Supervised Machine Learning Semester Project

Group members: Joshua Brobst, Emma Morse

1a). Our chosen supervised machine learning algorithm is Multivariate Logistic Regression. Since the chosen dataset is not a high dimensional dataset we don't anticipate the need for regularization. However, the dataset is unbalanced (with more than 75% of the dataset falling into a single response category), so we will need to explore strategies to mitigate this, such as over sampling or under sampling.

1b). Our chosen new supervised machine learning algorithm is Decision Trees. We wanted to look at this algorithm as it can help with unbalanced datasets, which may mitigate the amount of over/under sampling we have to perform.

1c). Our chosen dataset is from Plant-Health-Monitoring, although we are dropping the 'Health Score' column. The goal of the dataset is to measure the health of a plant (specifically, if it is 'healthy' or 'unhealthy') based on the environment around it. With this dataset, we will have the response variable be the binary indicator 'Plant Health', where a 0 represents an unhealthy plant and a 1 represents a healthy plant. The temperature covariate measures the temperature around the plant, measured in Celsius; the humidity % measures the moisture concentration in the air around the plant; soil moisture % measures the moisture in the soil of the plant; soil pH measures pH levels of the soil, with higher soil levels negatively impacting the availability of nutrients; nutrient levels, the actual levels of nutrients; light intensity, in lux, that the plant has direct exposure to. We are dropping the health score column as the data card did not list what was used to determine the score. The first column, Plant_ID, acts exclusively as an index and goes from Plant_1 to Plant_1000. All 1000 samples are complete, without any N/A or NaN values.

Milestone 2:

Step 2: Preliminary analysis. Analyze your dataset using existing R or Python libraries and packages. Groups should analyze the data using both the existing method and the new method.

- A two-page description of your analysis and what you learned about the population from which the data was sampled.
- A bibliography, in progress.
- R/Python script.
- A csv file of the dataset.

Milestone 2: Logistic Regression and Decision Tree Analysis on Plant Health Data

Exploratory data analysis

This dataset has six numerical variables and a response factor with levels 0 and 1. The variables are all normally distributed and have little to no correlation. The response factor on the other hand is very imbalanced with the 0 class having 174 observations and the 1 class having 826 observations.

According to the dataset documentation [1] the variables likely have interaction terms for example: Humidity is `crucial for plant transpiration and water intake` so likely has an effect in combination with Soil_Moisture. This will need to be explored later.

Logistic Regression

A Logistic regression model was created using all numerical variables following the formular: Health_Status ~ Temperature_C + Humidity + Soil_Moisture + Soil_pH + Nutrient_Level + Light_Intensity_lux. All variables were used as a first pass of the model to find any preliminary patterns and better understand the obstacles we may face with this dataset. With a significance level of a = 90, this model did not find any of the variables to be statistically significant predictors of Health_Status, except Temperature_C which has a P-value of 0.079.

To validate this model, we used a simple 80-20 train test split. The confusion matrix indicates that the unbalanced nature of the dataset is a problem with this method:

	0	1
0	0	0
1	27	173

Table 1. Confusion matrix for the logistic regression model using a 80-20 train test split and the formula Health_Status ~ Temperature_C + Humidity + Soil_Moisture + Soil_pH + Nutrient_Level + Light_Intensity_lux.

The model fails to predict any 0's in the response, likely due to the dataset being unbalanced.

Decision Tree

We also fitted a decision tree model following the same formula. This approach ended up with a very similar confusion matrix, once again due to the imbalanced nature of the dataset.

```
##       0   1

## 0   0   1

## 1 24 175
```

Table 2. Confusion matrix for the decision tree model using a 80-20 train test split and the formula Health_Status ~ Temperature_C + Humidity + Soil_Moisture + Soil_pH + Nutrient_Level + Light_Intensity_lux.

This makes sense, as decision trees may not always be effective when there is an overwhelming majority class[1]. This furthers the need for undersampling the majority class in the dataset to be able to fit a precise model

Next steps

1. Use under sampling to balance the population sample and re-attempt the 2 models.
2. Consider creating interaction terms logically according to the variable documentation provided and re-attempt the 2 models.

Bibliography

1. https://www.kaggle.com/datasets/ziya07/plant-health-monitoring
2. How do decision trees work with unbalanced datasets? | GeeksforGeeks
3. emgrotto/ds5220-plant-health: Home for the ds5220 final project.

[1] How do decision trees work with unbalanced datasets? | GeeksforGeeks