**CHAPTER 1**

**INTRODUCTION**

India being an agricultural country, its economy predominantly depends on agriculture yield growth and allied agro industry products. In India, agriculture is largely influenced by rainwater which is highly unpredictable. Agriculture growth also depends on diverse soil parameters, namely Nitrogen, Phosphorus, Potassium, Crop rotation, Soil moisture, Surface temperature and also on weather aspects which include temperature, rainfall, etc.

India now is rapidly progressing towards technical development. Thus, technology will prove to be beneficial to agriculture which will increase crop productivity resulting in better yields to the farmer. The proposed project provides a solution for Smart Agriculture by monitoring the agricultural field which can assist the farmers in increasing productivity to a great extent**.**

It uses data analytics techniques in order to predict the most profitable crop in the current weather and soil conditions. The proposed system will integrate the data obtained from repository, weather department and by applying machine learning algorithm: Multiple Linear Regression, a prediction of most suitable crops according to current environmental conditions is made.

This provides a farmer with variety of options of crops that can be cultivated. Thus, the project develops a system by integrating data from various sources, data analytics, prediction analysis which can improve crop yield productivity and increase the profit margins of farmer helping them over a longer run.

**1.1 System requirements**

* Python 3.6 or later
* Spyder IDE anaconda

**1.2 Special Terms used:**

**Input:** The prediction of crop is dependent on numerous factors such as Soil Nutrients, weather and past crop production in order to predict the crop accurately. All these factors are location reliant and thus the location of user is taken as an input to the system.

**Data Acquisition:** Data acquisition is the process of sampling signals that measure real world physical conditions and converting the resulting samples into digital numeric values that can be manipulated by a computer. Depending on the current user location, the system mines the soil properties in the respective area from the soil repository. In a similar approach, weather parameters are extracted from the weather data set.

The components of data acquisition systems include:

* Sensors, to convert physical parameters to electrical signals.
* Signal conditioning circuitry, to convert sensor signals into a form that can be converted to digital values.
* Analog-to-digital converters, to convert conditioned sensor signals to digital values.

Data acquisition begins with the [physical phenomenon](https://en.wikipedia.org/wiki/Physical_phenomenon) or [physical property](https://en.wikipedia.org/wiki/Physical_property) to be measured. Examples of this include temperature, light intensity, gas pressure, fluid flow, and force. Regardless of the type of physical property to be measured, the physical state that is to be measured must first be transformed into a unified form that can be sampled by a data acquisition system. The task of performing such transformations falls on devices called *sensors*.

A data acquisition system is a collection of software and hardware that allows one to measure or control physical characteristics of something in the real world.

**Data Processing:**

Data preprocessing is an important step in the [data mining](https://en.wikipedia.org/wiki/Data_mining) process. The phrase ["garbage in, garbage out"](https://en.wikipedia.org/wiki/GIGO) is particularly applicable to [data mining](https://en.wikipedia.org/wiki/Data_mining) and [machine learning](https://en.wikipedia.org/wiki/Machine_learning) projects. Data-gathering methods are often loosely controlled, resulting in [out-of-range](https://en.wikipedia.org/w/index.php?title=Range_error&action=edit&redlink=1) values (e.g., Income: −100), impossible data combinations (e.g., Sex: Male, Pregnant: Yes), [missing values](https://en.wikipedia.org/wiki/Missing_values), etc. Analyzing data that has not been carefully screened for such problems can produce misleading results. Thus, the representation and [quality of data](https://en.wikipedia.org/wiki/Data_quality) is first and foremost before running an analysis. Often, data preprocessing is the most important phase of a [machine learning](https://en.wikipedia.org/wiki/Machine_learning) project, especially in [computational biology](https://en.wikipedia.org/wiki/Computational_biology)

If there is much irrelevant and redundant information present or noisy and unreliable data, then [knowledge discovery](https://en.wikipedia.org/wiki/Knowledge_discovery) during the training phase is more difficult. Data preparation and filtering steps can take considerable amount of processing time. Data preprocessing includes [cleaning](https://en.wikipedia.org/wiki/Data_cleaning), [Instance selection](https://en.wikipedia.org/wiki/Instance_selection), [normalization](https://en.wikipedia.org/wiki/Data_normalization), [transformation](https://en.wikipedia.org/wiki/Data_transformation), [feature extraction](https://en.wikipedia.org/wiki/Feature_extraction) and [selection](https://en.wikipedia.org/wiki/Feature_selection), etc. The product of data preprocessing is the final [training set](https://en.wikipedia.org/wiki/Training_set).

Data pre-processing may affect the way in which outcomes of the final data processing can be interpreted.  This aspect should be carefully considered when interpretation of the results is a key point, such in the multivariate processing of chemical data.

Tasks of data pre-processing

* [Data cleansing](https://en.wikipedia.org/wiki/Data_cleansing) : **Data cleansing** or **data cleaning** is the process of detecting and correcting (or removing) corrupt or inaccurate [records](https://en.wikipedia.org/wiki/Storage_record) from a record set, [table](https://en.wikipedia.org/wiki/Table_(database)), or [database](https://en.wikipedia.org/wiki/Database) and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the [dirty](https://en.wikipedia.org/wiki/Dirty_data) or coarse data. Data cleansing may be performed [interactively](https://en.wikipedia.org/wiki/Interactively) with [data wrangling](https://en.wikipedia.org/wiki/Data_wrangling) tools, or as [batch processing](https://en.wikipedia.org/wiki/Batch_processing) through [scripting](https://en.wikipedia.org/wiki/Script_(computing)).
* [Data editing](https://en.wikipedia.org/wiki/Data_editing) : **Data editing** is defined as the process involving the review and adjustment of collected [survey data](https://en.wikipedia.org/wiki/Survey_data). The purpose is to control the quality of the collected data. Data editing can be performed manually, with the assistance of a computer or a combination of both.
* [Data reduction](https://en.wikipedia.org/wiki/Data_reduction): **Data reduction** is the transformation of numerical or alphabetical [digital information](https://en.wikipedia.org/wiki/Digital_information) derived empirically or [experimentally](https://en.wikipedia.org/wiki/Experimental_data) into a corrected, ordered, and simplified form. The basic concept is the reduction of multitudinous amounts of data down to the meaningful parts.

When information is derived from instrument readings there may also be a transformation from [analog to digital](https://en.wikipedia.org/wiki/Digitization) form. When the data are already in digital form the 'reduction' of the data typically involves some editing, [scaling](https://en.wikipedia.org/wiki/Scaling_(geometry)), [encoding](https://en.wikipedia.org/wiki/Encoding), [sorting](https://en.wikipedia.org/wiki/Sorting), collating, and producing tabular summaries. When the observations are discrete but the underlying phenomenon is continuous then [smoothing](https://en.wikipedia.org/wiki/Smoothing) and [interpolation](https://en.wikipedia.org/wiki/Interpolation) are often needed. Often the data reduction is undertaken in the presence of reading or [measurement errors](https://en.wikipedia.org/wiki/Measurement_error). Some idea of the nature of these errors is needed before the most likely value may be determined.

* [Data wrangling](https://en.wikipedia.org/wiki/Data_wrangling) : **Data wrangling**, sometimes referred to as **data munging**, is the process of transforming and [mapping data](https://en.wikipedia.org/wiki/Data_mapping) from one "[raw](https://en.wikipedia.org/wiki/Raw_data)" data form into another [format](https://en.wikipedia.org/wiki/Content_format) with the intent of making it more appropriate and valuable for a variety of downstream purposes such as analytics. A **data wrangler** is a person who performs these transformation operations.

A crop can be cultivable only if apropos conditions are met. These include extensive parameters allied to soil and weather. These constraints are compared and the apt crops are ascertained.

Multiple Linear Regression is used by the system to predict the crop. The prediction is based on past production data of crops i.e.: identifying the tangible weather and soil parameters and comparing it with current conditions which will predict the crop more accurately and in a practical manner.

Output: The most profitable crop is predicted by the system using Multiple Linear Regression algorithm and the user is provided with multiple suggestions of crop conferring to the duration of crop.

**CHAPTER 2**

**SRS**

**2.1 Purpose**

A Software Requirements Specifications (SRS) is a document that describes the nature of the project. In simple words, SRS document is a manual of a project provided it is prepared before you start a project. The purpose of this document is to build an online mail system using Python.

Moreover the use of python language and its libraries make it more simple as they have pre defined functions which are only to be called according to the need of the project.

**PURPOSE :-** The purpose of the project is to predict crop production based on weather dataset and different places.

* **Analyzing the dataset :-** For crop prediction we will analyze the dataset of weather . as weather plays important role in crop production.
* **Developing prediction module :-** After analyzing dataset , we will develop a prediction module which help in development of human interface.
* **Human-machine interface :-** Now we will use machine learning algorithms and python to develop human-machine interface.

2.2 FEASIBILITY STUDY

The proposed system takes into consideration the data related to soil, weather and past year production and suggests which are the best profitable crops which can be cultivated in theapropos environmental condition. As the system lists out all possible crops, it helps the farmer in decision making of which crop to cultivate. Also, this system takes into consideration the past production of data which will help the farmer get insight into the demand and the cost of various crops in market. As maximum types of crops will be covered under this system, farmer may get to know about the crop which may never have been cultivated.

In the future, all farming devices can be connected over the internet using IOT. The sensors can be employed in farm which will collect the information about the current farm conditions and devices can increase the moisture, acidity, etc. accordingly.

The vehicles used in farm like tractor will be connected to internet in future which will, in real time pass data to farmer about crop harvesting and the disease crops may be suffering from thus helping the farmer in taking appropriate action. Further the best profitable crop can also be found in light of the monetary and inflation ratio.

THE MODULES ARE AS FOLLOWS :-

**2.2.1 ) DATA PREPROCESSING**

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm.  
Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.

**Need of Data Preprocessing**

* For achieving better results from the applied model in Machine Learning projects the format of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format.

for example, Random Forest algorithm does not support null values, therefore to execute random forest algorithm null values have to be managed from the original raw data set.

* Another aspect is that data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithms are executed in one data set, and best out of them is chosen.

**3 different data preprocessing techniques for machine learning.**

1. **Rescale Data**  
   • When our data is comprised of attributes with varying scales, many machine learning algorithms can benefit from rescaling the attributes to all have the same scale.  
   • This is useful for optimization algorithms in used in the core of machine learning algorithms like gradient descent.  
   • It is also useful for algorithms that weight inputs like regression and neural networks and algorithms that use distance measures like K-Nearest Neighbors.  
   • We can rescale your data using scikit-learn using the [MinMaxScaler](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html) class.
2. **Binarize Data**   
   • We can transform our data using a binary threshold. All values above the threshold are marked 1 and all equal to or below are marked as 0.  
   • This is called binarizing your data or threshold your data. It can be useful when you have probabilities that you want to make crisp values. It is also useful when feature engineering and you want to add new features that indicate something meaningful.  
   • We can create new binary attributes in Python using scikit-learn with the [Binarizer](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.Binarizer.html) class.
3. **StandardizeData**  
   • Standardization is a useful technique to transform attributes with a Gaussian distribution and differing means and standard deviations to a standard Gaussian distribution with a mean of 0 and a standard deviation of 1.  
   • We can standardize data using scikit-learn with the [StandardScaler](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html) class.

**2.2.2 ) MODEL-BUILDING CODE:-**

* **Pandas :-**Pandas is the most popular python library that is used for data analysis. It provides highly optimized performance with back-end source code is purely written in Python.

importpandas as pd  # Import Panda Library

a =pd.Series(Data, index =Index)

* **Numpy :- Numpy**is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamentalpackage for scientific computing with Python. An array class in Numpy is called as **ndarray.** Elements in Numpy arrays are accessed by using square brackets and can be initialized by using nested Python Lists.

*importnumpy as np*

*arr =np.array([1, 2, 3])*

*print("Array with Rank 1: \n",arr)*

*arr =np.array([[1, 2, 3],[4, 5, 6]])*

*print("Array with Rank 2: \n", arr)*

*arr =np.array((1, 3, 2))*

*print("\nArray created using "*

*"passed tuple:\n", arr)*

# Matplotlib :- Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack.One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

python -mpip install -U matplotlib

from matplotlib import pyplot as plt

*or*

import matplotlib.pyplot as plt

# Multiple Linear Regression using Python :- Multiple Linear Regression attempts to model the Relationship between two or more features and a response by fitting a linear equation to observed data. The steps to perform multiple linear Regression are almost similar to that of simple linear Regression. The Difference Lies in the Evalution.

# Assumption of Regression Model :

* Linearity: The relationship between dependent and independent variables should be linear.
* Homoscedasticity: Constant variance of the errors should be maintained.
* Multivariate normality: Multiple Regression assumes that the residuals are normally distributed.
* Lack of Multicollinearity: It is assumed that there is little or no multicollinearity in the data.

Steps Involved in any Multiple Linear Regression Model

**Step #1:** Data PreProcessing  
 a) Importing the Libraries.  
 b) Importing the Data Set.  
 c) Encoding the Categorical Data.  
 d) Avoiding the Dummy Variable Trap.  
 e) Splitting the Data set into Training Set and Test Set.

**Step #2:** Fitting Multiple Linear Regression to the Training set

**Step #3:** Predicting the Test set results.

*import numpy as np*

*import matplotlib as mpl*

*from mpl\_toolkits.mplot3d import Axes3D*

*import matplotlib.pyplot as plt*

*def generate\_dataset(n):*

*x = []*

*y = []*

*random\_x1 = np.random.rand()*

*random\_x2 = np.random.rand()*

*for i in range(n):*

*x1 = i*

*x2 = i/2 + np.random.rand()\*n*

*x.append([1, x1, x2])*

*y.append(random\_x1 \* x1 + random\_x2 \* x2 + 1)*

*return np.array(x), np.array(y)*

*x, y = generate\_dataset(200)*

*mpl.rcParams['legend.fontsize'] = 12*

*fig = plt.figure()*

*ax = fig.gca(projection ='3d')*

*ax.scatter(x[:, 1], x[:, 2], y, label ='y', s = 5)*

*ax.legend()*

*ax.view\_init(45, 0)*

*plt.show(*

**CHAPTER 3**

**ARCHITECTURE DESIGN**

3.1 design view

This is the main idea of our project . basically, collecting all the data about the crops , then training the dataset and applying multilinear regression to get the output of the model.

**Linear Regression** is a machine learning algorithm based on **supervised learning**. It performs a **regression task**. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables, they are considering and the number of independent variables being used.

Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output). Hence, the name is Linear Regression.  
In the figure above, X (input) is the work experience and Y (output) is the salary of a person. The regression line is the best fit line for our model.

**Hypothesis function for Linear Regression :**



Fig i .

While training the model we are given :  
**x:** input training data (univariate – one input variable(parameter))  
**y:** labels to data (supervised learning)

When training the model – it fits the best line to predict the value of y for a given value of x. The model gets the best regression fit line by finding the best θ1 and θ2 values.  
**θ1:** intercept  
**θ2:** coefficient of x

Once we find the best θ1 and θ2 values, we get the best fit line. So when we are finally using our model for prediction, it will predict the value of y for the input value of x.

**Cost Function (J):**  
By achieving the best-fit regression line, the model aims to predict y value such that the error difference between predicted value and true value is minimum. So, it is very important to update the θ1 and θ2 values, to reach the best value that minimize the error between predicted y value (pred) and true y value (y).

Cost function(J) of Linear Regression is the **Root Mean Squared Error (RMSE)** between predicted y value (pred) and true y value (y).

[**Gradient Descent**](https://www.geeksforgeeks.org/gradient-descent-in-linear-regression/)**:**  
To update θ1 and θ2 values in order to reduce Cost function (minimizing RMSE value) and achieving the best fit line the model uses Gradient Descent. The idea is to start with random θ1 and θ2 values and then iteratively updating the values, reaching minimum cost.

**3.2 Algorithm used**

**Multiple linear regression** (MLR/multiple regression) is a statistical technique. It can use several variables to predict the outcome of a different variable. The **goal of multiple regression is to model the linear relationship between your independent variables and your dependent variable**. It looks at how multiple independent variables are related to a dependent variable.

**Multiple linear regression** is what you can use when you have a bunch of different independent variables!

Multiple regression analysis has three main uses.

* You can look at the strength of the effect of the independent variables on the dependent variable.
* You can use it to ask how much the dependent variable will change if the independent variables are changed.
* You can also use it to predict trends and future values.

A Linear Regression model that contains more than one predictor variable is called a Multiple Linear Regression model. The following model is A Multiple Linear Regression model with two predictor variables, 𝑥1and 𝑥2.

𝑌 = 𝛽0 + 𝛽1𝑥1 + 𝛽2𝑥2+ ∈

Where,

𝛽0,𝛽1,𝛽2 … are coefficients of Multiple LinearRegression

𝑥𝑖1, 𝑥𝑖2... are independent variables.

The model is linear because it is linear in the parameters 𝛽0, 𝛽1and 𝛽2. The model describes a plane in the three- dimensional space of 𝑌, 𝑥1𝑎𝑛𝑑𝑥2. The parameter 𝛽0is the intercept of this plane. Parameters 𝛽1and𝛽2are referred to as partial regression coefficients. Parameter 𝛽1represents the change in the mean response corresponding to a unit change in 𝑥1when 𝑥2is held constant. Parameter 𝛽2represents the change in the mean response corresponding to a unit change in 𝑥2when 𝑥1is heldconstant.

**Input**: The prediction of crop is dependent on numerous factors such as Soil Nutrients,

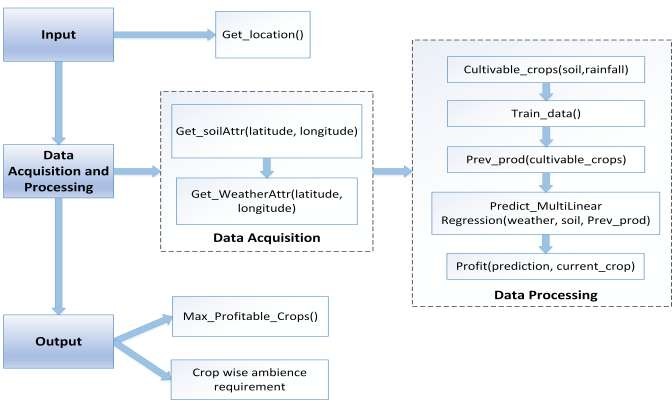


Fig ii

weather and past crop production in order to predict the crop accurately. All these factors are location reliant and thus the location of user is taken as an input to the system.

**Data Acquisition**: Depending on the current user location, the system mines the soil properties in the respective area from the soil repository. In a similar approach, weather parameters are extracted from the weather data set.

**Data Processing**: A crop can be cultivable only if apropos conditions are met. These include extensive parameters allied to soil and weather. These constraints are compared and the apt crops are ascertained. Multiple Linear Regression is used by the system to predict the crop. The prediction is based on past production data of crops i.e.: identifying the tangible weather and soil parameters and comparing it with current conditions which will predict the crop more accurately and in a practicalmanner.

**Output**: The most profitable crop is predicted by the system using Multiple Linear Regression algorithm and the user is provided with multiple suggestions of crop conferring to the duration of crop.

**CHAPTER 4**

**PROJECT METHODOLOGY**

**4.1 multiple linear regression**

The difference between simple linear regression and multiple linear regression is that, multiple linear regression has (>1) independent variables, whereas simple linear regression has only 1 independent variable. In this project, Multiple Linear Regression algorithm is used to predict the crops. Multiple Regression is an extension of simple Linear Regression.

It is used when we want to predict the value of a variable based on the value of two or more other variables. The variable we want to predict is called the dependent variable (or sometimes, the outcome, target or criterion variable). The variables we are using to predict the value of the dependent variable are called the independent variables (or sometimes, the predictor, explanatory or regressor variables).

For example, Multiple Regression to understand whether exam performance can be predicted based on revision time, test anxiety, lecture attendance and gender. Multiple Regression also allows you to determine the overall fit (variance) of the model and the relative contribution of each of the predictors to the total variance.

* Though it is always fascinating to read about future, the most important part is the technology that paves the way for it. Agricultural machine learning, for instance, is not a mysterious trick or magic, but a set of well-defined models that collect specific data and apply specific algorithms to achieve expected results.

So far, the distribution of machine learning is unequal throughout the agriculture. Mostly, machine learning techniques are used in crop management processes, following with farming conditions management and livestock management.

ANNs are inspired by the human brain functionality and represent a simplified model of the structure of the biological neural network emulating complex functions such as pattern generation, cognition, learning, and decision making. Such models are typically used for regression and classification tasks which prove their usefulness in crop management and detection of weeds, diseases, or specific characteristics. The recent development of ANNs into deep learning that has expanded the scope of ANN application in all domains, including agriculture.

SVMs are binary classifiers that construct a linear separating hyperplane to classify data instances. SVMs are used for classification, regression, and clustering. In farming, they are used to predict yield and quality of crops as well as livestock production.

More intricate tasks, such as animal welfare measurement, require different approaches, such as multiple classifier systems in ensemble learning or Bayesian models — probabilistic graphical models in which the analysis is undertaken within the context of Bayesian inference.

Though still in the beginning of its journey, ML-driven farms are already evolving into artificial intelligence systems. At present, machine learning solutions tackle individual problems, but with further integration of automated data recording, data analysis, machine learning, and decision-making into an interconnected system, farming practices would change into with the so-called knowledge-based agriculture that would be able to increase production levels and products quality.

The multiple linear regression explains the relationship between**one continuous dependent variable** (y) and **two or more independent variables**(x1, x2, x3… etc)**.**

Note that it says **CONTINUOUS** dependant variable. Since y is the sum of beta, beta1 x1, beta2 x2 etc etc, the resulting y will be a number, a continuous variable, instead of a “yes”, “no” answer (categorical).

For example, with linear regression, I would be trying to find out **how much Decibels of**noise is being produced, and not if it’s noisy or not (Noisy | Not).

**4.2 backward elimination**

#### 4.2.1 What is Backward Elimination

* The idea of Backward Elimination is to remove independent variables that are not statistically significant.
* If your dataset is huge, this could make a great difference, because your model can run with less data.
* Our goal here is to find a group of independent variables that all big impact to the dependent variable.

#### 4.2.2Mechanism of Backward Elimination

1. Select a significant level (ie: 0.05 ; If the P value is greater than this significant level, then we will remove it)
2. First fit ALL variables to the model.
3. Find the P values for ALL variables.
4. Remove the variable with the largest P value.
5. Fit the model with a variable removed from Step 4.
6. Repeat Step 4 & 5 , until all P values are smaller than the significant level defined in Step 1.
7. Model is ready.

#### 4.2.3 How to do Backward Elimination in Python

* Recall that the formula to Linear Regression is :  y= b + x(n)…x(n)
  + b is the intercept.
  + We do not have b here in this model.
  + To create a fake intercept, we concat our model to a column that fills with 1.
  + X=np.append(arr = np.ones((50,1)).astype(int), values = X, axis = 1)
  + np stands for numpy, which is a library that we have imported at the beginning.
* We are going to use statsmodels.formula.api. Hence we need to import it as sm.
* xelimination is created. (ie: xelimination = X[:,[0,1,2,3,4,5]] ) The first “:” means that it is going to include all of the rows, and then the barack after the commas indicate the column index that it includes. This way, we can remove a column ( or remove an independent variable easily)
  + For example, if we want to remove the second column ( column index =1) now, then we are going to write xelimination = X[:,[0, 2,3,4,5]].
* regressorOLS = sm.OLS(y, xelimination).fit() ; is going to fit the Multivariate Linear Regression. Beware to indicate the y variable and the x variable correctly.
* regressorOLS.summary() ; this is going to show you the p value in the regression !
  + if regressorOLS.summary() does not work, then try print(regressorOLS.summary()).
* OLS here stands for “Ordinary Least Square”.
* The example below shows you how to do Backward Elimination in Python.

#### Example of Backward Elimination in Python

**#Import libraries**

import numpy as np  
import matplotlib.pyplot as plt  
import pandas as pd

**#Import data**  
dataset = pd.read\_csv(‘multivariate\_data.csv’)  
x = dataset.iloc[:,:-1].values  
y =dataset.iloc[:,4].values

**#Encode Categorical Data using LabelEncoder and OneHotEncoder**  
from sklearn.preprocessing import LabelEncoder,OneHotEncoder  
labelencoder\_x=LabelEncoder()  
x[:,3]=labelencoder\_x.fit\_transform(x[:,3])  
onehotencoder=OneHotEncoder(categorical\_features =[3])  
x=onehotencoder.fit\_transform(x).toarray()

**#Remove Dummy Variable Trap**  
x=x[:, 1:]

**#splitting training set and testing set**  
from sklearn.cross\_validation import train\_test\_split  
xtrain, xtest, ytrain, ytest =train\_test\_split(x,y,test\_size=0.2)

**# Training the Multivariate Linear Regression Model**  
from sklearn.linear\_model import LinearRegression  
regressor = LinearRegression()  
regressor.fit(xtrain, ytrain)

**# Predicting the Test set results**  
y\_prediction= regressor.predict(xtest)

**# Backward Eliminiation**

**# Insert B Intercept**  
X=np.append(arr = np.ones((50,1)).astype(int), values = X, axis = 1)

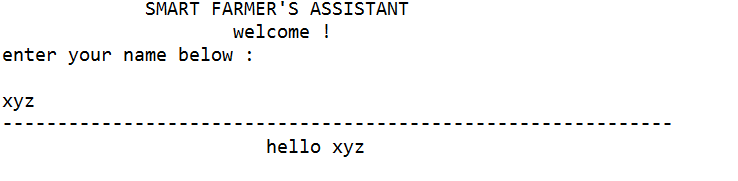
**# Call Ordinary Least Square**  
import statsmodels.formula.api as sm  
xelimination = X[:,[0,1,2,3,4,5]]  
regressorOLS = sm.OLS(y, xelimination).fit()  
regressorOLS.summary()  
xelimination = X[:,[0,1,3,4,5]]  
regressorOLS = sm.OLS(y, xelimination).fit()  
regressorOLS.summary()  
xelimination = X[:,[0,3,4,5]]  
regressorOLS = sm.OLS(y, xelimination).fit()  
regressorOLS.summary()  
xelimination = X[:,[0,3,5]]  
regressorOLS = sm.OLS(y, xelimination).fit()  
regressorOLS.summary()  
xelimination = X[:,[0,3]]  
regressorOLS = sm.OLS(y, xelimination).fit()  
regressorOLS.summary()

* If all of the independent variables have large P values, then you should try model that are not linear.
  + Try Kernal SVM

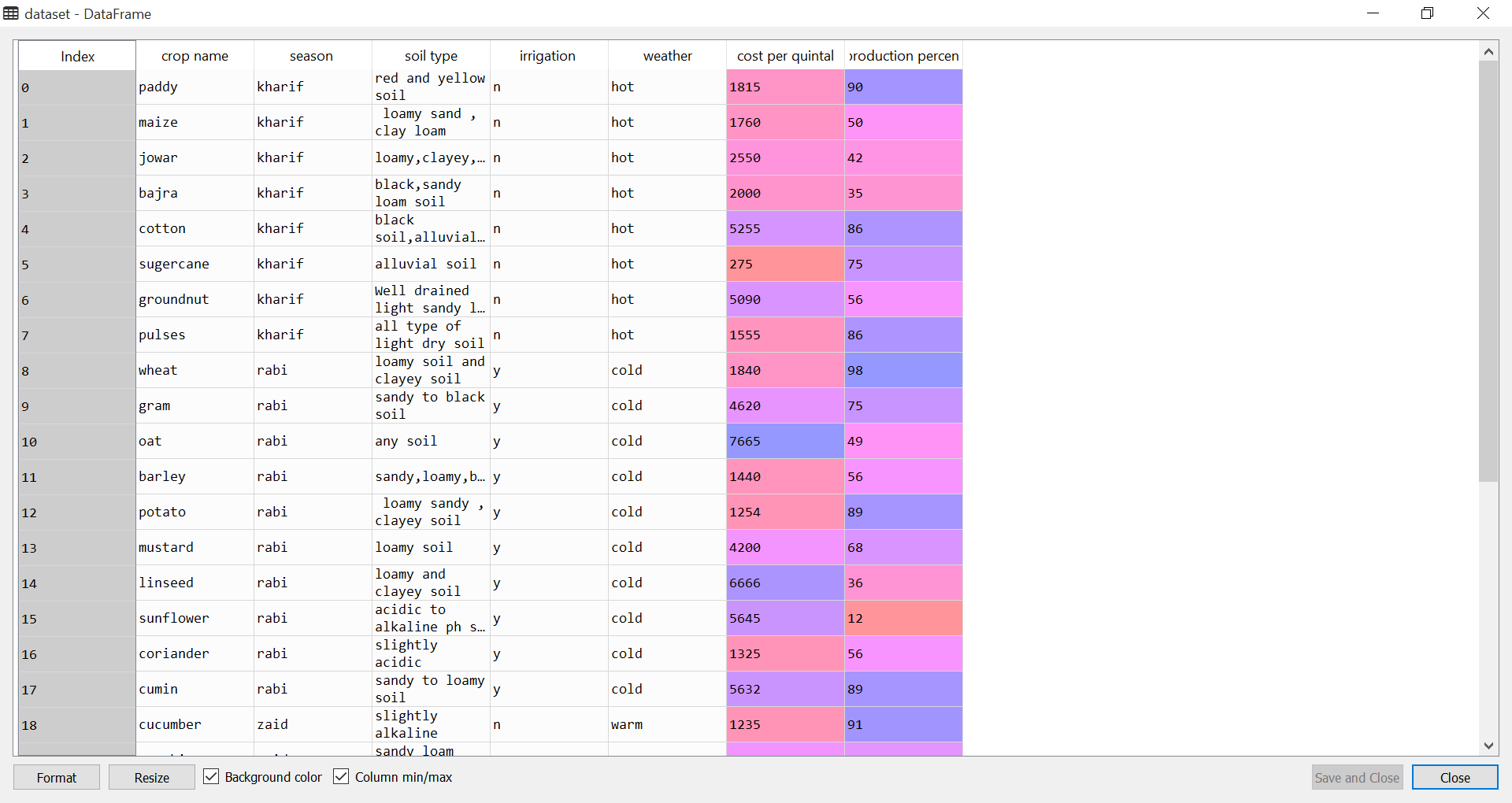
**CHAPTER 5**

**RESULTS**

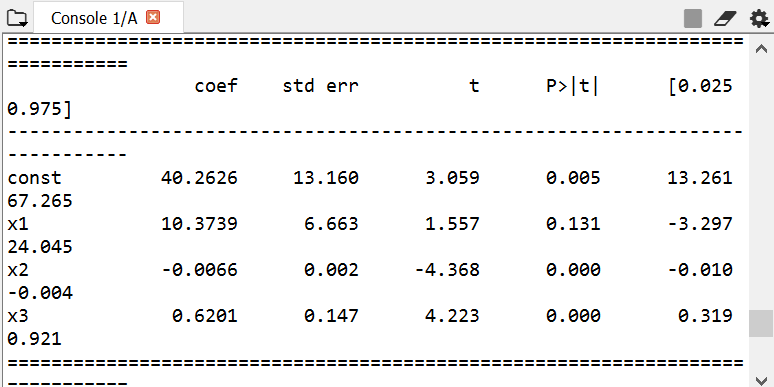
5.1 WELCOME PAGE

****

5.2 Dataset



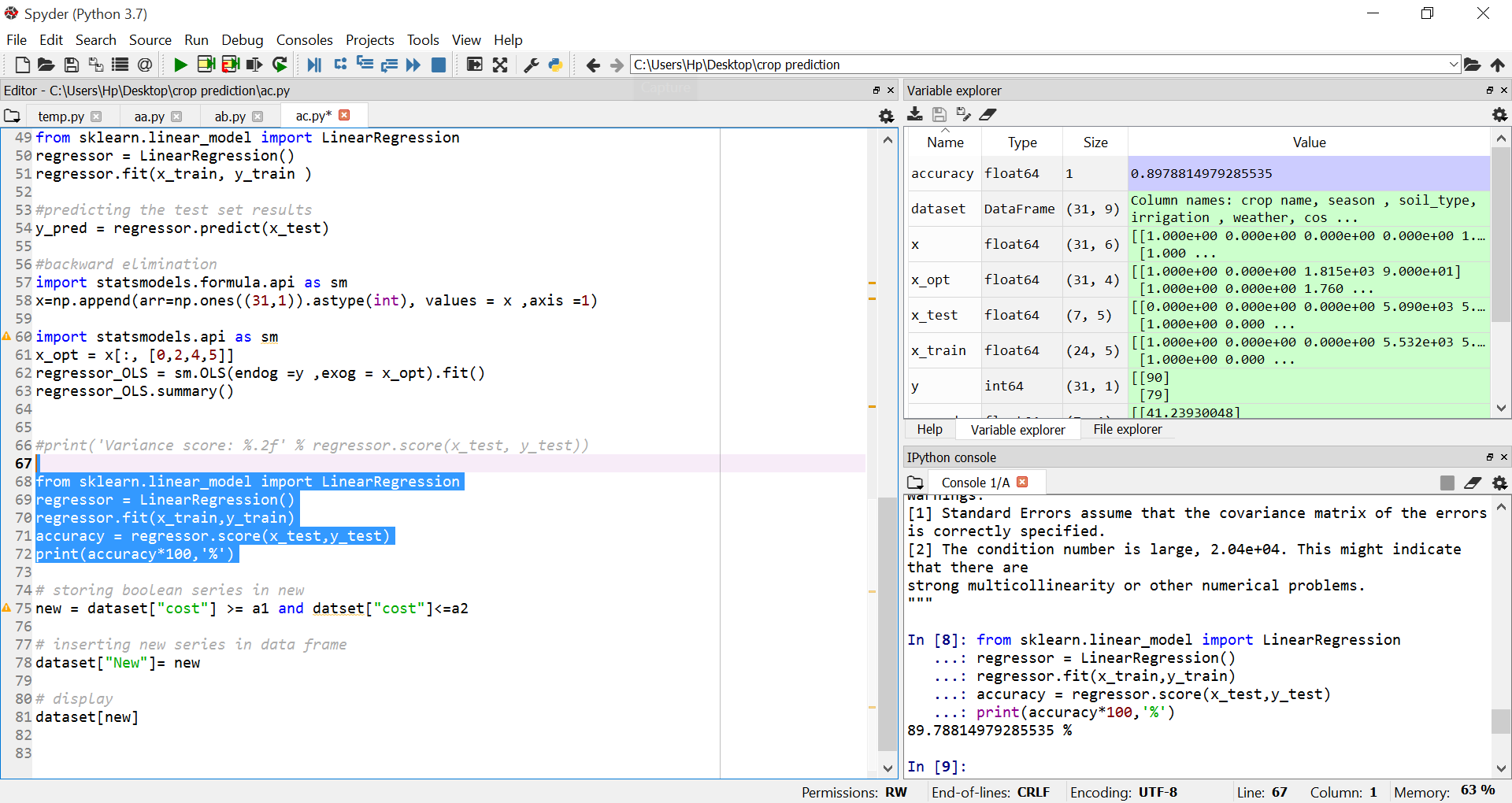
5.3 Effectiveness of model’s attributes

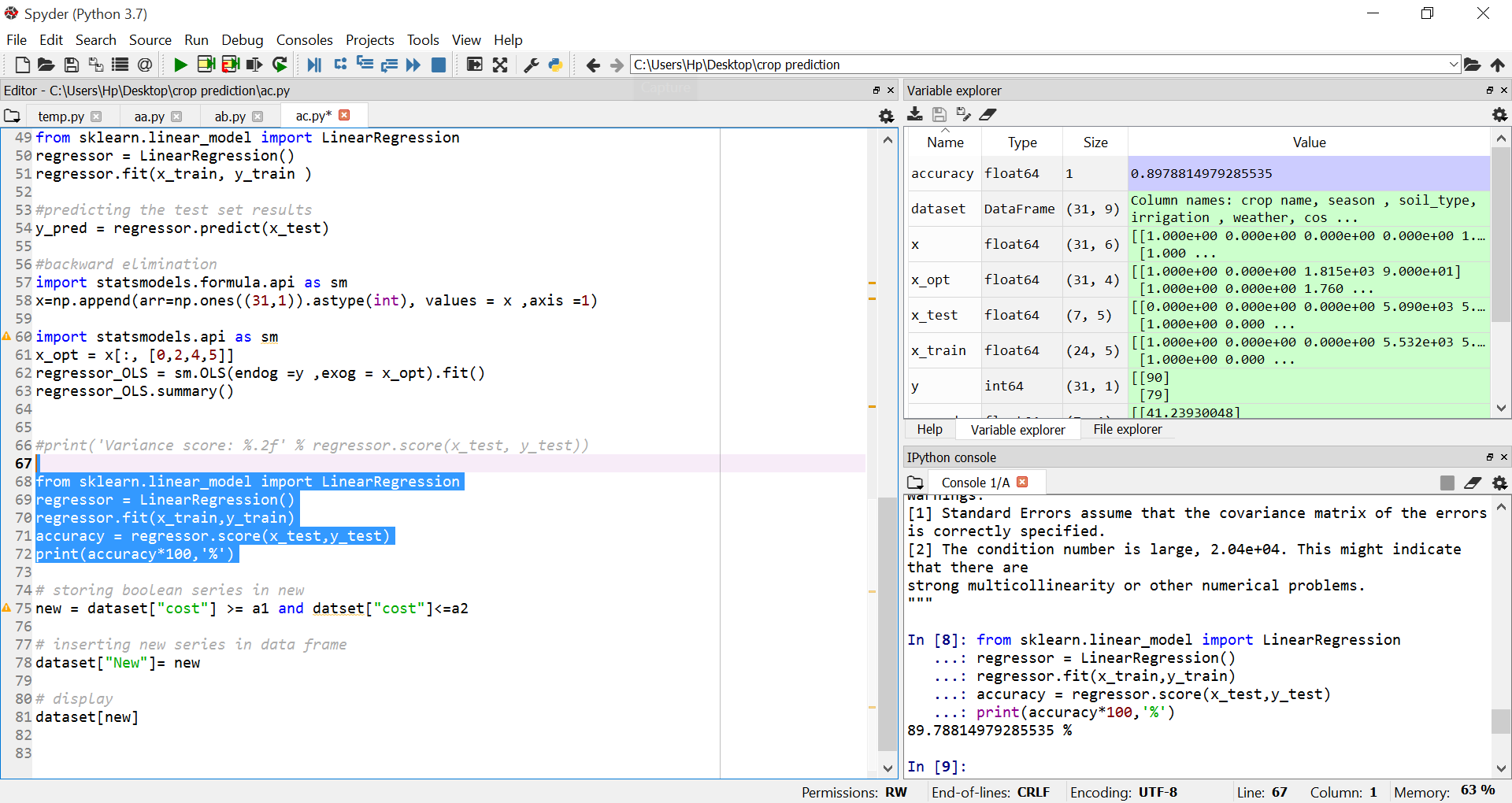
****

With significance level of 0.5 % the summarization is as follows :

The P value of all the three attributes used in predicting the result are less than 0.5 % .

5.4 Accuracy

****

****

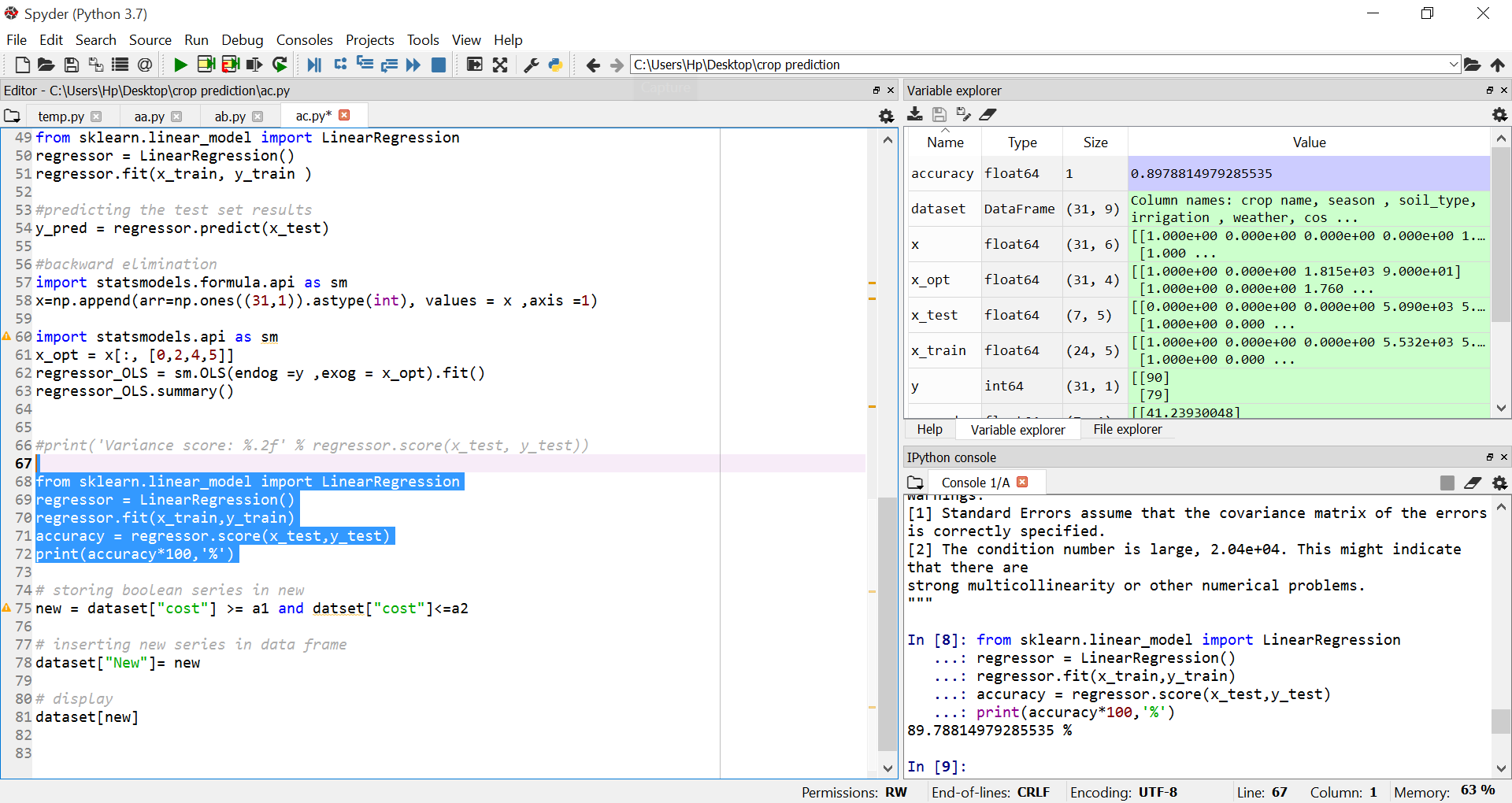
Accuracy comes out to be 89 % . and that’s means models has fitted well .

**CHAPTER 6**

**CONCLUSION AND FUTURE SCOPE**

* 1. **CONCLUSION**

Smart farmer’s assistant is a python console based application . this helps the farmers to select the suitable crop that fits his budget and in accorandance with crop season as well. This app has its own database and provides you the wide variety of crops.



Accuracy of model comes out 89 % . all parameter used are helfull in predicting the results

* 1. **FUTURE SCOPE**

In future we can determine the efficient algorithm based on their accuracy metrics that will helps to choose an efficient algorithm for crop yield prediction. We can predict crop yield by using more efficient

**6.2.1 Species management**

* **Species Breeding:** Our favourite, this application is so logical and yet so unexpected, because mostly you read about harvest prediction or ambient conditions management at later stages.

Species selection is a tedious process of searching for specific genes that determine the effectiveness of water and nutrients use, adaptation to climate change, disease resistance, as well as nutrients content or a better taste. Machine learning, in particular, deep learning algorithms, take decades of field data to analyze crops performance in various climates and new characteristics developed in the process. Based on this data they can build a probability model that would predict which genes will most likely contribute a beneficial trait to a plant.

* **Species Recognition:** While the traditional human approach for plant classification would be to compare color and shape of leaves, machine learning can provide more accurate and faster results analyzing the leaf vein morphology which carries more information about the leaf properties.
  + 1. **Field conditions management**
* **Soil management:** For specialists involved in agriculture, soil is a heterogeneous natural resource, with complex processes and vague mechanisms. Its temperature alone can give insights into the climate change effects on the regional yield. Machine learning algorithms study evaporation processes, soil moisture and temperature to understand the dynamics of ecosystems and the impingement in agriculture.
* **Water Management:** Water management in agriculture impacts hydrological, climatological, and agronomical balance. So far, the most developed ML-based applications are connected with estimation of daily, weekly, or monthly evapotranspiration allowing for a more effective use of irrigation systems and prediction of daily dew point temperature, which helps identify expected weather phenomena and estimate evapotranspiration and evaporation.
  + 1. **Crop management**
* **Yield Prediction:**Yield prediction is one of the most important and popular topics in precision agriculture as it defines yield mapping and estimation, matching of crop supply with demand, and crop management. State-of the-art approaches have gone far beyond simple prediction based on the historical data, but incorporate computer vision technologies to provide data on the go and comprehensive multidimensional analysis of crops, weather, and economic conditions to make the most of the yield for farmers and population.
* **Crop Quality:**The accurate detection and classification of crop quality characteristics can increase product price and reduce waste. In comparison with the human experts, machines can make use of seemingly meaningless data and interconnections to reveal new qualities playing role in the overall quality of the crops and to detect them.
* **Disease Detection:** Both in open-air and greenhouse conditions, the most widely used practice in pest and disease control is to uniformly spray pesticides over the cropping area. To be effective, this approach requires significant amounts of pesticides which results in a high financial and significant environmental cost. ML is used as a part of the general precision agriculture management, where agro-chemicals input is targeted in terms of time, place and affected plants.
* **Farmer’s Little Helper:** This is an application that can be called a bonus: imagine a farmer sitting late at night and trying to figure out the next steps in management of his crops. Whether he could sell more now to a local producer or head to a regional fair? He needs someone to talk through the various options to take a final decision. To help him, companies are now working on development specialized chatbots that would be able to converse with farmers and provide them with valuable facts and analytics. Farmers’ chatbots are expected to be even smarter than consumer-oriented Alexa and similar helpers, since they would be able not only to give figures, but analyze them and consult farmers on tough matters.