INDIRA GANDHI DELHI TECHNICAL UNIVERSITY FOR WOMEN

Computer Science & Engineering Department

DATA WAREHOUSING AND DATA MINING

PROJECT REPORT

Submitted To:
Ms. Divanshi Wangoo
CSE department

Submitted By:
Group 3:
Sakshi Vats (087)
Medha Bhattacharya (100)
Sirshasree Tripathy (109)
Aakanksha Singh (156)
(CSE- 2, 5th sem)

Predicting the next water cycle for agricultural pumping systems using Data Mining techniques

PROBLEM STATEMENT

Water supply is one of the main components of agriculture and farming. Till now many farmers depend on manual watering. In general households also, people rely on a friend to water their plants when they are out of town. There is a requirement of a system that can irrigate crops regularly using an automated water pumping system for commercial agriculture as well as in household gardens.

AIM

The aim of our project is to generate an automated system that decides when to turn on or off the water supply in a farm irrigation system by taking into consideration the moisture content of soil. This would make the agricultural process more efficient, allowing workers to concentrate on other important farming tasks and minimizing the water wastage.

DATA MINING SOLUTION AND ALGORITHM

To build an automated water pumping system, it is necessary to predict the next watering cycle for the crops i.e. deciding on the number of days after which the farm has to be watered again. Taking into account the dataset describing the moisture content of the soil, we used regression analysis for this task. Regression is a data mining technique used to predict a range of numeric values, given a particular dataset. Using the datasets, we are computing the number of days after which the farm has to be watered. Regression analysis is mainly used for forecasting an effect, and trend forecasting. We first determined that there is some significant association between the variables of interest. As a result, linear regression was chosen as the appropriate data mining algorithm to measure the amount of water required and to predict the next water cycle.

Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable.
A linear regression line has an equation of the form $Y = a + bX$, where X is the explanatory variable and Y is the dependent variable. The slope of the line is b, and a is the intercept (the value of y when $x = 0$). The most common method for fitting a regression line is the method of least-squares. This method calculates the best-fitting line for the observed data 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.

Using the above model, there are chances of changing the frequency of the irrigation process. Along with it we can start and stop the water cycle exactly when required, thus minimising energy consumption.

DATASET USED

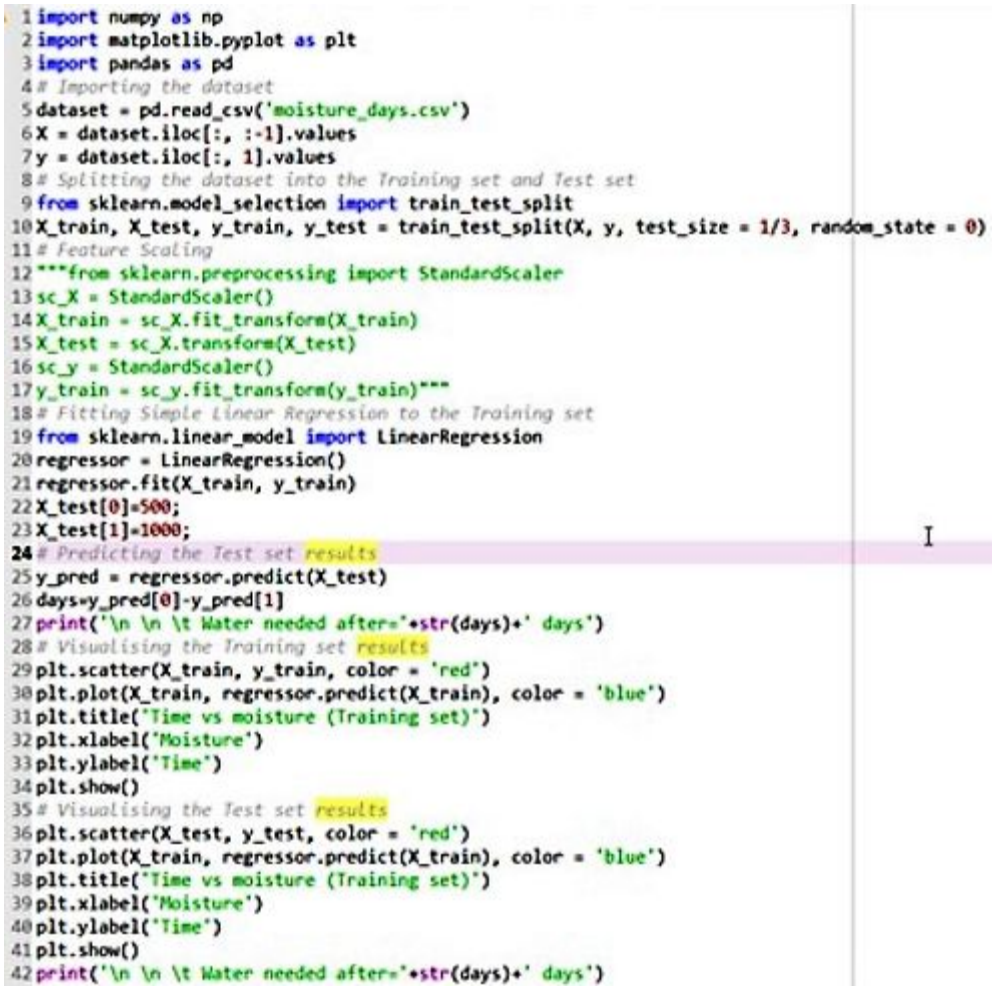
MoistureContentCycle.csv
This dataset contains different instances having two attributes - **MOISTURE** : depicting the moisture content present in the soil, and **DAYS** : describing the number of days that moisture content remains in the soil.
LINK : https://github.com/iAmOffended/supremeBassoon

IMPLEMENTATION

Python code for predicting next water cycle using linear regression:

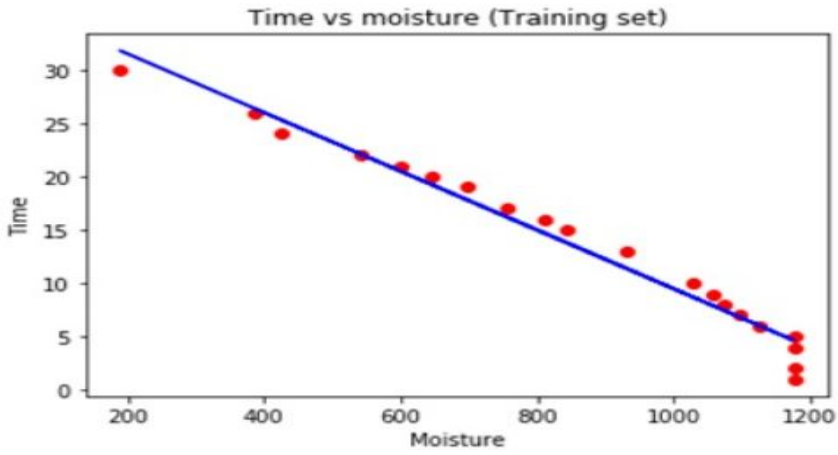
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
dataset                                                                    =
pd.read_csv("C://Users//hp//Downloads//MoistureContentCycle.csv")
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, 1].values

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
1/3, random_state = 0)

#Splitting the DataSet into the following Training Set and Test Set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
1/3, random_state = 0)

#Feature Scaling
from sklearn.preprocessing import StandardScaler
sc_x = StandardScaler()
X_train = sc_x.fit_transform(X_train)
X_test = sc_x.transform(X_test)
sc_y = StandardScaler()
y_train = sc_y.fit_transform(y_train)

#Fitting Simple Linear Regression to the Training Set
regressor = LinearRegression()
regressor.fit(x_train, y_train)
X_test[0] = 500;
X_test[1] = 1000;
```

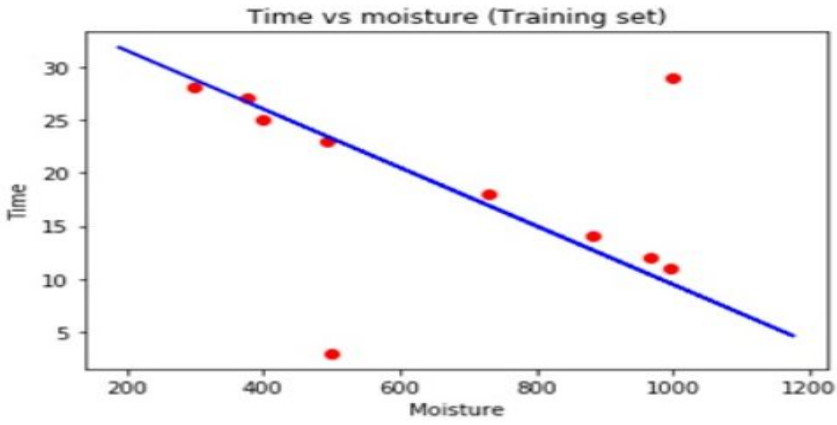
```
#Predicting the Test Set results
y_pred = regressor.predict(x_test)
days = y_pred[1]
print('\n \n \n \t Water Needed After = '+str(days)+' days')

#Visualising the Training Set Results
plt.scatter(X_train, y_train, color= 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Time vs Moisture(Training Set)')
plt.xlabel('Moisture')
plt.ylabel('Time')
plt.show()

#Visualising the Test set results
plt.scatter(X_test, y_test, color= 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Time vs Moisture(Training Set)')
plt.xlabel('Moisture')
plt.ylabel('Time')
plt.show()
print('\n \n \n \t Water Needed After = '+str(days)+' days'
```



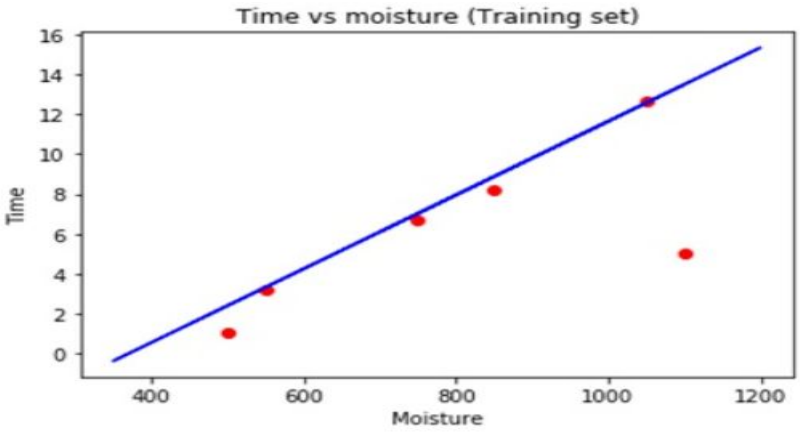
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
# Importing the dataset
dataset = pd.read_csv('moisture_days.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, 1].values
# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 1/3, random_state = 0)
# Feature Scaling
"""from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
sc_y = StandardScaler()
y_train = sc_y.fit_transform(y_train)"""
# Fitting Simple Linear Regression to the Training set
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
X_test[0]=500;
X_test[1]=1000;
# Predicting the Test set results
y_pred = regressor.predict(X_test)
days=y_pred[0]-y_pred[1]
print('\n \n \t Water needed after='+str(days)+' days')
# Visualising the Training set results
plt.scatter(X_train, y_train, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Time vs moisture (Training set)')
plt.xlabel('Moisture')
plt.ylabel('Time')
plt.show()
# Visualising the Test set results
plt.scatter(X_test, y_test, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Time vs moisture (Training set)')
plt.xlabel('Moisture')
plt.ylabel('Time')
plt.show()
print('\n \n \t Water needed after='+str(days)+' days')
```

OUTPUT ANALYSIS






```
Water needed after=13.76961813881012 days
```




```
Water needed=2335.2153987167735 liters
```

By applying linear regression on the dataset MoistureContentCycle.csv, which describes the moisture content of soil, we are able to visualize and predict the minimum number of days after which we will have to water the soil.

FUTURE SCOPE

To enhance our model, we can further predict the soil dryness using the Naive Bayes algorithm. This is an intuitive method that uses the probabilities of each attribute belonging to each class to make a prediction. It simplifies the calculation of probabilities by assuming that the probability of each attribute belonging to a given class value is independent of all other attributes. This is a strong assumption but results in a fast and effective method. The probability of a class value given a value of an attribute is called the conditional probability. By multiplying the conditional probabilities together for each attribute for a given class value, we have a probability of a data instance belonging to that class. To make a prediction we can calculate probabilities of the instance belonging to each class and select the class value with the highest probability.

REFERENCES

1. https://www.statisticssolutions.com/what-is-linear-regression/?__cf_chl_jschl_tk__=bcc29fc912feacbc7167a23f8830249e9ac9601c-1607784917-0-AUMZyN0iXNRqICpPWHj_smTGx89ERHJhORTni3wADLi77Hb6
2. https://sswm.info/sswm-university-course/module-4-sustainable-water-supply/further-resources-water-use/automatic-irrigation#:~:text=Executive%20Summary,or%20computers%20or%20mechanical%20appliances.
3. http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm#:~:text=Linear%20regression%20attempts%20to%20model,to%20be%20a%20dependent%20variable.
4. https://www.lifewire.com/regression-1019655
5. https://en.wikipedia.org/wiki/Naive_Bayes_classifier
6. https://github.com/iAmOffended/supremeBassoon

INDIVIDUAL CONTRIBUTIONS

1. **Sakshi Vats (08701012018) :** Background research and creation of problem statement, finding relevant data mining algorithms to implement our project, designing the code, and output analysis.

2. **Medha Bhattacharya (10001012018) :** Finding the relevant datasets and appropriate data mining algorithms to implement our project, implementing and debugging the code, and output analysis.

3. **Sirshasree Tripathy (10901012018) :** Finding the relevant datasets and appropriate data mining algorithms to implement our project, implementing and debugging the code, and output analysis.

4. **Aakanksha Singh (15601012018) :** Background research and creation of problem statement, finding relevant data mining algorithms to implement our project, designing the code, and output analysis.