CROP PREDICTION WITH MACHINE LEARNING

Agriculture is one of the most important occupation practiced in our country. It is the broadest economic sector and plays an important role in overall development of the country. About 60 % of the land in the country is used for agriculture in order to suffice the needs of 1.2 billion people. Thus, modernization of agriculture is very important and thus will lead the farmers of our country towards profit.

Data analytic (DA) is the process of examining data sets in order to draw conclusions about the information they contain, increasingly with the aid of specialized systems and software. Earlier yield prediction was performed by considering the farmer's experience on a particular field and crop. However, as the conditions change day by day very rapidly, farmers are forced to cultivate more and more crops. Being this as the current situation, many of them don't have enough knowledge about the new crops and are not completely aware of the benefits they get while farming them. Also, the farm productivity can be increased by understanding and forecasting crop performance in a variety of environmental conditions. Thus, the proposed system takes the location of the user as an input. From the location, the nutrients of the soil such as Nitrogen, Phosphorous, Potassium is obtained. The processing part also take into consideration two more datasets i.e. one obtained from weather department, forecasting the weather expected in current year and the other data being static data. This static data is the crop production and data related to demands of various crops obtained from various government websites. The proposed system applies machine learning and prediction algorithm like Multiple Linear Regression to identify the pattern among data and then process it as per input conditions. This in turn will propose the best feasible crops according to given environmental conditions.Thus, this system will only require the location of the user and it will suggest number of profitable crops providing a choice directly to the farmer about which crop to cultivate. As past year production is also taken into account, the prediction will be more accurate.

Regression: Regression analysis is a form of predictive modelling technique which investigates the association between a dependent (targets) and autonomous variable (s) (independent variables).
Algorithmic Classification

Linear Regression:Linear regression is a linear methodology for demonstrating the link between a scalar dependent variable y and one or more independent variables denoted X. The instance of solitary independent variable is called simple linear regression.

Many Names of Linear Regression:

When you start looking into linear regression, things can get very confusing.

The reason is because linear regression has been around for so long (more than 200 years). It has been studied from every possible angle and often each angle has a new and different name.

Linear regression is a linear model, e.g. a model that assumes a linear relationship between the input variables (x) and the single output variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x).

When there is a single input variable (x), the method is referred to as simple linear regression. When there are multiple input variables, literature from statistics often refers to the method as multiple linear regression.

Different techniques can be used to prepare or train the linear regression equation from data, the most common of which is called Ordinary Least Squares. It is common to therefore refer to a model prepared this way as Ordinary Least Squares Linear Regression or just Least Squares Regression.

Now that we know some names used to describe linear regression, let's take a closer look at the representation used.

Linear Regression Model Representation
Linear regression is an attractive model because the representation is so simple.

The representation is a linear equation that combines a specific set of input values (x) the solution to which is the predicted output for that set of input values (y). As such, both the input values (x) and the output value are numeric.

The linear equation assigns one scale factor to each input value or column, called a coefficient and represented by the capital Greek letter Beta (B). One additional coefficient is also added, giving the line an additional degree of freedom (e.g. moving up and down on a two-dimensional plot) and is often called the intercept or the bias coefficient.

For example, in a simple regression problem (a single x and a single y), the form of the model would be:

$y = B0 + B1*x$

In higher dimensions when we have more than one input (x), the line is called a plane or a hyper-plane. The representation therefore is the form of the equation and the specific values used for the coefficients (e.g. B0 and B1 in the above example).

It is common to talk about the complexity of a regression model like linear regression. This refers to the number of coefficients used in the model.

When a coefficient becomes zero, it effectively removes the influence of the input variable on the model and therefore from the prediction made from the model (0 * x = 0). This becomes relevant if you look at regularization methods that change the learning algorithm to reduce the complexity of regression models by putting pressure on the absolute size of the coefficients, driving some to zero.

Now that we understand the representation used for a linear regression model, let's review some ways that we can learn this representation from data.

Linear Regression Learning the Model
Learning a linear regression model means estimating the values of the coefficients used in the representation with the data that we have available.

In this section we will take a brief look at four techniques to prepare a linear regression model. This is not enough information to implement them from scratch, but enough to get a flavor of the computation and trade-offs involved.

There are many more techniques because the model is so well studied. Take note of Ordinary Least Squares because it is the most common method used in general. Also take note of Gradient Descent as it is the most common technique taught in machine learning classes.

1. Simple Linear Regression
With simple linear regression when we have a single input, we can use statistics to estimate the coefficients.

This requires that you calculate statistical properties from the data such as means, standard deviations, correlations and covariance. All of the data must be available to traverse and calculate statistics.

This is fun as an exercise in excel, but not really useful in practice.

2. Ordinary Least Squares
When we have more than one input we can use Ordinary Least Squares to estimate the values of the coefficients.

The Ordinary Least Squares procedure seeks to minimize the sum of the squared residuals. This means that given a regression line through the data we calculate the distance from each data point to the regression line, square it, and sum all of the squared errors together. This is the quantity that ordinary least squares seeks to minimize.

This approach treats the data as a matrix and uses linear algebra operations to estimate the optimal values for the coefficients. It means that all of the data must be available and you must have enough memory to fit the data and perform matrix operations.

It is unusual to implement the Ordinary Least Squares procedure yourself unless as an exercise in linear algebra. It is more likely that you will call a procedure in a linear algebra library. This procedure is very fast to calculate.

3. Gradient Descent
When there are one or more inputs you can use a process of optimizing the values of the coefficients by iteratively minimizing the error of the model on your training data.

This operation is called Gradient Descent and works by starting with random values for each coefficient. The sum of the squared errors are calculated for each pair of input and output values. A learning rate is used as a scale factor and the coefficients are updated in the direction towards minimizing the error. The process is repeated until a minimum sum squared error is achieved or no further improvement is possible.

When using this method, you must select a learning rate (alpha) parameter that determines the size of the improvement step to take on each iteration of the procedure.

Gradient descent is often taught using a linear regression model because it is relatively straightforward to understand. In practice, it is useful when you have a very large dataset either in the number of rows or the number of columns that may not fit into memory.

4. Regularization
There are extensions of the training of the linear model called regularization methods. These seek to both minimize the sum of the squared error of the model on the training data (using ordinary least squares) but also to reduce the complexity of the model (like the number or absolute size of the sum of all coefficients in the model).

Two popular examples of regularization procedures for linear regression are:

Lasso Regression: where Ordinary Least Squares is modified to also minimize the absolute sum of the coefficients (called L1 regularization).
Ridge Regression: where Ordinary Least Squares is modified to also minimize the squared absolute sum of the coefficients (called L2 regularization).
These methods are effective to use when there is collinearity in your input values and ordinary least squares would overfit the training data.

Now that you know some techniques to learn the coefficients in a linear regression model, let’s look at how we can use a model to make predictions on new data.

Making Predictions with Linear Regression:

Given the representation is a linear equation, making predictions is as simple as solving the equation for a specific set of inputs.

Let's make this concrete with an example. Imagine we are predicting weight (y) from height (x). Our linear regression model representation for this problem would be:

y = B0 + B1 * x1

or

weight =B0 +B1 * height

Where B0 is the bias coefficient and B1 is the coefficient for the height column. We use a learning technique to find a good set of coefficient values. Once found, we can plug in different height values to predict the weight.

For example, lets use B0 = 0.1 and B1 = 0.5. Let's plug them in and calculate the weight (in kilograms) for a person with the height of 182 centimeters.

weight = 0.1 + 0.5 * 182

weight = 91.1

You can see that the above equation could be plotted as a line in two-dimensions. The B0 is our starting point regardless of what height we have. We can run through a bunch of heights from 100 to 250 centimeters and plug them to the equation and get weight values, creating our line.

Preparing Data For Linear Regression:

Linear regression is been studied at great length, and there is a lot of literature on how your data must be structured to make best use of the model.

As such, there is a lot of sophistication when talking about these requirements and expectations which can be intimidating. In practice, you can uses these rules more as rules of thumb when using Ordinary Least Squares Regression, the most common implementation of linear regression.

Try different preparations of your data using these heuristics and see what works best for your problem.

Linear Assumption. Linear regression assumes that the relationship between your input and output is linear. It does not support anything else. This may be obvious, but it is good to remember when you have a lot of attributes. You may need to transform data to make the relationship linear (e.g. log transform for an exponential relationship).
Remove Noise. Linear regression assumes that your input and output variables are not noisy. Consider using data cleaning operations that let you better expose and clarify the signal in your data. This is most important for the output variable and you want to remove outliers in the output variable (y) if possible.
Remove Collinearity. Linear regression will over-fit your data when you have highly correlated input variables. Consider calculating pairwise correlations for your input data and removing the most correlated.
Gaussian Distributions. Linear regression will make more reliable predictions if your input and output variables have a Gaussian distribution. You may get some benefit using transforms (e.g. log or BoxCox) on you variables to make their distribution more Gaussian looking.
Rescale Inputs: Linear regression will often make more reliable predictions if you rescale input variables using standardization or normalization

PLATFORM:SPYDER

CODE SNIPPET:

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

dataset = pd.read_csv('datasheetfinal.csv')
x = dataset.iloc[:, :-1].values
y = dataset.iloc[:, 2].values


from sklearn.preprocessing import LabelEncoder
labelencoder_y = LabelEncoder()
y = labelencoder_y.fit_transform(y)


"""from sklearn.preprocessing import StandardScaler
sc_y = StandardScaler()
y_train = sc_x.fit_transform(y_train)
y_test = sc_x.transform(y_test)"""
```

```
from sklearn.cross_validation import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x, y, test_size = 0.8, random_state = 0)

"""from sklearn.preprocessing import StandardScaler
sc_x = StandardScaler()
x_train = sc_x.fit_transform(x_train)
x_test = sc_x.transform(x_test)"""

from sklearn.linear_model import LinearRegression
regressor =  LinearRegression()
regressor.fit(x_train,y_train)

y_pred = regressor.predict(x_test)

values = np.array([y_pred])
index_max = np.argmin(values)
number=index_max
print(number)


filename = 'finalized_model.sav'
pickle.dump(regressor, open(filename, 'wb'))


loaded_regressor = pickle.load(open(filename, 'rb'))
result = loaded_regressor.score(x_test, y_test)
print(result)

values = np.array([y_pred])
index_max = np.argmin(values)
number=index_max
print(number)


def crop_name (number):
    if number == 1:
        return "Garbera"
    elif number == 2:
        return "Jasmine"
    elif number == 3:
        return "Barley"
```

```
    elif number == 4:
        return "Wheat"
    elif number == 5:
        return "Pumpkin"
    elif number == 6:
        return "French Bean"
    elif number == 7:
        return "Marigold"
    elif number == 8:
        return "Caunation"
    elif number == 9:
        return "Field pea"
    elif number == 10:
        return "Sugarcane"
    elif number == 11:
        return "Sunhemp"

    crop= crop_name (number)
    print(crop)


import smtplib
def send_email(subject, msg):
    try:
        server = smtplib.SMTP('smtp.gmail.com:587')
        server.ehlo()
        server.starttls()
        server.login('iotwithml@gmail.com','amartyaanirban')
        message = 'Subject: {}\n\n{}'.format(subject, msg)
        server.sendmail('iotwithml@gmail.com','amartyaroy1998@gmail.com', message)
        server.quit()
        print("Success: Email sent!")
    except:
        print("Email failed to send.")
subject = "Test "
msg =  crop
send_email(subject, msg)
```

CONCLUSION:

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 the

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.