Machine Learning

Machine Learning is an application of artificial intelligence(AI) that provides systems the ability to automatically learn and it uses statistical methods to enable machines to improve with an experience without being explicitly programmed.

Types of Machine Learning :

* Supervised Learning
* Unsupervised Learning
* Reinforcement Learning

**Supervised Learning :**

It is where we have input variables (x) and an output variables (y) and we use an algorithm to learn the mapping function from the input to the output. It is called supervised learning because the process of an algorithm learning from the training dataset can be thought as a teacher supervising the learning process.

Supervised learning problems are categorized into two types :

* Regression problems
* Classification problems

Example : House price prediction , Iris flower classification.

**Unsupervised Learning :**

Unsupervised learning is the training of machine using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance.The task of machine is to group unsorted information according to similarities, patterns and differences without any prior training of data.

Unsupervised learning problems are categorized into two types :

* Clustering problems
* Association rule problems

Example : Image classification , Market basket analysis

**Reinforcement Learning :**

Reinforcement learning is also a type of machine learning where an agent learns to behave in an environment by performing actions and seeing the results.

Example : Chess game

**1.Regression**:

In a regression problem we are trying to predict results within a continuous output, meaning that we are trying to map input variables to some continuous function.It is a set of statistical processes for estimating the relationship between a dependent variable and one or more independent variables.Example: revenue of a company or the temperature of an engine based on number of working hours.

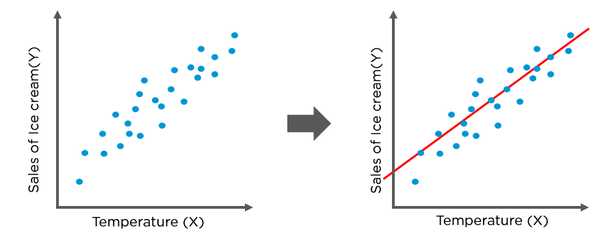
**Linear Regression :**

Linear regression is a regression model which is used for predicting the unknown value of a variable (Dependent variable) from the known value of another variables (Independent variable).A dependent variable is the variable to be predicted or explained in a regression model.An independent variable is the variable related to the dependent variablein a regression equation.



Examples:

* Sales of Ice cream and temperature are linearly related.In summer the sales will be very high(as shown in below figure), but in winter and rainy season the sales will be low.
* Depending on height and weight of a boy the body mass index will be measured.



**Equation of a Line :**

Equation of a straight-line can be represented by the **y** **=** **a** **+** **bx**.

- **y** indicates the Dependent variable

- **x** indicates the Independent variable.

- **b** is the slope it suggests how much the line rises for increase in x. If it’s a positive value it defines the line that has upward slope and the negative values defines that the line that slope is downward.

- **a** is known as the y-intercept it specifies the point where the line crosses the vertical y axis. It indicates the value of y when x = 0.

## **Assumption made in Linear Regression:**

* Formal assumptions includes

a.The regression model can be expressed in a linear way.

b.The expected mean error of the regression model is zero.

c.The variance of the errors is constant

* Linearity: The relationship between x and the mean of y is linear. Correlation should be present either positive or negative.
* Homoscedasticity: The variance of residual is the same for any value of x.
* Independence: Observations are independent of each other.
* Normality: For any fixed value of x and y is normally distributed.Probability distribution of error is normal.i.e. mean = 0 and standard deviation = 1.
* No Auto correlation: the errors should be independent; output should not affect the previous or next error.
* Multi collinearity exists: Independent variables are not independent with each other means independent variables has sum dependency between them. One independent variable affects another independent variable.

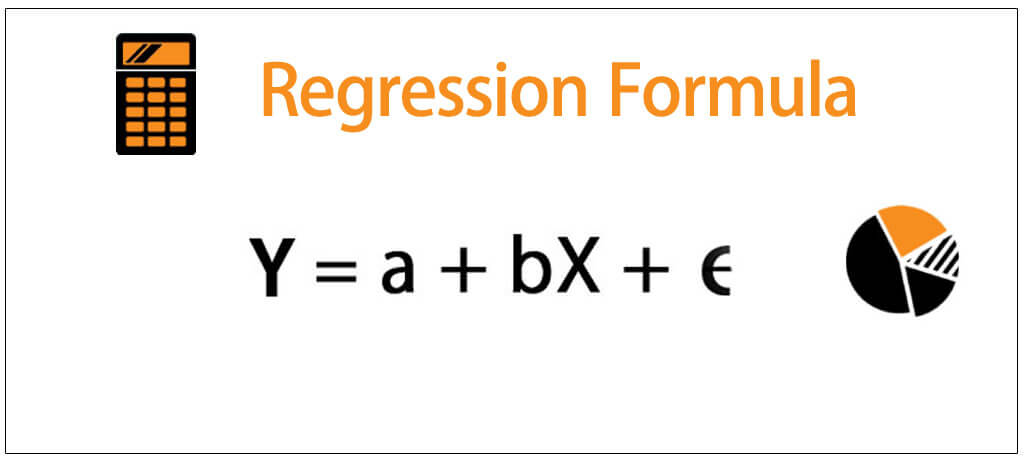
## **Types of Linear Regression:**

These are the important types of regressions -

1. Simple Linear Regression.
2. Multiple Linear Regression.
3. Polynomial Regression.

### **Simple Linear Regression:**

Simple linear regression is a linear regression model with a single explanatory variable. That is, it concerns two-dimensional sample points with one independent variable and one depedent variable (conventionally, the x and y coordinates in a Cartesian coordinate system) and finds a linear function (a non-vertical straight line) that, as accurately as possible, predicts the dependent variable values as a function of the independent variables.



Y = predicted value

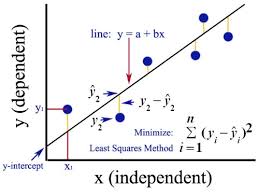
X = independent value

a = slope of the equation.

b = y-axis intercept

**\*Ordinary least squares :**

Ordinary least squareregression is a statistical method of analysis that estimates the relationship between one or more independent variables and a dependent variable; the method estimates the relationship by minimizing the sum of the squares in the difference between the observed and predicted values of the dependent variable configured as a straight line. In this entry, OLS regression will be discussed in the context of a bivariate model, that is, a model in which there is only one independent variable ( X ) predicting a dependent variable ( Y ).



In mathematical terms, the goal of OLS regression can be written as the task of minimizing the equation:



Here **yi** is the actual value and **ŷ**i is predicted value. This equation defines *e* (the error) as the difference between the actual *y* value and the predicted *y* value. The error values are squared and summed across all the points in the data. The solution for *a* depends on the value of *b*. It can be obtained using the formula.

**a**

****

– represents the mean of the independent variables

– represents the mean of dependent variable.

**b = - a**

If we break this equation apart into its component pieces, we can simplify the equation. The denominator for b should look familiar; it is very similar to the variance of x, which is denoted as Var(x), the variance involves finding the average squared deviation from the mean of *x*. This can be expressed as:



The numerator involves taking the sum of each data point's deviation from the mean *x* value multiplied by that point's deviation away from the mean *y* value. This is similar to the **covariance** function for *x* and *y*, denoted as *Cov(x, y)*. The covariance formula is:



If we divide the covariance function by the variance function, the n terms get cancelled and we can rewrite the formula for b as:



**a**

**Use Case :**

Study of relationship between monthly salary of a person based on his years of experience . The following table represents the salary of an employee on his years of experience.

|  |  |
| --- | --- |
| Years of experience | Salary of an employee |
| 1.1 | 39343 |
| 1.3 | 46205 |
| 1.5 | 37731 |
| 2 | 43525 |
| 2.2 | 39891 |
| 2.9 | 56642 |
| 3 | 60150 |
| 3.2 | 54445 |
| 3.2 | 64445 |
| 3.7 | 57189 |
| 3.9 | 63218 |
| 4 | 55794 |

**Solution :**

In the above used case we can observe that the dependent variable is salary of an employees and the independent variable is the years of experience.



The two variables are linearly correlated. i.e. the salary increases as the years of experience increases. So these two variables are linearly related with each other.

**Multiple Linear Regression :**

Multiple linear regression attempts to model the relationship between two or more explanatory variables and a response variable by fitting a linear equation to observed data. Every value of the independent variable *x* is associated with a value of the dependent variable *y*. The population regression line for *p* explanatory variables *x*1, *x*2, ... , *x*p is defined to be http://www.stat.yale.edu/Courses/1997-98/101/mu2.gify = http://www.stat.yale.edu/Courses/1997-98/101/beta.gif0 + http://www.stat.yale.edu/Courses/1997-98/101/beta.gif1*x*1 + http://www.stat.yale.edu/Courses/1997-98/101/beta.gif2*x*2 + ... + http://www.stat.yale.edu/Courses/1997-98/101/beta.gifp*x*p. This line describes how the mean response http://www.stat.yale.edu/Courses/1997-98/101/mu2.gify changes with the explanatory variables. The observed values for *y* vary about their means http://www.stat.yale.edu/Courses/1997-98/101/mu2.gify and are assumed to have the same standard deviation http://www.stat.yale.edu/Courses/1997-98/101/sigma2.gif. The fitted values *b0*, *b1*, ..., *bp* estimate the parameters http://www.stat.yale.edu/Courses/1997-98/101/beta.gif0, http://www.stat.yale.edu/Courses/1997-98/101/beta.gif1, ..., http://www.stat.yale.edu/Courses/1997-98/101/beta.gifp of the population regression line.

Since the observed values for *y* vary about their means http://www.stat.yale.edu/Courses/1997-98/101/mu2.gify, the multiple regression model includes a term for this variation. In words, the model is expressed as DATA = FIT + RESIDUAL, where the "FIT" term represents the expression http://www.stat.yale.edu/Courses/1997-98/101/beta.gif0 + http://www.stat.yale.edu/Courses/1997-98/101/beta.gif1*x*1 + http://www.stat.yale.edu/Courses/1997-98/101/beta.gif2*x*2 + ... http://www.stat.yale.edu/Courses/1997-98/101/beta.gifp*x*p. The "RESIDUAL" term represents the deviations of the observed values *y* from their means http://www.stat.yale.edu/Courses/1997-98/101/mu2.gify, which are normally distributed with mean 0 and variance http://www.stat.yale.edu/Courses/1997-98/101/sigma2.gif. The notation for the model deviations is http://www.stat.yale.edu/Courses/1997-98/101/eps.gif.

**Formally, the model for multiple linear regression, given *n* observations, is  
*y*i = http://www.stat.yale.edu/Courses/1997-98/101/beta.gif0 + http://www.stat.yale.edu/Courses/1997-98/101/beta.gif1*x*i1 + http://www.stat.yale.edu/Courses/1997-98/101/beta.gif2*x*i2 + ... http://www.stat.yale.edu/Courses/1997-98/101/beta.gifp*x*ip + http://www.stat.yale.edu/Courses/1997-98/101/eps.gifi for *i* = 1,2, ... *n*.**

In the least-squares model, the best-fitting line for the observed data is calculated by minimizing the sum of the squares of the vertical deviations from each data point to the line (if a point lies on the fitted line exactly, then its vertical deviation is 0). Because the deviations are first squared, then summed, there are no cancellations between positive and negative values. The least-squares estimates *b0*, *b1*, ... *bp* are usually computed by statistical software.

The values fit by the equation *b0* + *b1xi1* + ... + *bpxip* are denoted *http://www.stat.yale.edu/Courses/1997-98/101/yhat.gifi*, and the residuals *ei* are equal to *yi - http://www.stat.yale.edu/Courses/1997-98/101/yhat.gifi*, the difference between the observed and fitted values. The sum of the residuals is equal to zero.

The variance http://www.stat.yale.edu/Courses/1997-98/101/sigma2.gif² may be estimated by ***s*² = **, also known as the mean-squared error (or MSE).  
The estimate of the standard error *s* is the square root of the MSE.

**Use Case :**

The below dataset is on petrol consumption with various factors includes petrol tax, average income, paved highways so on have been given in the table. Fit a line and predict the petrol consumption based on the various factors.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Petrol\_tax | Average\_income | Paved\_Highways | Population\_Driver\_licence(%) | Petrol\_Consu-mption |
| 9 | 3571 | 1976 | 0.525 | 541 |
| 9 | 4092 | 1250 | 0.572 | 524 |
| 9 | 3865 | 1586 | 0.58 | 561 |
| 7.5 | 4870 | 2351 | 0.529 | 414 |
| 8 | 4399 | 431 | 0.544 | 410 |
| 10 | 5342 | 1333 | 0.571 | 457 |
| 8 | 5319 | 11868 | 0.451 | 344 |
| 8 | 5126 | 2138 | 0.553 | 467 |
| 8 | 4447 | 8577 | 0.529 | 464 |

Here the table represents the data with multiple linear regression. Here there are various independent variables and one dependent variable. The Petrol consumption are calculated based on the various features namely Petrol\_tax, Average\_income, Paved\_Highways, Population\_Driver\_licence(%). The equation of the plane would be:

**y = b +a1x1 + a2x2 + a3x3 + a4x4**

Here we fit a plane in a 4-dimension (Not imaginable by humans) space. We can further calculate the Petrol Consumption based on various features provided.

**Polynomial Regression:**

We regard polynomial regression as a generalized case of linear regression. In case of polynomial regression, we have the terms of multiple linear regression and in addition to those terms like a₁𝑥₁, the regression function ‘y’ can include non-linear terms such as a₂𝑥₁², a₃𝑥₁³, or even a₄𝑥₁𝑥₂, a₅𝑥₁²𝑥₂, and so on.

The simplest example of polynomial regression has only one independent variable, and the regression function ‘y’ is a polynomial of degree 2: y = 𝑏 + a₁𝑥 + a₂𝑥². Now, we have to calculate 𝑏, a₁, and a₂, which minimizes error. These are the known co-efficient of independent variables.

In the case of two variables and the polynomial of degree 2, the regression function has this form: y = 𝑏 + a₁𝑥₁ + a₂𝑥₂ + a₃𝑥₁² + a₄𝑥₁𝑥₂ + a₅𝑥₂². The procedure for solving the problem is identical to the previous case. You apply linear regression for five inputs: 𝑥₁, 𝑥₂, 𝑥₁², 𝑥₁𝑥₂, and 𝑥₂². We get the result of regression which are the values of six weights to minimize error: 𝑏, a₁, a₂, a₃, a₄, and a₅.

**Overfitting and Underfitting :**

**Overfitting :**

* Good performance on the training data, poor generliazation to other data.
* A statistical model is said to be overfitted, when we train it with a lot of data.
* When a model gets trained with so much of data, it starts learning from the noise and inaccurate data entries in our data set.
* Then the model does not categorize the data correctly, because of too much of details and noise.
* The causes of overfitting are the non-parametric and non-linear methods because these types of machine learning algorithms have more freedom in building the model based on the dataset and therefore they can really build unrealistic models.
* A solution to avoid overfitting is using a linear algorithm if we have linear data or using the parameters like cross-validation.

**Underfitting :**

* Poor performance on the training data and poor generalization to other data.
* A statistical model or a machine learning algorithm is said to have underfitting when it cannot capture the underlying trend of the data.
* Underfitting destroys the accuracy of our machine learning model. Its occurrence simply means that our model or the algorithm does not fit the data well enough.
* It usually happens when we have less data to build an accurate model and also when we try to build a linear model with a non-linear data.
* In such cases the rules of the machine learning model are too easy and flexible to be applied on such a minimal data and therefore the model will probably make a lot of wrong predictions.
* Underfitting can be avoided by using more data and also reducing the features by feature selection.

**Implementing the regression model with Python:**

The model can be built using Python with the help of few libraries also by applying proper packages and functions and classes.

Packages used to build the linear regression model:

**Numpy:**

It deals with the numerical python. It allows high performance operations on single and multi-dimensional arrays. Many mathematical computations can be performed using this library.

**Pandas:**

Pandas deals with the data or the data frame. It allows various operations to be performed with the data.

**Matplotlib:**

This library helps to plot various graphs, charts etc.

**Seaborn:**

This library focuses on visuals of statistical models.

**Scikit-learn:**

This library provides various models of regression for machine learning implementation.

**Linear Regression with Python:**

**Problem Statement – predicting medical expenses using linear regression**

Q. The 50\_startups.csv file includes 50 examples of various expenditures and profit of the startups. These dataset contains various features like R&D Spend, Administration, Marketing Spend and Profit. The features identify or predict the Profits of the startup. The features are:

• **R&D Spend**: It defines the amount of money spent on Research and Development of the startup.

• **Administration**: The amount spent on administration.

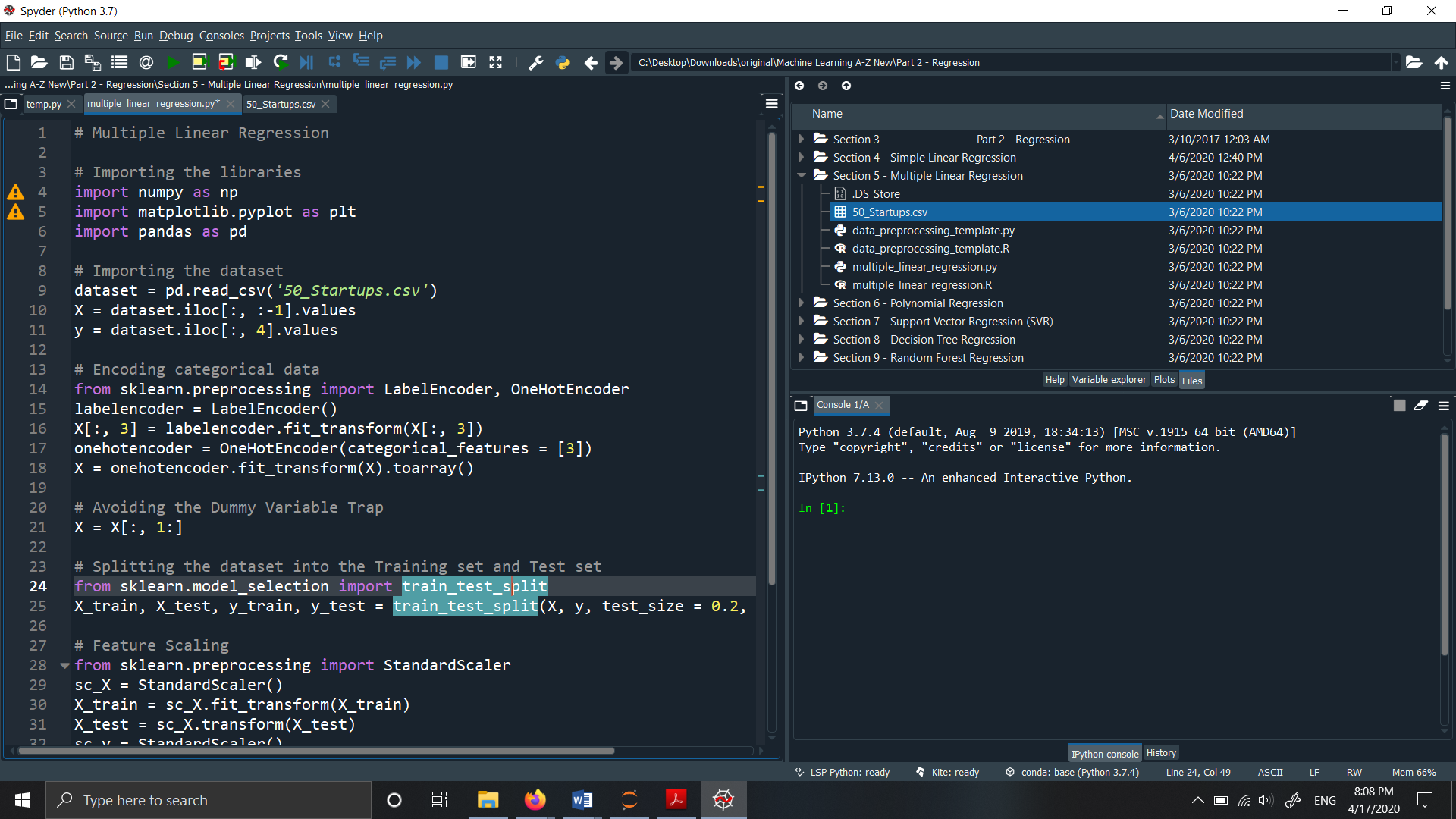
• **Marketing Spend**: Amount spent on marketing.

• **Profit:** The profits made by the various startups.

The steps for creating the model are as below:

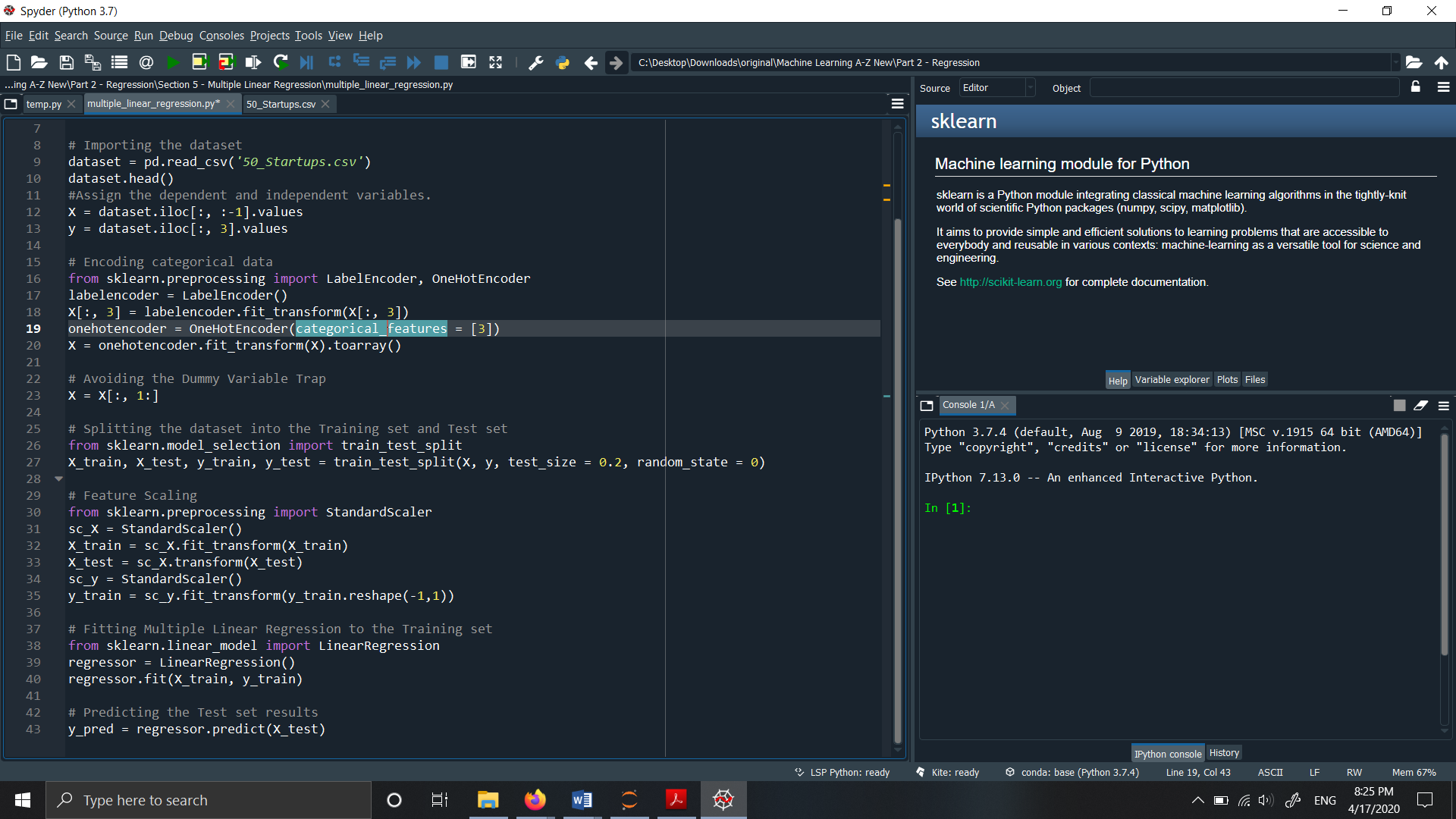
**Step1: Import all the necessary packages and methods from the library:**

* Import the necessary libraries like numpy, pandas, matplotlib.

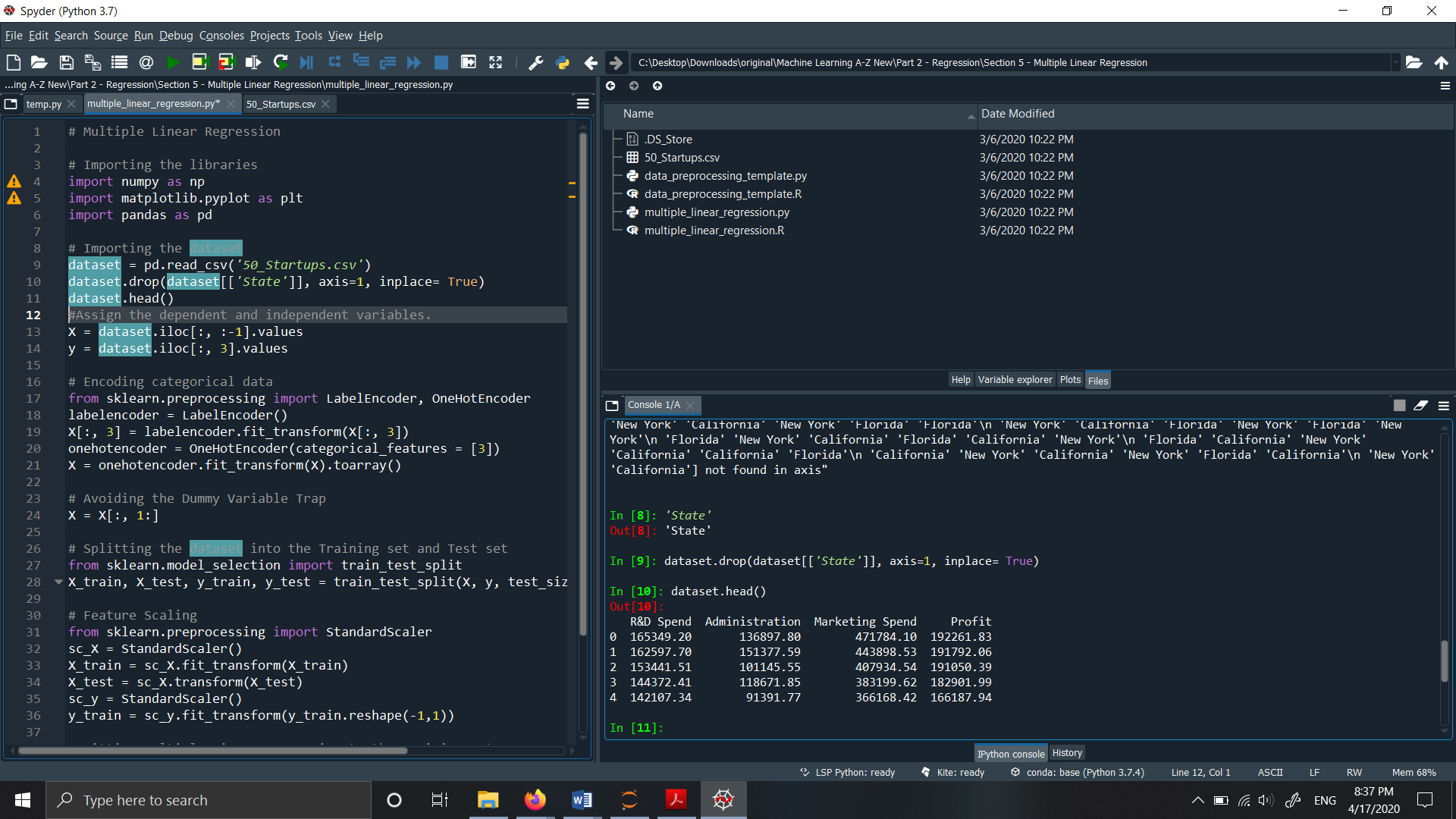


**Step 2: Import the data and assign the independent variables to X and dependent variables to y.**

* Import the data using pd.read\_csv(“File Name”) and load it to the dataset variable.
* Use the command dataset.head() to look at the top 5 observations in the dataset.
* Also we assign the independent variables to the X variable and the dependent ones to the y variable. We clearly know that the last column in the data-frame(Profit) is the dependent variable and rest all of them are independent variable.



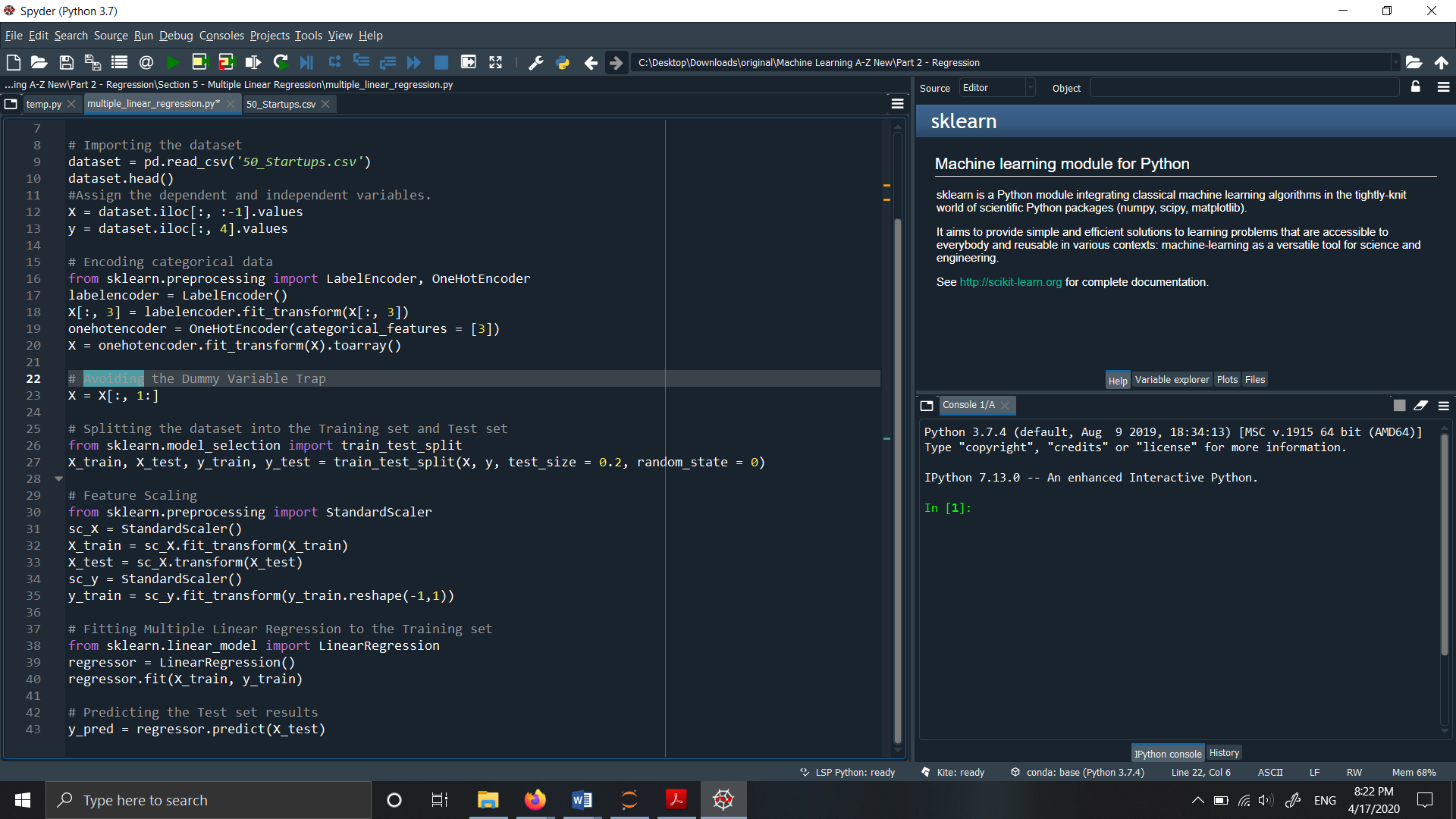
**Output:**



**Step 3: Split the data as training set and test data.**

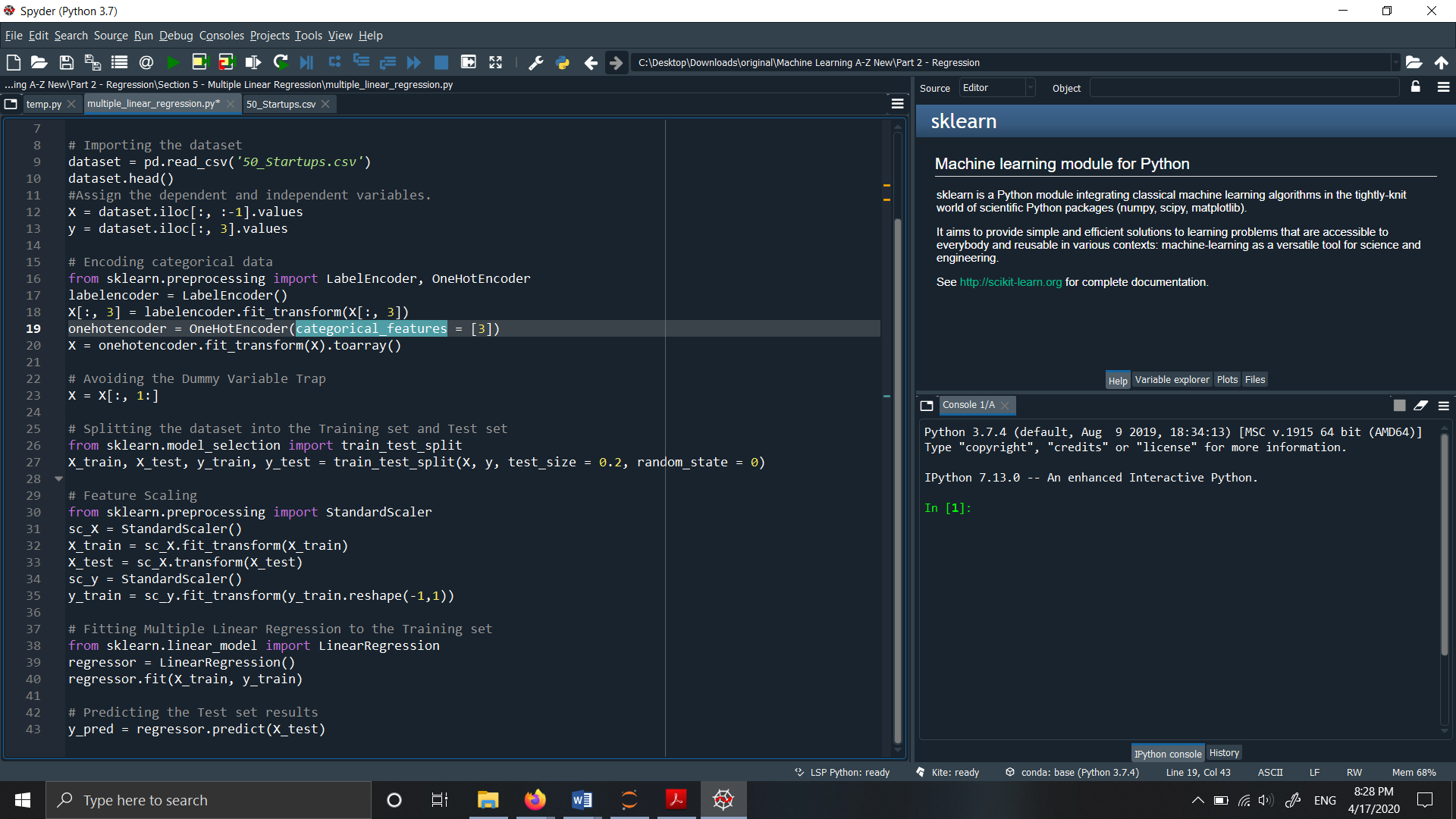
In this step the data is split into training and test data.

* Training data is used to train the model and test data is used to evaluate the model.
* test\_size specifies the percentage of data in test set.
* Different Random states shuffle the order of the observations in different ways. Specifying value to random\_state helps to reproduce the same shuffled observation everytime.



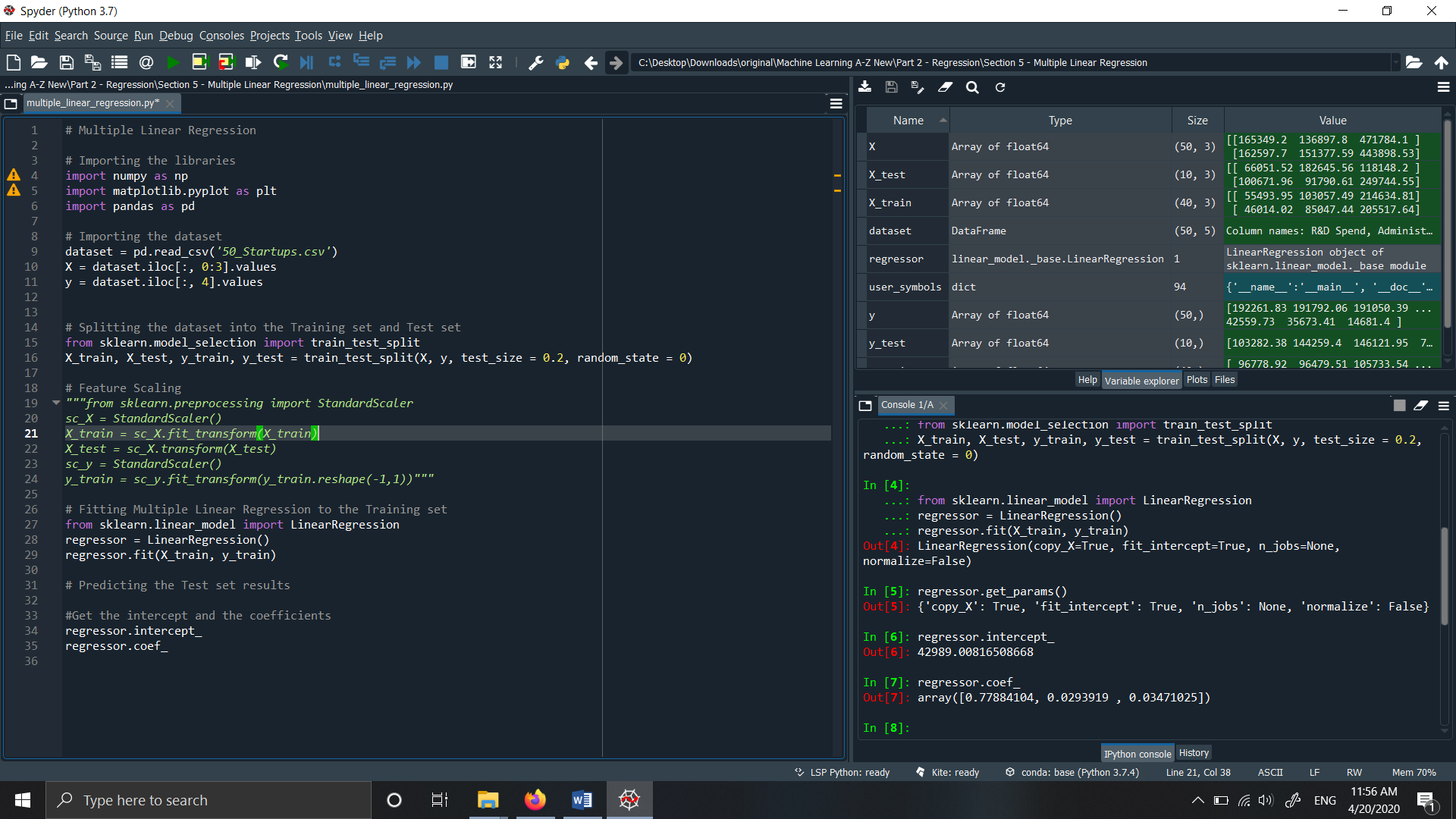
**Step 4: Fit the model to the training Set:**

* The model is fit to the training set. LinearRegression object is called from the sklearn.linear\_model library.
* Fit the training data- X\_train and the test data y\_train to the model ‘regressor’ .

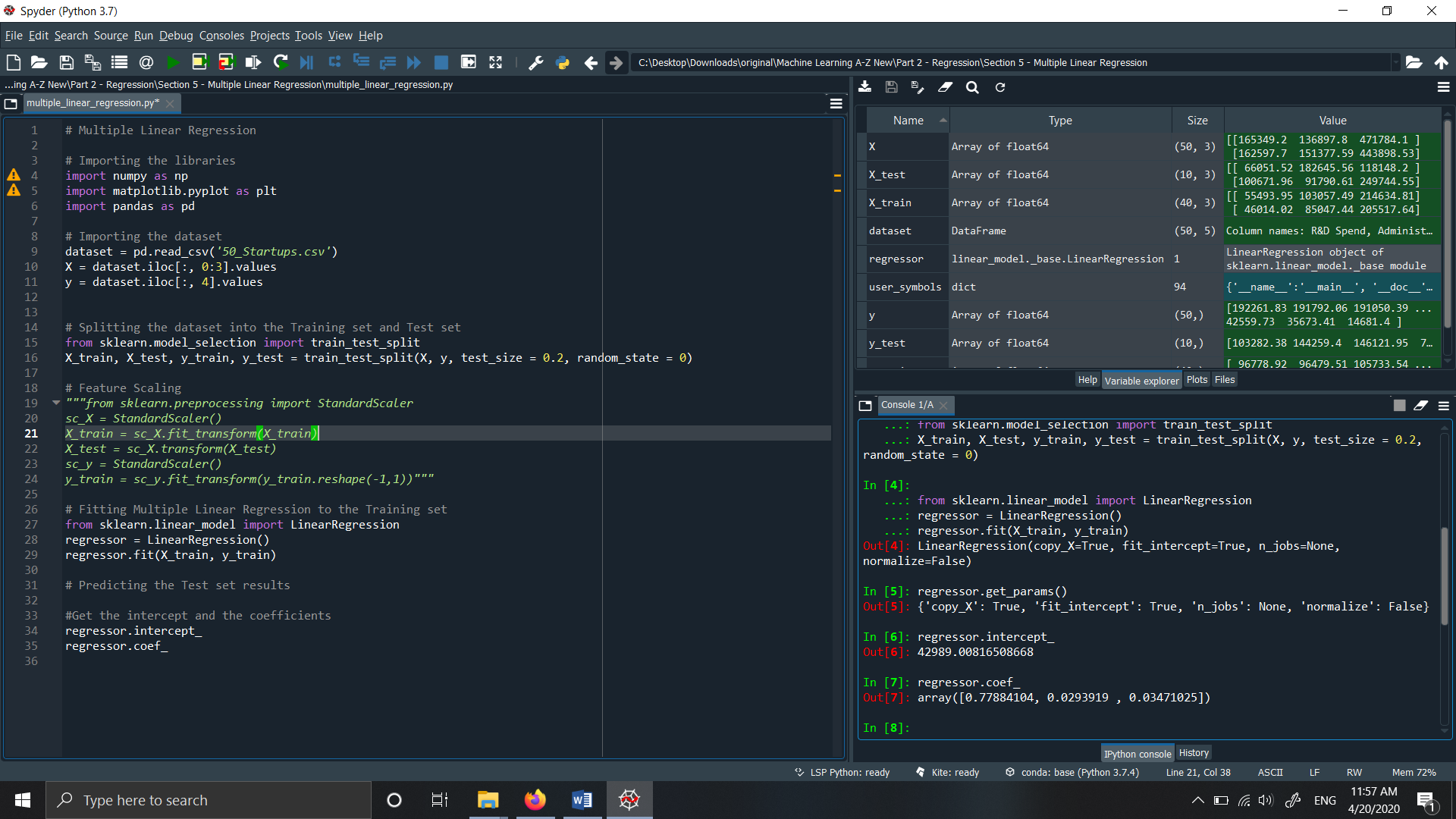


**Step 5: Get the intercept(b) and the coefficient of independent variables (a1, a2, a3)**

* Use .intercept\_ and .coef\_ to find the intercept values.



**Output:**



Here, b = 42989.00816508668

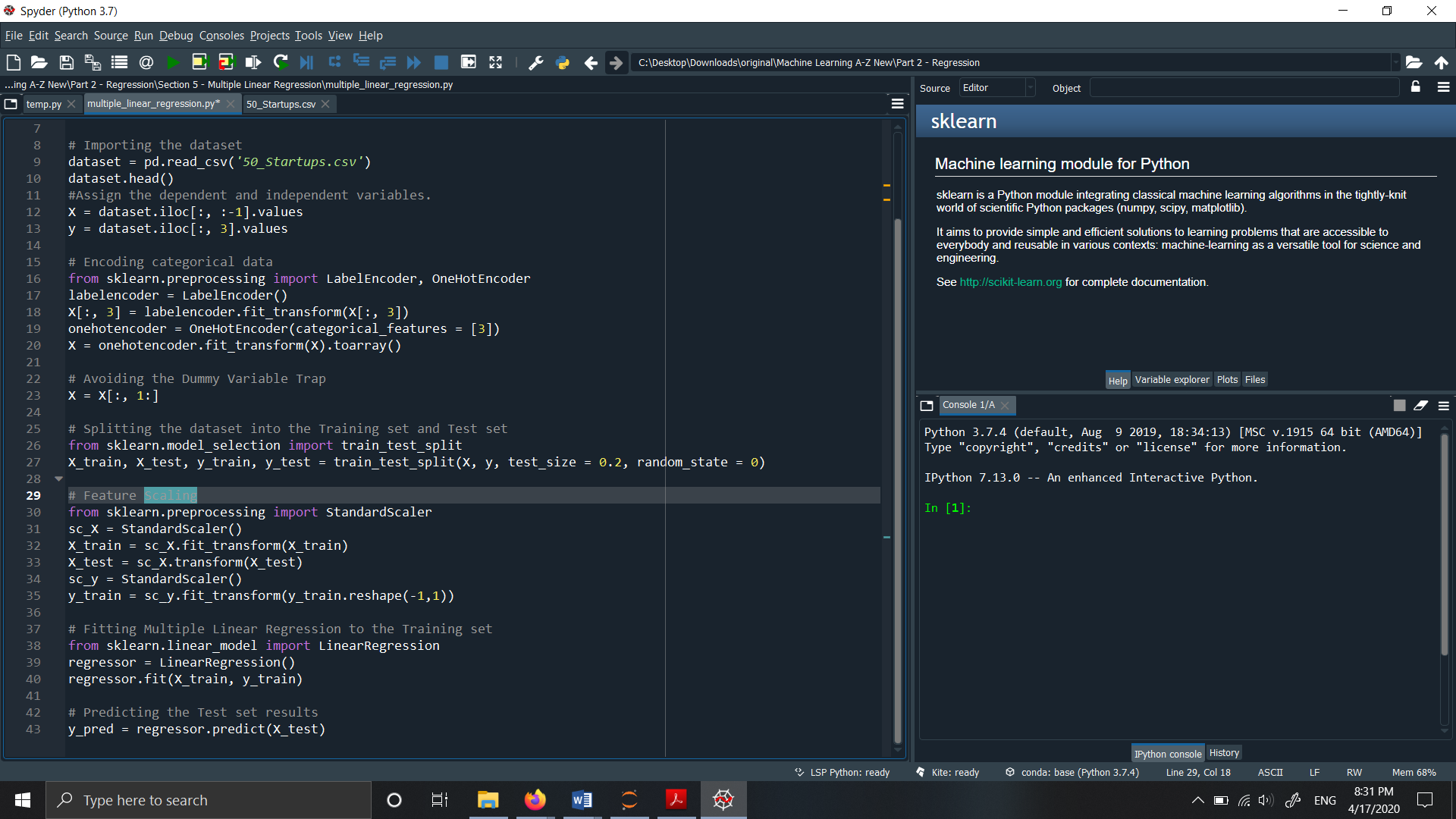
a1 = 0.77884104

a2 = 0.0293919

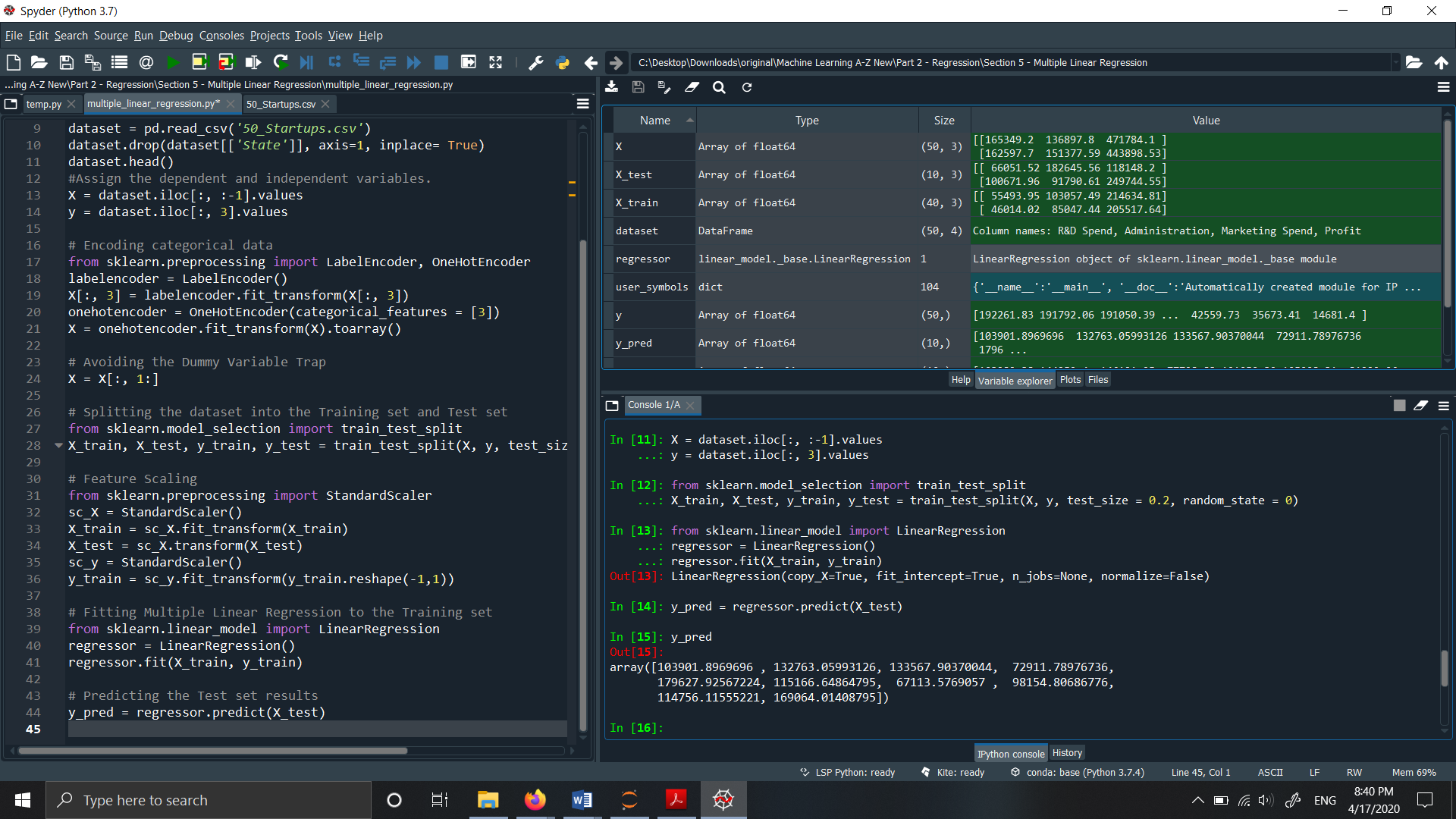
a3 = 0.03471025

**Step 6: Predict the output for the test set:**

* In this step the model predicts the result for the test set which was earlier split.

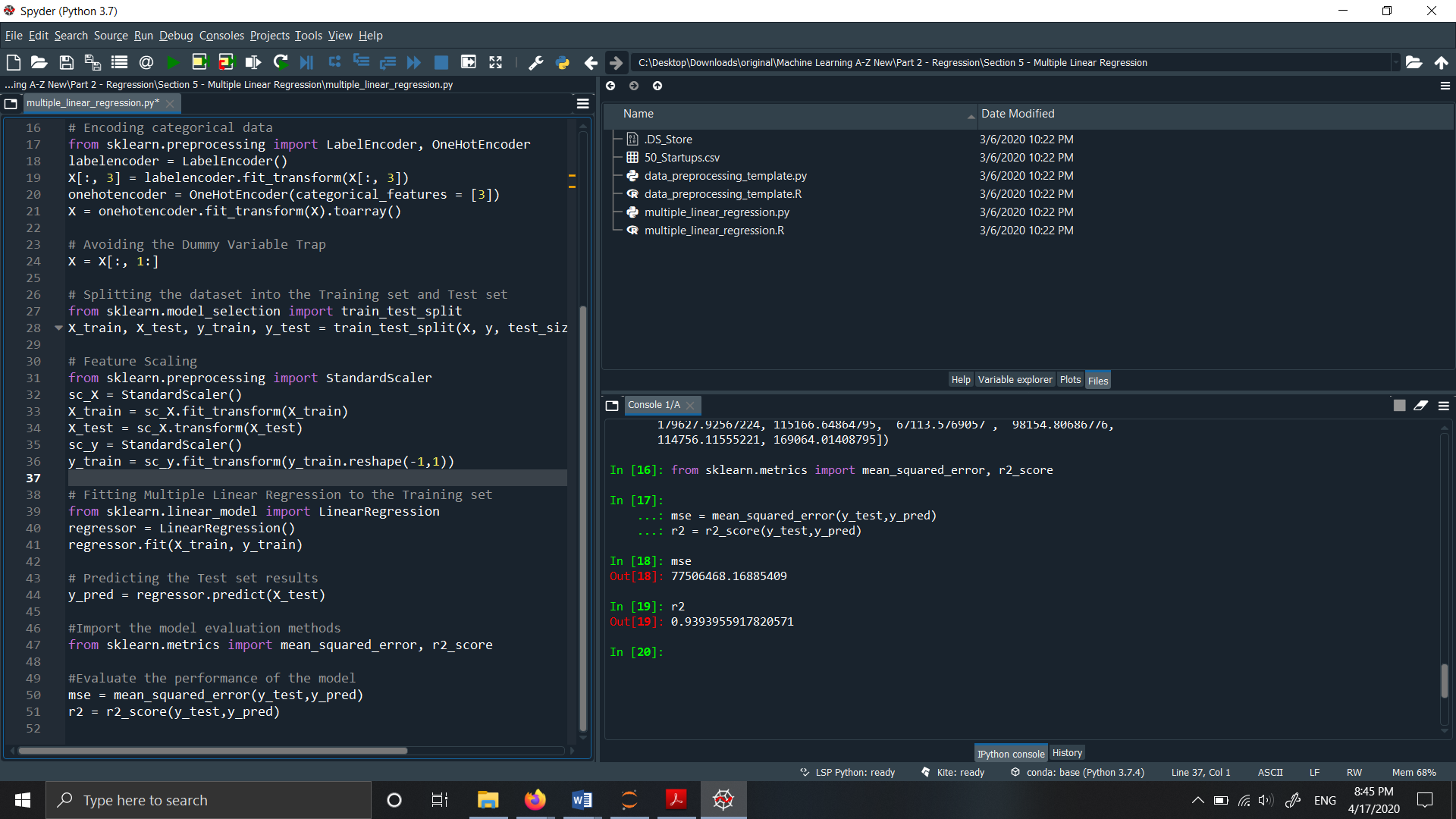


Output: The output of the predicted y\_pred will be an array of values.

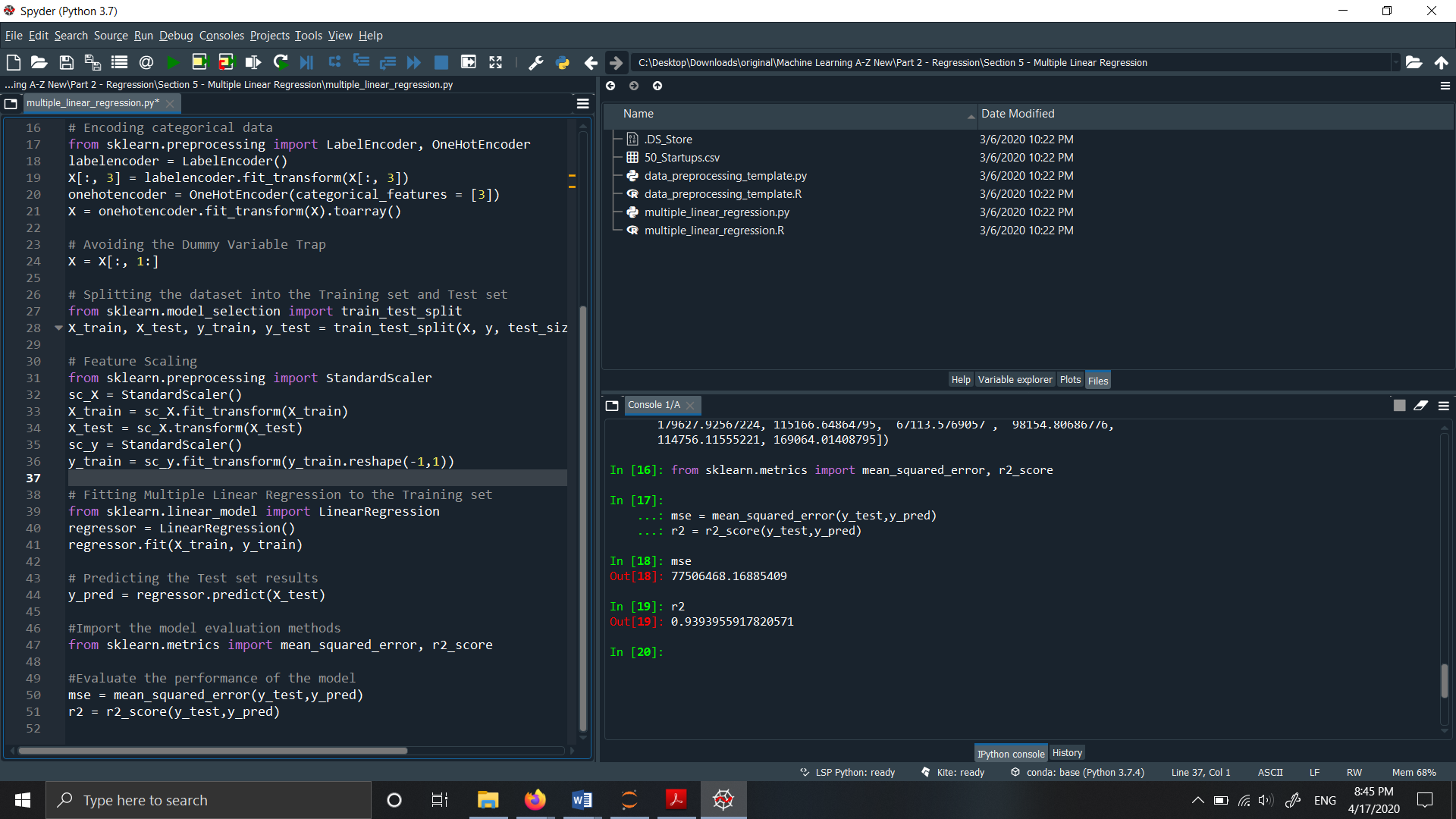


**Step 7: Evaluate the model performance:**

* The mean squared error and the r2 score is calculated for the predicted values and the y\_test values.



**Output :**



Here we have obtained the mean squared error and the R squared value. We have to reduce the mean squared error and we have obtained a very good R-squared value which shows that the model is fitting most of the data.

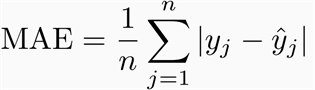
That’s the prediction using a linear regression model.

**Evaluation Metrics:**

The evaluation of the performance of the model is done by using these metrics:

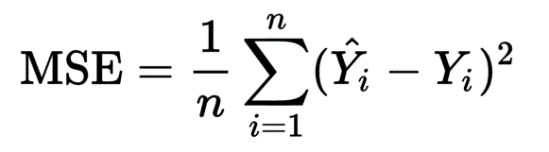
* Mean Absolute Error:

It is summation of differences between the actual and the predicted values. Lower the value better is the performance of the model.



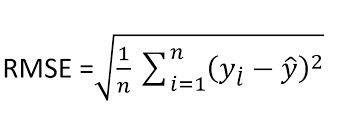
* + Mean Squared Error:

It is the square of sum summation of differences between the actual and the predicted values. Lower the value better is the performance of the model.



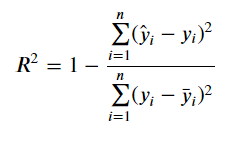
* + Root Mean Squared Error:

It is the squared root of Mean Squared Error.



* + R-Squared score or R2 score:

It ranges between 0 to +1. Higher the value better is the performance of the model.



**Advantages and Disadvantages of Linear Regression:**

**Advantages:**  
1. Linear Regression performs well when the given dataset is linearly separable. We can find the nature of the relationship among the different variables in the dataset.  
  
2. Linear Regression is very easy to build, implement, interpret and very efficient to train.   
  
3. Overfitting can happen in linear regression but it can be easily avoided by using some dimensionality reduction techniques, regularization (L1 and L2) techniques and cross-validation.  
  
**Disadvantages:**   
1. The main disadvantage of Linear Regression is the assumption of linearity between the dependent variable and the independent variables. In the real world, most of the times, the data is linearly inseparable. It assumes that there is a straight-line relationship between the dependent and independent variables which is incorrect many times.  
  
2. Linear Regression is prone to noise and overfitting: If the number of observations are lesser, Linear Regression may lead to overfit because is starts considering noise while building the model.  
  
3. Outliers: Linear regression is very sensitive to outliers. So, outliers should be resolved or removed before applying Linear Regression to the dataset.  
  
4. Prone to multicollinearity: Before applying Linear regression, multicollinearity should be removed (using dimensionality reduction techniques) because it assumes that there is no relationship among independent variables.

**2.Classification :**

In this problem ,we are instead trying to predict results in a discrete output.In other words, we are trying to map input variables into discrete categories. A classification problem is when the output variable is a category, such as “red” or “blue” or “disease” and “no disease”. A classification model attempts to draw some conclusion from observed values. Given one or more inputs a classification model will try to predict the value of one or more outcomes.

For example, when filtering emails “spam” or “not spam”, when looking at transaction data, “fraudulent”, or “authorized”. In short Classification either predicts categorical class labels or classifies data (construct a model) based on the training set and the values (class labels) in classifying attributes and uses it in classifying new data. There are a number of classification models. Classification models include logistic regression, decision tree, random forest, gradient-boosted tree, multilayer perceptron, one-vs-rest, and Naive Bayes.

**K Nearest Neighbors :**

Definition : The KNN algorithm is simplest supervised machine learning algorithm mostly used for classification.It classifies a data point based on how its neighbors are classified.

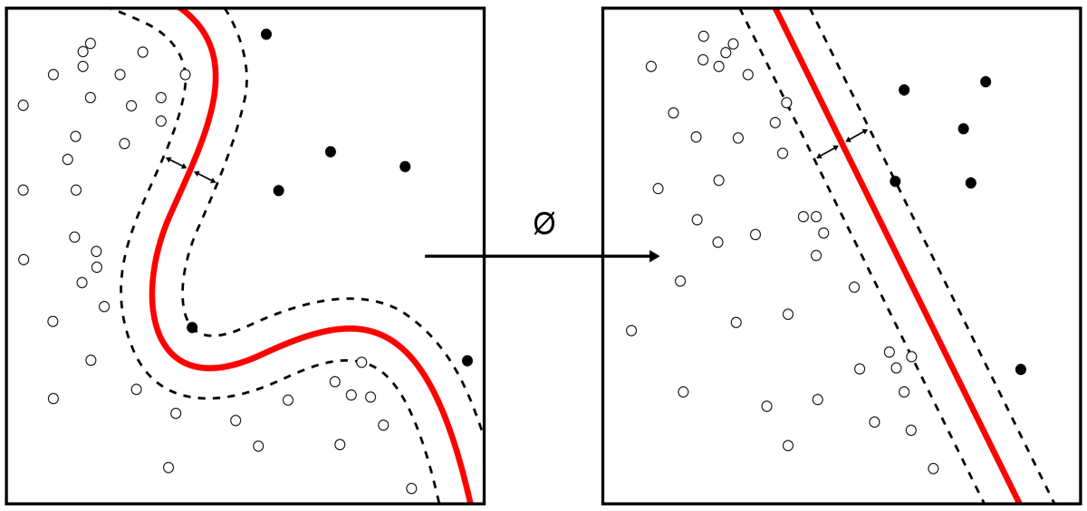
What KNN is :

* Implicitly constructs the decision boundaries.
* The goal is to classify an unknown training sample into one of C classes.
* KNN is both Linearly separable and Non linearly separable data.
* KNN captures the idea of similarity (sometimes called distance,proximity).It is based on feature similarity.
* KNN algorithm assumes that similar things exist in class proximity.Therefore,similar things are close to each other.

KNN algorithm works pretty well with a small number of input variables (p), but there are more chances of error in prediction when the number of inputs becomes very large.

**The KNN only requires**

* An integer K
* A set of labeled examples (training data)
* A metric to measure “closeness”



**Notation of KNN:**

Training samples : (x1,y1) , (x2,y2) , (x3,y3)………….(xn,yn)

* **x** is the feature vector with n number of features.
* Whereas ,**y** is a class label of {1,2,3………n}.

Goal is to determine **Y(new)** for **X(new)** usingdistance metrics **.**

**Example:**



**Why KNN**

KNN has some nice properties: it is automatically non-linear, it can detect linear or non-linear distributed data, it tends to perform very well with a lot of data points. On the minus side KNN needs to be carefully tuned, the choice of K and the metric (distance) to be used are critical.

**Use Case :**

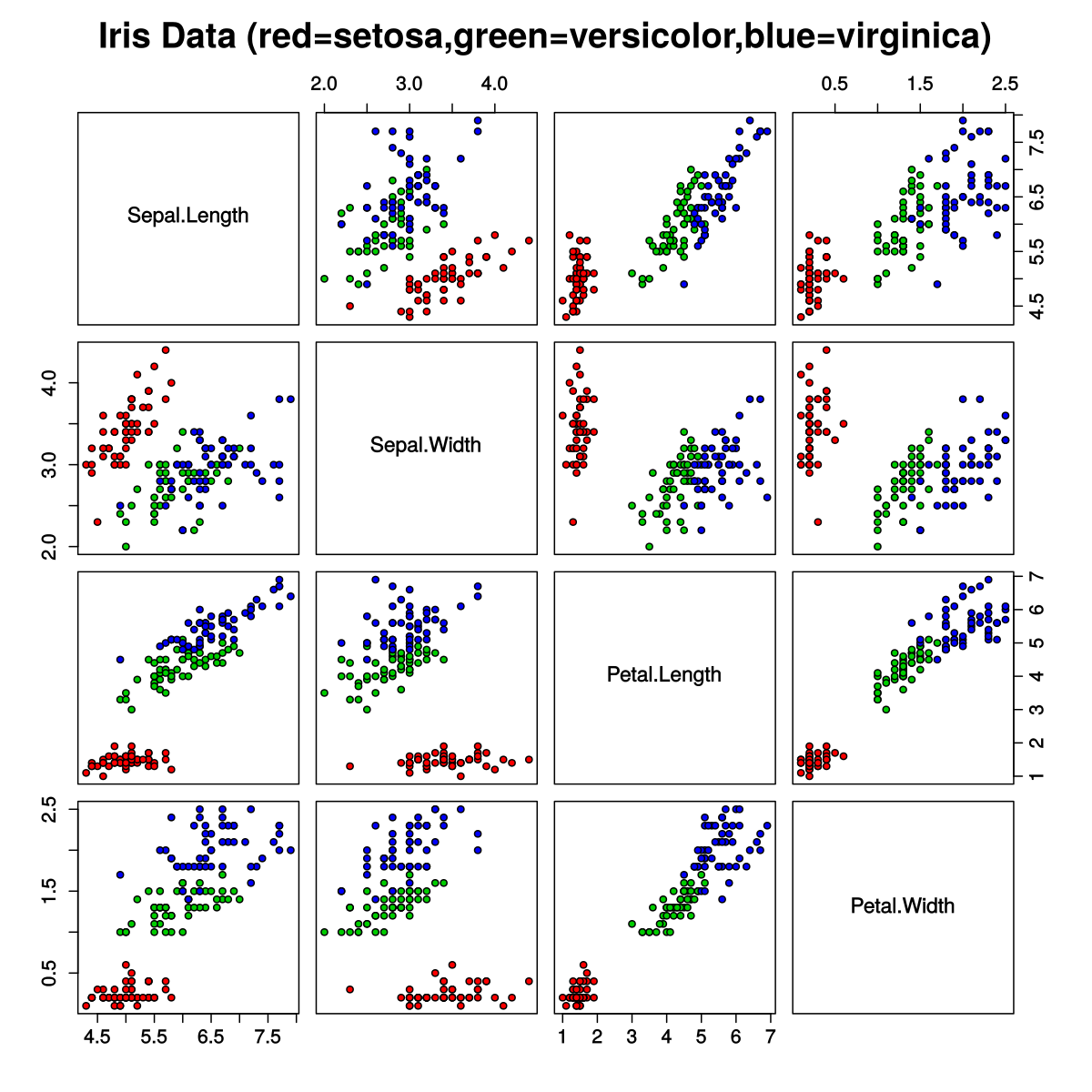
Consider an classification of Iris flower dataset which contains feature like sepal length, sepal width, petal length, petal width depending upon these the flower is going to be classified into species like Iris\_setosa, Iris\_versicolor, Iris\_virginica.

As shown in the below table :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| sepal\_length | sepal\_width | petal\_length | petal\_width | Species |
| 5.1 | 3.5 | 1.4 | 0.2 | Iris\_setosa |
| 4.9 | 3 | 1.4 | 0.2 | Iris\_setosa |
| 5.4 | 3.9 | 1.7 | 0.4 | Iris\_setosa |
| 6.4 | 3.2 | 4.5 | 1.5 | Iris\_versicolor |
| 5.6 | 2.9 | 3.6 | 1.3 | Iris\_versicolor |
| 6.1 | 3 | 4.7 | 1 | Iris\_versicolor |
| 5.8 | 2.7 | 5.1 | 1.9 | Iris\_virginica |
| 7.2 | 3.6 | 6.1 | 2.5 | Iris\_virginica |
| 7.6 | 3 | 5.8 | 2.2 | Iris\_virginica |

**Solution:**

Depending on the features like sepal\_length , sepal\_width , petal\_length , petal\_width the flower will be classified under three different species as mentioned above.Based on similarity the new data point will be classified.

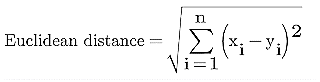
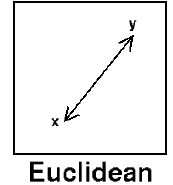
nnn

**Distance metrics :**

**1. Euclidian Distance**

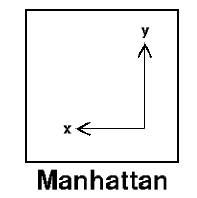
* We use this method as it treats each feature as equally important.
* It is the most popular and familiar choice which we use to calculate the distance between the two data points in a plane.(p=2)

x = (x1,x2,…………xn) and y = (y1,y2,…………yn), is

**2.Manhattan Distance**

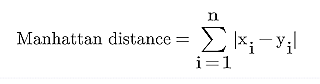
* We use Manhattan Distance if we need to calculate the distance between two data points in a grid like path.
* Let’s say, we want to calculate the distance, between two data points x and y.



* Manhattan distance will be calculated using an absolute sum of difference between its cartesian co-ordinates as below :
* where, n - number of variables, **xi** and **yi** are the variables of vectors x and y respectively, in the two dimensional vector space.(p=1)

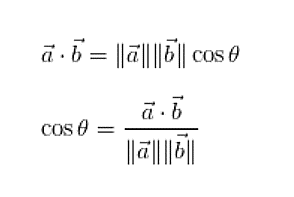
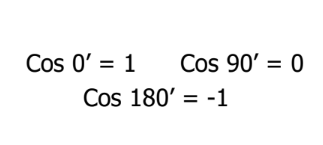
i.e.**x = (x1,x2,x3,...)  and  y = (y1,y2,y3,…),** itwillbecalculatedasbelow

**(x1 - y1)** + **(x2 - y2)**+ **(x3 - y3)** + … +**(xn - yn)**.



**3.Cosine Distance**

* Mostly Cosine distance metric is used to find similarities between different data points. In cosine metric we measure the degree of angle between two documents/vectors.This particular metric is used when the magnitude between vectors does not matter but the orientation.
* Cosine similarity formula can be derived from the equation of dot products :

* Here cosine value 1 is for vectors pointing in the same direction i.e. there are similarities between the data points. At zero for orthogonal vectors i.e. Unrelated (some similarity found). Value -1 for vectors pointing in opposite directions(No similarity).
* It ranges from [-1 , 1].

**Working of KNN :**

* Step 1 − For implementing any algorithm, we need dataset. So during the first step of KNN, we must load the training as well as test data.
* Step 2 − Next, we need to choose the value of K i.e. the nearest data points. K can be any integer.
* Step 3 − For each point in the test data do the following −
* 3.1 − Calculate the distance between test data and each row of training data with the help of any of the method namely: Euclidean, Manhattan or cosine distance. The most commonly used method to calculate distance is Euclidean.
* 3.2 − Now, based on the distance value, sort them in ascending order.
* 3.3 − Next, it will choose the top K rows from the sorted array.
* 3.4 − Now, it will assign a class to the test point based on most frequent class of these rows.
* Step 4 − End

**How to choose K value :**

1.Determine the value of K.

The first step is to determine the value of K. The determination of the K value varies greatly depending on the case. If using the Scikit-Learn Library the default value of K is 5.

2.Calculate the distance of new data with training data.

To calculate distances, 3 distance metrics that are often used are Euclidean Distance, Manhattan Distance, and Cosine Distance.When we use Sk-Learn, the default distance used is Euclidean. It can be seen in the Minkowski distance formula that there is a Hyperparameter p , if set p = 1 then it will use the Manhattan distance and p = 2 to be Euclidean.

3. Find the closest K-neighbors from the new data.

After calculating the distance, then look for K-Neighbors that are closest to the new data. If using K = 3, look for 3 training data that is closest to the new data.

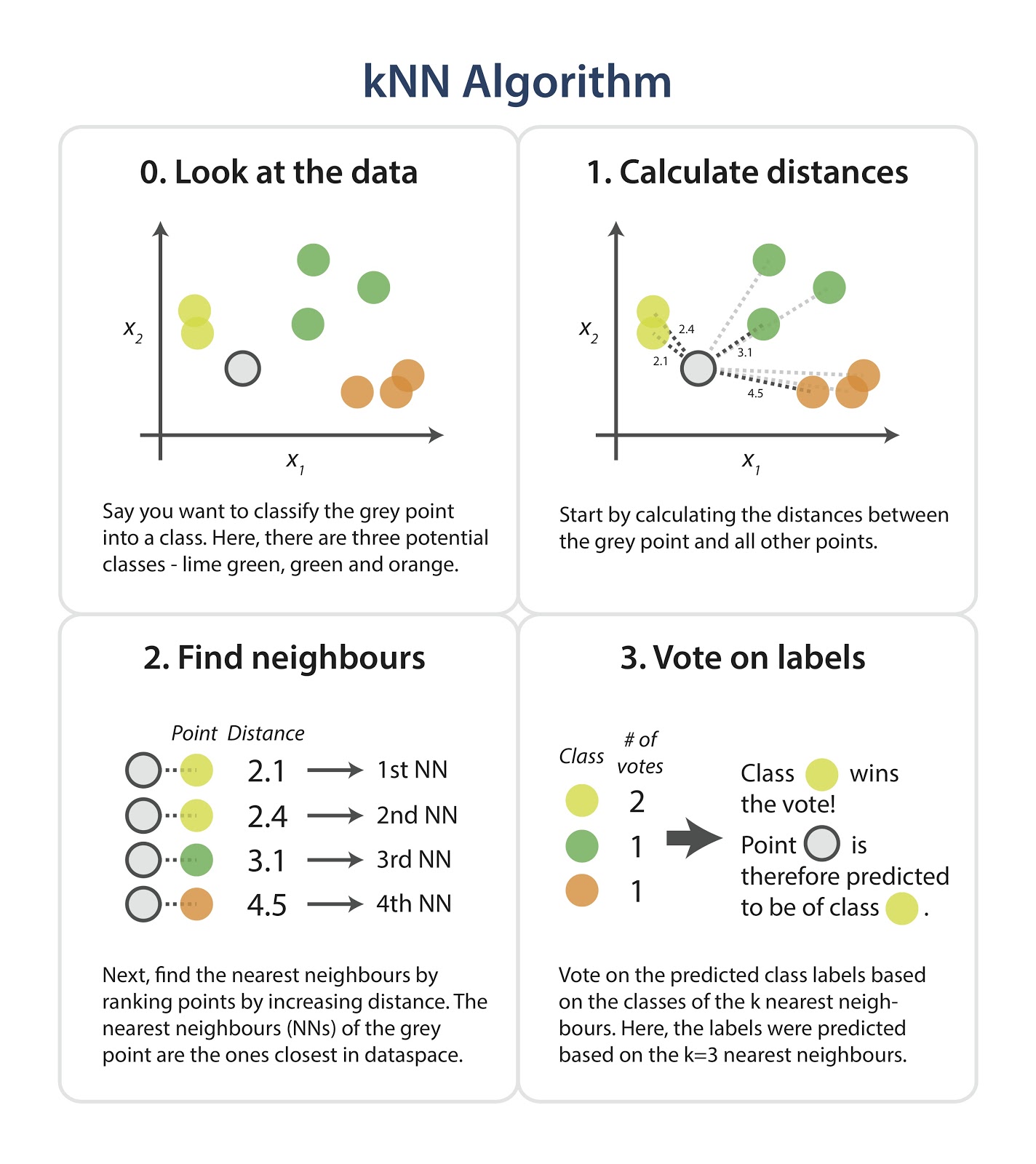
4. New Data Class Prediction.

To determine the class of new data, select the class of training data that closest to the new data and have the highest quantity.

5. Evaluation.

Calculate the accuracy of the model, if the accuracy is still low, then this process can be repeated again from step 1.

**Pictorial Represntation of KNN algorithm**

****

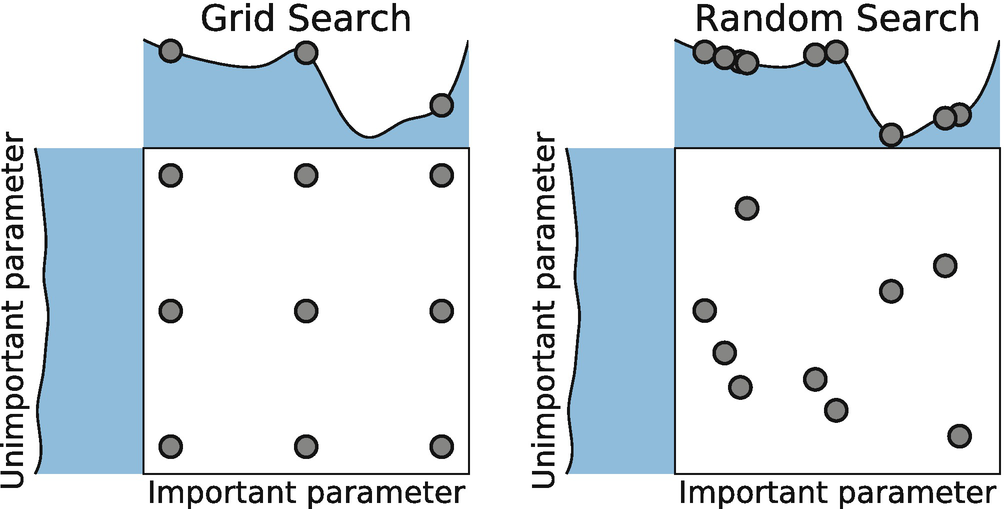
**Hyperparameter Tuning :**

* Hyperparameter tuning is choosing a set of optimal hyperparameters for a learning algorithm.
* In sklearn,hyperparameters are passed in as arguments to the constructor of the model classes.While this is an important step in modeling, it is by no means the only way to improve performance.
* Tuning Strategies

There are two different methods for optimizing hyperparameters:

1.Grid Search

2.Random Search



**1.Grid Search**

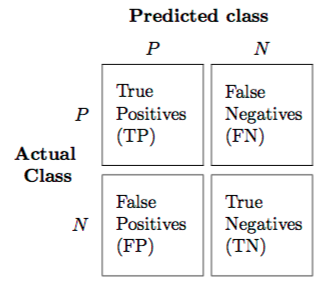
* Grid search is a traditional way to perform hyperparameter optimization. It works by searching exhaustively through a specified subset of hyperparameters.
* The benefit of grid search is that it is guaranteed to find the optimal combination of parameters supplied. The drawback is that it can be very time consuming and computationally expensive.
* We can combat this with random search.

**2.Random Search**

* Random search differs from grid search mainly in that it searches the specified subset of hyperparameters randomly instead of exhaustively. The major benefit being decreased processing time.
* There is a tradeoff to decreased processing time, however. We aren’t guaranteed to find the optimal combination of hyperparameters.

**Performance Metrics :**

The most widely used technique for summarizing the performance of a classification algorithm is the Confusion Matrix .Figure shows the confusion matrix for the case of binary classification with the following elements:



1.TruePositives(TP) is defined by the total number of accurate outputs when the actual class of the data object was True and the prediction was also the True value.

2.TrueNegatives(TN) is defined by the total number of accurate outputs when the actual class of the data object was False and the predicted is also the False value.

3.FalsePositives(FP) when the actual class of the data object was False and the output value was the True value.

4.FalseNegatives(FN) when the actual class of the data object was True and the output value was the False value.

**Metrics computed from a confusion matrix :**

A confusion matrix gives a useful information about how well the model does. However, its elements can be used to calculate many performance metrics.

1.Accuracy - is the most intuitive performance measure, and defined as the ratio of the number of correctly classified objects to the total number of objects evaluated.

2.Precision - it is simply a ratio of correctly predicted positive data objects to the total predicted positive data objects.

3.Recall - it is defined by the number of correct positive results divided by the total number of relevant samples (all samples that should have been identified as positive).

4.F-score - it can be defined as a weighted average of the precision and recall. An F-score is considered perfect when reaches its best value at 1, while the model is a total failure when it reaches the 0 value.

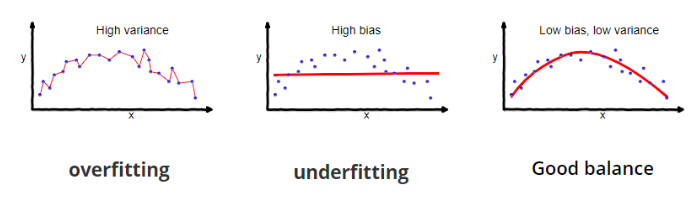
**Overfitting and Underfitting :**

**Overfitting:**

* Good performance on the training data, poor generliazation to other data.
* A statistical model is said to be overfitted, when we train it with a lot of data.
* When a model gets trained with so much of data, it starts learning from the noise and inaccurate data entries in our data set.
* Then the model does not categorize the data correctly, because of too much of details and noise.
* The causes of overfitting are the non-parametric and non-linear methods because these types of machine learning algorithms have more freedom in building the model based on the dataset and therefore they can really build unrealistic models.
* A solution to avoid overfitting is using a linear algorithm if we have linear data or using the parameters like cross-validation.
* If K=1,it will be overfitted model.

**Underfitting :**

* Poor performance on the training data and poor generalization to other data.
* A statistical model or a machine learning algorithm is said to have underfitting when it cannot capture the underlying trend of the data.
* Underfitting destroys the accuracy of our machine learning model. Its occurrence simply means that our model or the algorithm does not fit the data well enough.
* It usually happens when we have less data to build an accurate model and also when we try to build a linear model with a non-linear data.
* In such cases the rules of the machine learning model are too easy and flexible to be applied on such a minimal data and therefore the model will probably make a lot of wrong predictions.
* Underfitting can be avoided by using more data and also reducing the features by feature selection.
* If K=n then it will be underfitted model.

****

**Time and Space Complexity :**

* Time Complexity 🡪 O(nxd)

- As n training samples of d dimension.

* Space Complexity 🡪 O(nxd)

- We must be able to keep the entire training dataset in memory and performing classifications can be computationally expensive as the algorithm pass through all data points for each classification.

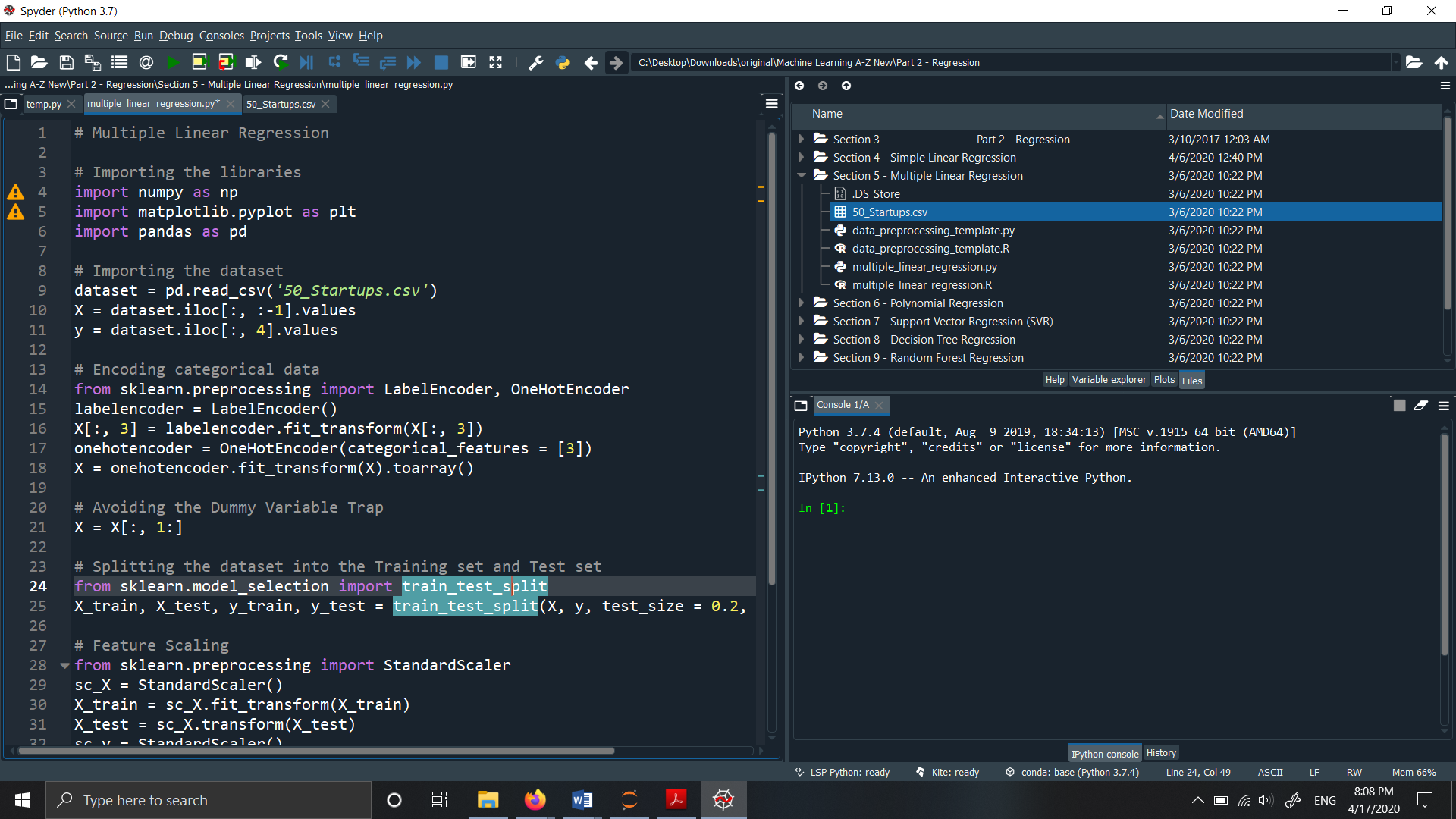
KNN with Python

Problem Statement : Given Sepal and Petal lengths and width predict the class of Iris flower.

The steps for creating the model are as below:

**Step1: Import all the necessary packages and methods from the library:**

* Import the necessary libraries like numpy, pandas, matplotlib.



**Step 2: Import the data and assign the independent variables to X and dependent variables to y.**

* Import the data using pd.read\_csv (“File Name”) and load it to the dataset variable.
* Use the command dataset.head() to look at the top 5 observations in the dataset.

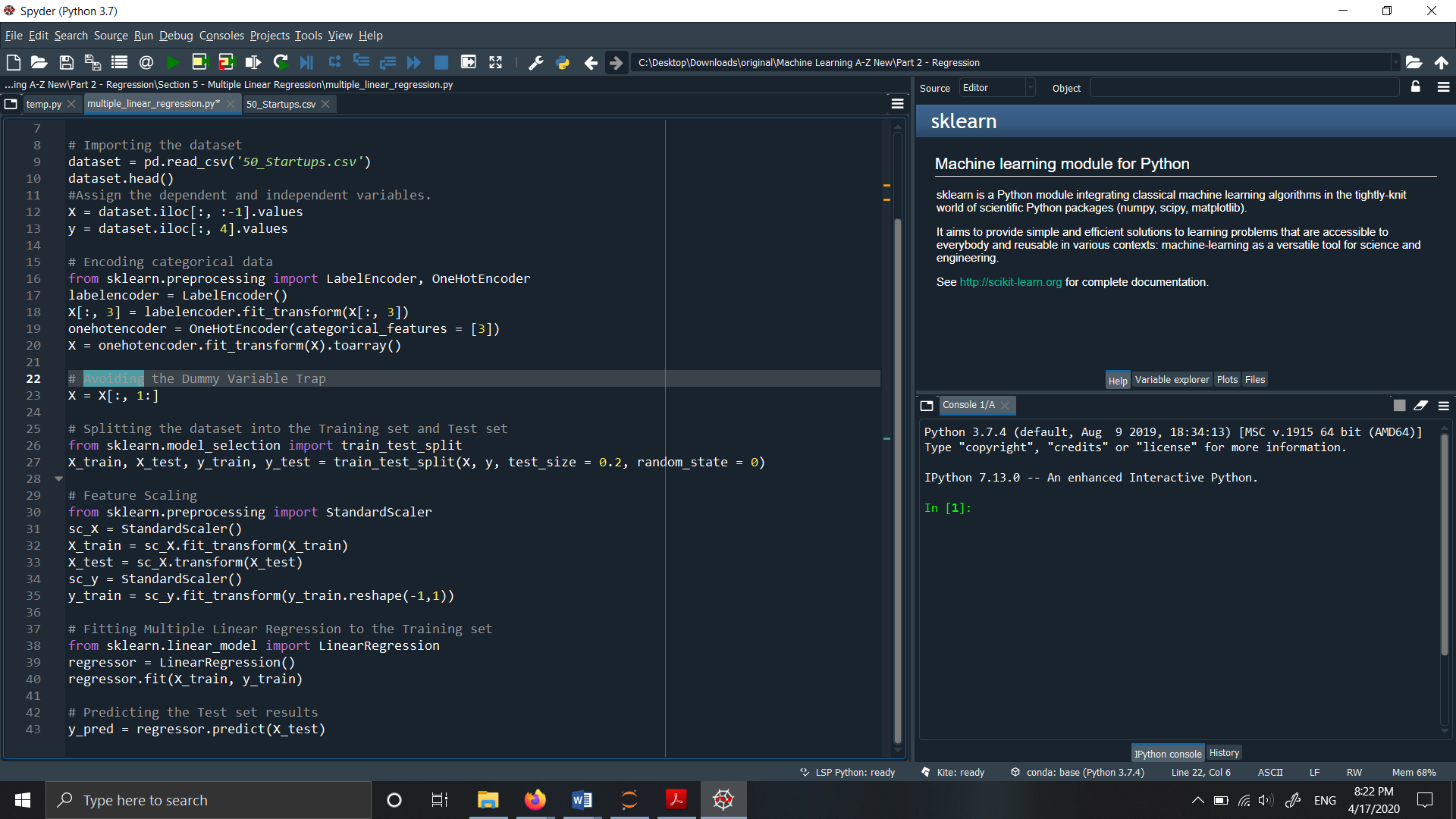
**train = pd.read\_csv("IRIS.csv")**

**train.head()**

**Step 3: Split the data as training set and test data.**

In this step the data is split into training and test data.

* Training data is used to train the model and test data is used to evaluate the model.
* test\_size specifies the percentage of data in test set.
* Different Random states shuffle the order of the observations in different ways. Specifying value to random\_state helps to reproduce the same shuffled observation everytime.



**Step 4: Fit the model to the training Set:**

* The model is fit to the training set. K nearest neighbor object is called from the sklearn.neighbors library.
* Fit the training data- X\_train and the test data y\_train to the model.

**from sklearn.neighbors import KNeighborsClassifier**

**knn = KNeighborsClassifier(n\_neighbors=5)**

**knn.fit(X\_train, y\_train)**

**y\_pred = knn.predict(X\_test)**

**Step 5 : Print the accuracy score of predicted value.**

* In this step the model prints the accuracy of the model.

**from sklearn.metrics import accuracy\_score**

**print("Accuracy:",accuracy\_score(y\_test, y\_pred))**

**Output:**

Accuracy: 0.9777777777777777

**Step 6: Printing both confusion matrix and classification report for the dataset**

**from sklearn.metrics import confusion\_matrix**

**from sklearn.metrics import classification\_report**

**cm = confusion\_matrix(y\_test, y\_pred)**

**print ("Confusion Matrix : \n", cm)**

**print(classification\_report(y\_test, y\_pred))**

**Output :**

Confusion Matrix :

[[16 0 0]

[ 0 19 1]

[ 0 0 9]]

precision recall f1-score support

Iris-setosa 1.00 1.00 1.00 16

Iris-versicolor 1.00 0.95 0.97 20

Iris-virginica 0.90 1.00 0.95 9

accuracy 0.98 45

macro avg 0.97 0.98 0.97 45

weighted avg 0.98 0.98 0.98 45

**Advantages and Disadvantages KNN:**

**Advantages :**

* It is very simple algorithm to understand and interpret.
* It is very useful for nonlinear data because there is no assumption about data in this algorithm.
* It is a versatile algorithm as we can use it for classification as well as regression.
* It has relatively high accuracy but there are much better supervised learning models than KNN.

**Disadvantages:**

* It is computationally a bit expensive algorithm because it stores all the training data.
* High memory storage required as compared to other supervised learning algorithms.
* Prediction is slow in case of big N.
* It is very sensitive to the scale of data as well as irrelevant features.

**Applications of KNN :**

The following are some of the areas in which KNN can be applied successfully −

* Banking System

KNN can be used in banking system to predict weather an individual is fit for loan approval or not.

* Calculating Credit Ratings

KNN algorithms can be used to find an individual’s credit rating by comparing with the persons having similar traits.

* Politics

With the help of KNN algorithms, we can classify a potential voter into various classes like “Will Vote”, “Will not Vote”, “Will Vote to Party ‘Rose’, “Will Vote to Party ‘Lily’.Other areas in which KNN algorithm can be used are Speech Recognition, Handwriting Detection, Image Recognition and Video Recognition.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***