Machine Learning and Predictive Analysis Assessment

Crop Classification from Soil Characteristics

Daniel Sean Brown

***Abstract:***

**This project aimed to create a classification model for suggesting the ideal crop to grow given a set of soil characteristics. This was achieved through a variety of supervised learning models and further work was done to test how models fair with distorted data. Coefficient analysis also revealed which soil characteristics affect which crops and how.**

Table of Contents

[I. Introduction 1](#_Toc198768823)

[A. Context 1](#_Toc198768824)

[B. Aims and Goals 1](#_Toc198768825)

[II. Exploratory Data Analysis 1](#_Toc198768826)

[A. Data Profile 1](#_Toc198768827)

[B. Data Cleaning 1](#_Toc198768828)

[Data cleaning was quick and smooth with no duplicates and no null values. 1](#_Toc198768829)

[C. Data Exploration and Visualisation 1](#_Toc198768830)

[Having gained a strong foundational understanding of the dataset, outlier removal was conducted next. 2](#_Toc198768831)

[D. Outlier Removal 2](#_Toc198768832)

[III. Preliminary Plan 2](#_Toc198768833)

[A. Logistic Regression 2](#_Toc198768834)

[B. Further Supervised Learning 2](#_Toc198768835)

[C. Model Robustness 3](#_Toc198768836)

[D. Data Simulation 3](#_Toc198768837)

[E. Clustering 3](#_Toc198768838)

[IV. Results 3](#_Toc198768839)

[A. Logistic Regression 3](#_Toc198768840)

[1) Initial Regressor 3](#_Toc198768841)

[2) Regularisation 3](#_Toc198768842)

[3) Coeffcient Analysis 3](#_Toc198768843)

[B. Further Supervised Learning Models 4](#_Toc198768844)

[C. Noise/Null Value Injection 4](#_Toc198768845)

[D. Coefficient Simulation 5](#_Toc198768846)

[E. Clustering 5](#_Toc198768847)

[V. Discussion of Outcomes 5](#_Toc198768848)

[VI. References 5](#_Toc198768849)

# Introduction

## Context

Crop classification is an increasingly important task. With changing climates and environments, traditional knowledge of land may become outdated. By being able to make best use of land, not only can yield be increased but also loss and waste can be minimised. This leading to higher food security, profits and being more environmentally sustainable means high performance models could be a crucial part of informed agricultural decision making – assisting not only farmers, but also environmentalists and policy makers.

## Aims and Goals

The aim of this project is to:

G1. Construct a high-performance model for crop classification.

G2. To interpret models for understanding which properties of soil are most crucial for differing crops.

# Exploratory Data Analysis

Supplementary code for EDA should be available in the zip file within which this report is uploaded. The jupyterlabs markdown file contains all the python scripts used to conduct analysis for this report. Alongside the markdown file and report, the dataset is also included in the zip file – this enables easy running of the script should one wish to. A Kaggle link is alternatively also provided.

## Data Profile

The dataset consists of 2200 rows, seven continuous features and one categorical target column. These feature columns are nitrogen, phosphorus, potassium, temperature, humidity, ph and rainfall. The target column consists of 22 types of crops, these being: rice, maize, jute, cotton, coconut, papaya, orange, apple, muskmelon, watermelon, grapes, mango, banana, pomegranate, lentil, blackgram, mungbean, mothbeans, pigeonpeas, kidneybeans, chickpea and coffee. The dataset is balanced, giving each crop 100 records of data. It is publicly available at:

<https://www.kaggle.com/datasets/madhuraatmarambhagat/crop-recommendation-dataset>

## Data Cleaning

### Data cleaning was quick and smooth with no duplicates and no null values.

## Data Exploration and Visualisation

To begin with feature histograms were plotted to observe distributions. Here temperature and ph were seen to be normally distributed.

**Figure 1. Feature Histograms**

A graph of a graph

AI-generated content may be incorrect.

Next, correlation heatmap is plotted to observe relationships between features. Here it can be seen that most features have little to no correlation between each other - except for potassium and phosphorus which have a significant positive correlation. To explore further this relationship is plotted.

**Figure 2. Feature Correlations Heatmap**

A graph of heatmap

AI-generated content may be incorrect.

Here it is observed how distinct classes are, with the distribution of points forming clear, separable rectangles. The rectangle distributions seem to exclusively have their ‘shorter side’ facing the potassium axis. This suggests that crops are overall more sensitive to potassium levels rather than phosphorus. Also, the correlation may be in part due to grapes and apples requiring higher levels of the two minerals than other crops.

**Figure. 3 Phosphorus and Potassium Scatterplot**

A screen shot of a graph

AI-generated content may be incorrect.

The distinct class groupings prompted further exploration of distributions within scatterplots. This, and the need for more class specific understanding of the overarching dataset motivated creating a pair plot.

Here similar distributions are again observed. The long and thin shape of rectangles suggest that some features may be much more important for classification than others.

**Figure 4. Pairwise Scatterplot of Dataset**

A screenshot of a graph

AI-generated content may be incorrect.

Finally, PCA is attempted to gain 2-D projection of the dataset for a summary visualisation. Sum of variance on first two PCs was seen to be more than 50% - so projection was deemed valid.

Here again, apples and grapes can be seen to lie further out than the other crops. Tight class groupings are also observed, staying consistent with previous pairwise scatterplot.

**Figure 5. PCA Results on First Two Principal Components**

A graph with many colored dots

AI-generated content may be incorrect.

## Having gained a strong foundational understanding of the dataset, outlier removal was conducted next.

## Outlier Removal

Outlier detection was approached with caution due to fear of information loss. EDA revealed distribution per class to be non-gaussian. Therefore z-values could not be used so instead inter quartile range was used for features on a per label basis. This however did not detect any outliers and with data seemingly consistent during EDA, this was deemed sufficient.

# Preliminary Plan

Model evaluation will primarily focus on F1 score and then on precision and recall, since this is a multinomial classification problem.

## Logistic Regression

An initial logistic regressor will be built as it is quick and highly interpretable. This model will then be evaluated for some initial insights and then improved using Lasso, Ridge and Elastic Net regularisation – the best performing one being then selected to further analyse model coefficients. Though not necessarily a high-performance model, logistic regression can work well – especially when groupings are easily separable as observed during EDA. This section is aimed to assist G2.

On logistic regression, it is worth quickly noting that for multinomial classification scikit-learn does not perform repeated One vs All binary classification[1]. Instead the softmax function is used for true multinomial classification. This is mentioned for help with coefficient interpretation.

## Further Supervised Learning

Next other supervised learning models will be explored. As well as robust, high performance models such as support vector and random forest classifiers, the nature of the dataset should work well with k nearest neighbour and naïve bayes models as well (due to low dimensionality and largely independent feature set). Other tree-based models will also be used, these being Gradient Boost and XGBoost. Hyperparameter tuning will be used on all models. This section contains traditionally strong models and should serve to complete G1 the most, however some interpretation could serve to help with G2. Deep learning methods will not be explored due to limited interpretability.

## Model Robustness

To simulate real world uncertainty, noise injection and arbitrary missing values will be used to create further datasets. These will then be used to examine model robustness and assess best strategies for overcoming them. This is aimed at trying to detect or rather reduce overfitting as well as finding the most robust model.

## Data Simulation

Bootstrapping (random selection with replacement) will be used to generate datasets for further logistic regressors. Their coefficients and model metrics will be extracted for confirmation of performance as well as to perform clustering on coefficients.

## Clustering

Finally, simulated coefficients will have clustering performed on them to assess model stability as well as to cluster labels by coefficients. This can then be used to examine groupings of crops by how similar they are in terms of their ideal environments.

# Results

## Logistic Regression

### Initial Regressor

Initial model performed surprisingly well during five-fold cross validation with an average F1 score of 0.964 and standard deviation of 0.017. Classification was accurate on all crops with significant drops only occurring on Jute, Rice, Papaya and Moth Beans.

**Figure 6. Initial Logistic Regression Metrics**

A graph of a logistic regression

AI-generated content may be incorrect.

### Regularisation

Nested hyperparameter tuning was then performed on regularisation techniques with five-fold cross validation again, using the initial model. Elastic net regularisation performed the best and was evaluated using the unseen validation set. Note parameter tuning picked an alpha value of 0.8, meaning that regularisation favoured Lasso significantly over Ridge. The largest available cost parameter was also chosen (10), which being an inverse of the lambda cost variable means that smaller regularisation was favoured.

On model metrics, these all improved slightly over initial model. Noting also that effects of overfitting should be reduced due to regularisation smoothing decision boundaries.

**Figure 7. Elastic Net Logistic Regression Metrics**

A screenshot of a graph

AI-generated content may be incorrect.

### Coeffcient Analysis

Following this, regularised model coefficients were used to produce a heatmap demonstrating which coefficients were shrunk to zero as well as how soil conditions effect crop suitability. Summed model coefficients here revealed that soil temperature and ph play a relatively small role in classification compared to the other features. Zero coefficients were blacked out for immediacy on which crops require fewer features for accurate classification. Note however, this is not equivalent to that crop being the most robust, often crops with binned features still had significant coefficients on other features. It should also be noted for visual interpretation that model pipeline included data standardisation.

**Figure 8. Coefficient Heatmap for Regularised Model**

A screenshot of a computer screen

AI-generated content may be incorrect.

## Further Supervised Learning Models

Performance was again consistently high across all models with a lowest F1 score of 0.9787 from KNN and highest of 0.9970 from Random Forest and XGBoost. Model metrics were plotted against each other, where consistent high performance can be seen again. Optimal hyperparameters were also observed, these however will be discussed during noise/null injection for the most robust model.

**Figure 9. Model Metrics for Tuned Learning Models**

A screenshot of a color chart

AI-generated content may be incorrect.

## Noise/Null Value Injection

Though the EDA would suggest that classes are clearly separable, exceedingly high model performance continued to fuel fears of overfitting. This highlighted the importance of model robustness further.

The tuned models all performed relatively well and difference in performance seemed to come more from imputation technique. Average F1 scores were plotted by imputation technique. To clarify, that is averaged across all noise and missingness levels.

**Figure 10. Model Performance by Imputation Technique**

A chart of a number of colors

AI-generated content may be incorrect.

F1 scores were also plotted to show performance change as noise levels increased. Note here that noise level refers to the standard deviation of values generated during dataset augmentation.

The figure here shows tree-based models to handle noise the best with Random Forest, XGBoost and Gradient Boosting finishing the highest. On the contrary, the Support Vector Classifier saw the steepest decline in performance. On the Random Forest model, it was the only model to maintain scores above 0.9.

**Figure 11. Model Performance by Noise Level.**

A graph of different colored lines

AI-generated content may be incorrect.

Model performance was also plotted specifically for KNN imputation, following its clear lead over other techniques. Here it was again observed that Random Forest performerd the best – although with a more steep decline compared to noise. Still, with roughly 20% of the data missing, F1 metric was above 0.85.

**Figure 12. Model Performance by Missingness Level**

A graph of different colored lines

AI-generated content may be incorrect.

To conclude on robustness, Random Forest comes out as the clear best choice – alongside KNN for dealing with null values. This is surprising as tuned hyperparameters suggested that Random Forest had overfit with a maximum depth of 20minimum samples on a lead of 1, seeing as deep and thin trees usually perform better at the cost of generalisation. Nevertheless, it can be said that G2 was achieved here.

## Coefficient Simulation

Bootstrapping simulation was performed faster than expected. F1 Scores were calculated for each sample model, and these were then subsequently plotted. Here high performance, gaussian distribution and low spread can be observed. This all suggest good model stability on the logistic regressor, giving further credence to its coefficient’s trustworthiness.

**Figure 13. F1 Scores for Bootstrapped Logistic Regressor Models**

A graph of a distribution of a number of scores

AI-generated content may be incorrect.

## Clustering

Initial hierarchical clustering was performed on averaged coefficients per class and feature. This produced a clear dendrogram, showing crop groupings by coefficient similarity. Here crop groupings can be split depending on chosen level to get clusters of crops by most similar ideal conditions.

**Figure 14. Hierarchical Clustering on Simulated Coefficients**

A graph with different colored lines

AI-generated content may be incorrect.

The dendrogram here when split for 4 groups gives the following clusters:

**Table 1. Coefficient Clusters**

| Crops by Most Similar Ideal Conditions | | | |
| --- | --- | --- | --- |
| ***Cluster 1*** | Cluster 2 | Cluster 3 | Cluster 4 |
| Apple | Grapes | Pomegranate | Maize |
| Banana | Watermelon | Chickpea | Rice |
| Muskmelon | Cotton | Coconut | Papaya |
|  | Coffee | Moth Beans | Orange |
|  | Kidney Beans | Mung Beans | Pigeon Peas |
|  | Black Gram | Mango | Lentil |
|  |  |  | Jute |

# Discussion of Outcomes

Overall, objectives were met with the tuned Random Forest Classifier providing a high performance and robust model for classification (G1) and the regularised logistic regressor giving insight into exactly which soil properties benefit which crop (G2). Clustering also gave insight into which crops are most similar for ideal conditions, which could serve to suggest alternative crops when one is unavailable for comparable soil characteristics.

Main challenges came from the large class types, these making visualisations often unwieldly and difficult to interpret. An improvement could be to restrict classes to fewer crop types – perhaps focusing on a specific food group.

To extend, more focus could be given to specific similarities in conditions on a feature level and some sort of yield prediction could be performed from further data.

# References

[1] - scikit-learn developers, 2024. *sklearn.linear\_model.LogisticRegression*. [online] scikit-learn. Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html> [Accessed 15 May 2025]

**(2188 words)**