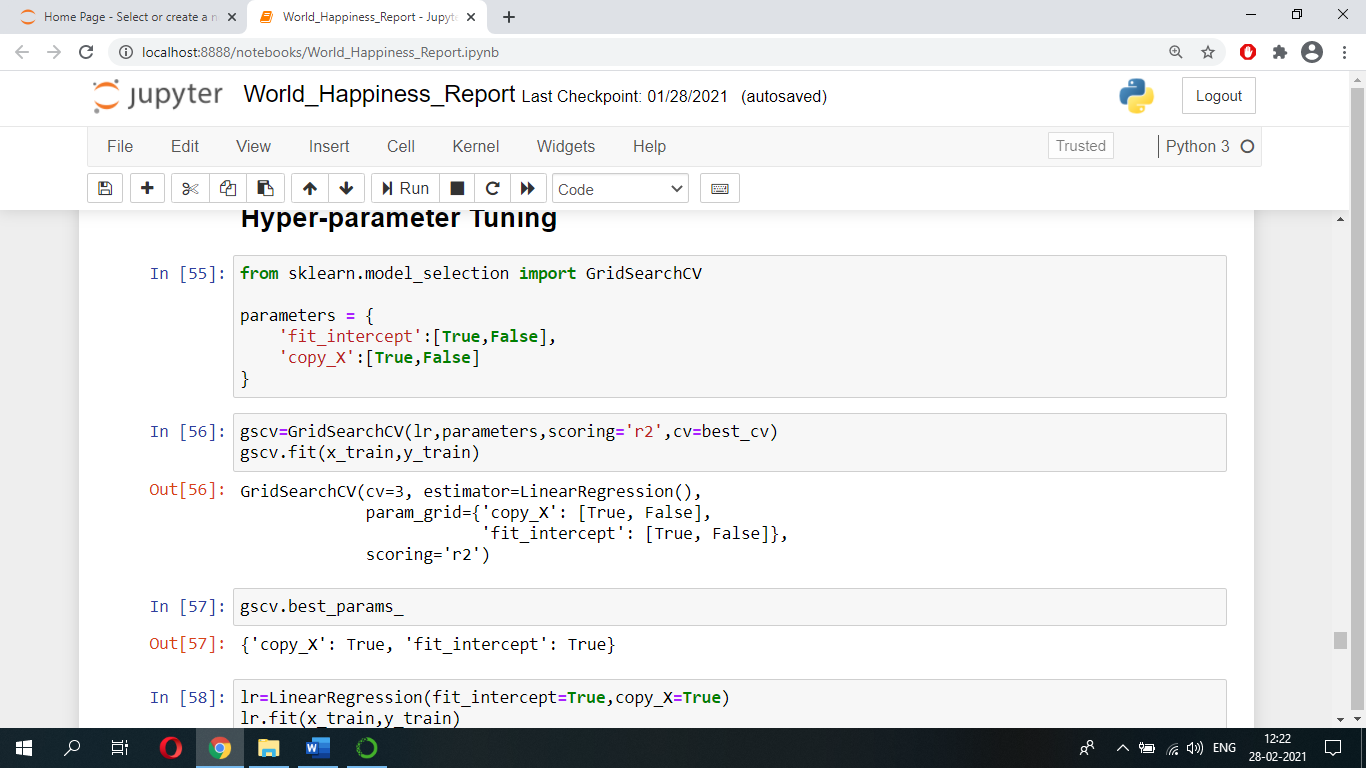


After running the code shown above, we came to know that ‘Linear Regression’ model is performing better with r2\_score of 98 to 99.8.

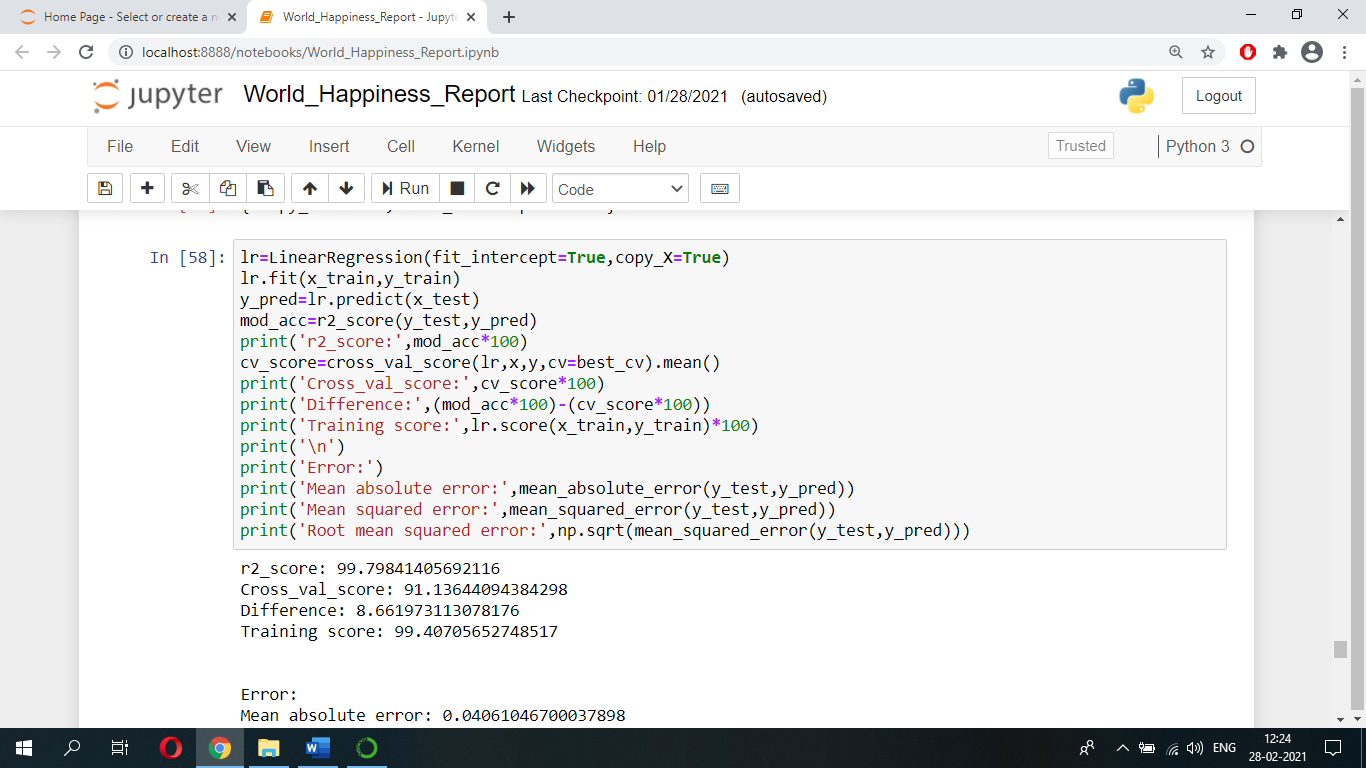


Let's perform hyper-parameter tuning on ‘Linear Regression’ model using GridSearchCV and find out which parameters of ‘Linear Regression’ model can be used so that model’s performance can be enhanced.

After running below code, we got these parameters which can be used on model to get better performance.

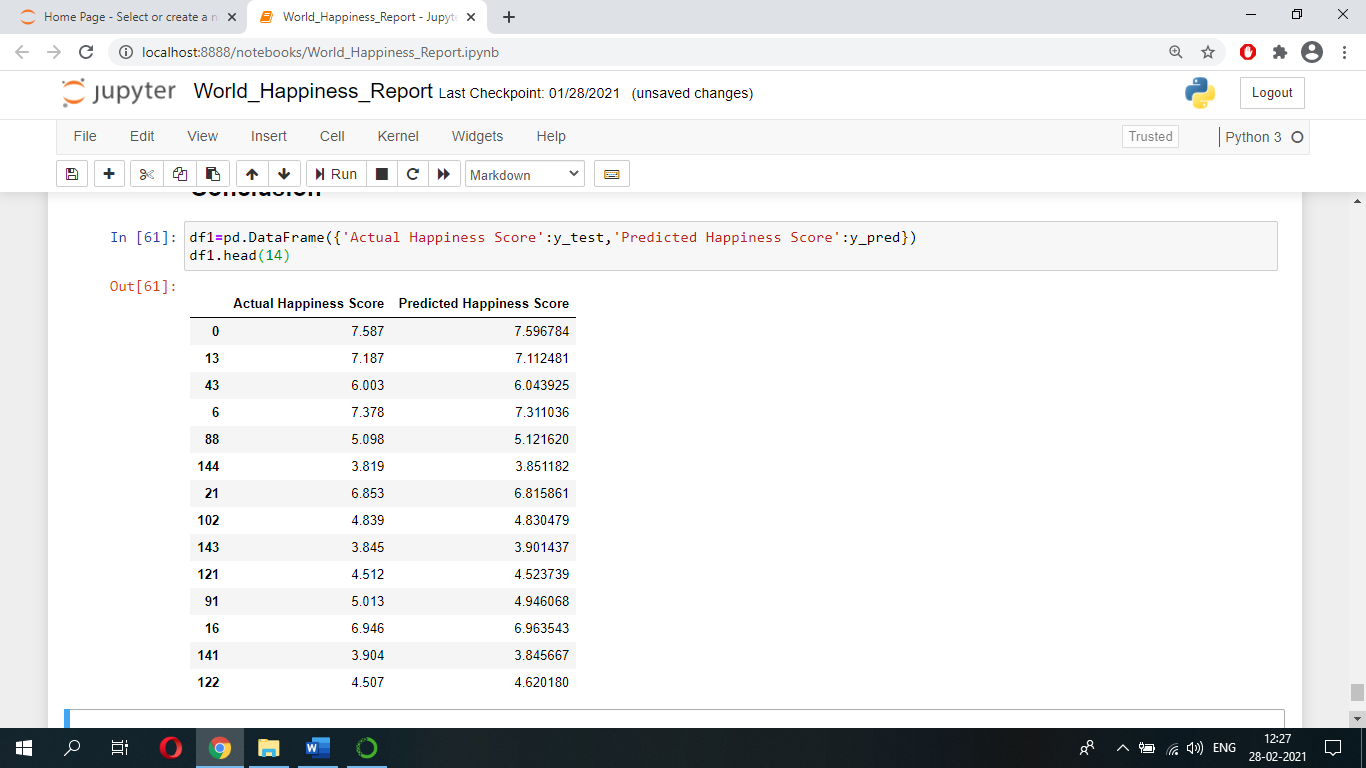


Let’s use these parameters and see what r2\_score we are going to get.

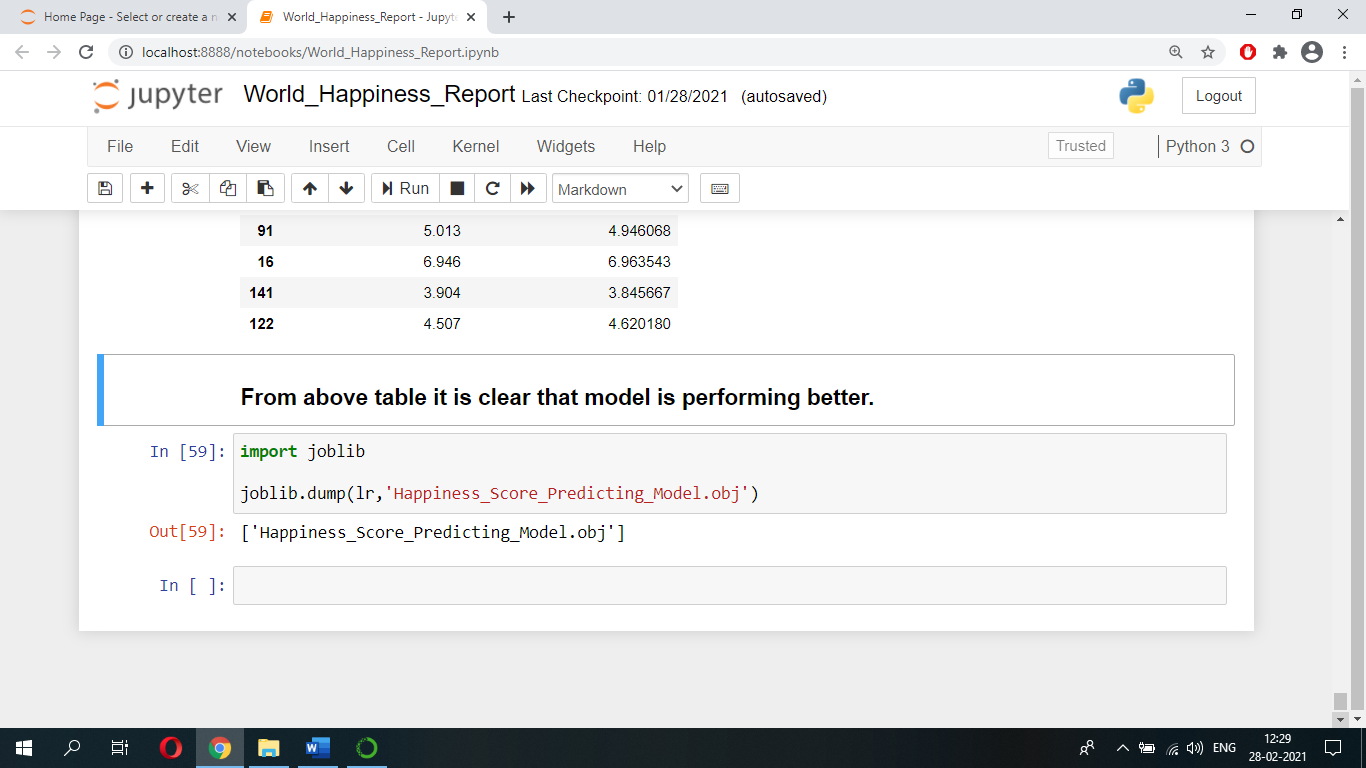


Finally, we got r2\_score of 99.79.

We can see how our model is performing i.e., predicting ‘Happiness Score’ from below table.



Lastly, let’s save this model for production and for future prediction using dump function of joblib.

****