Portfolio Task 4 Report: AI Model Development and Analysis

AI Model Development for Vegemite Production Consistency

Name: Arnob Ghosh

Student Number: 103494114

Course: COS40007 Artificial Intelligence for Engineering

Date of Submission: 25/08/24

Studio Class: Monday 4:30 – 6:30 PM

Project File: Portfolio Task 4

1. Introduction

Using data from machine processes and settings, the task's goal was to create and assess an AI model that can forecast vegemite production consistency. More than 15,000 data points make up the dataset, which is divided into three goal classes that correspond to varying degrees of consistency. My objectives were to examine and evaluate the data, identify the most pertinent features, train several machine learning models, and then decide which model performed the best so that it could be examined in more detail. In addition, I wanted to use the chosen model to create rules that would advise human operators on the best way to configure the machine.

2. Data Preparation

To begin, I prepared the dataset by performing the following steps:

1. Shuffling and Splitting the Data:

After randomly selecting 1,000 data points for testing, I shuffled the dataset to make sure there was a near-equal distribution for each class. The machine learning models were trained using the remaining 14,000+ data points.

Screenshot:

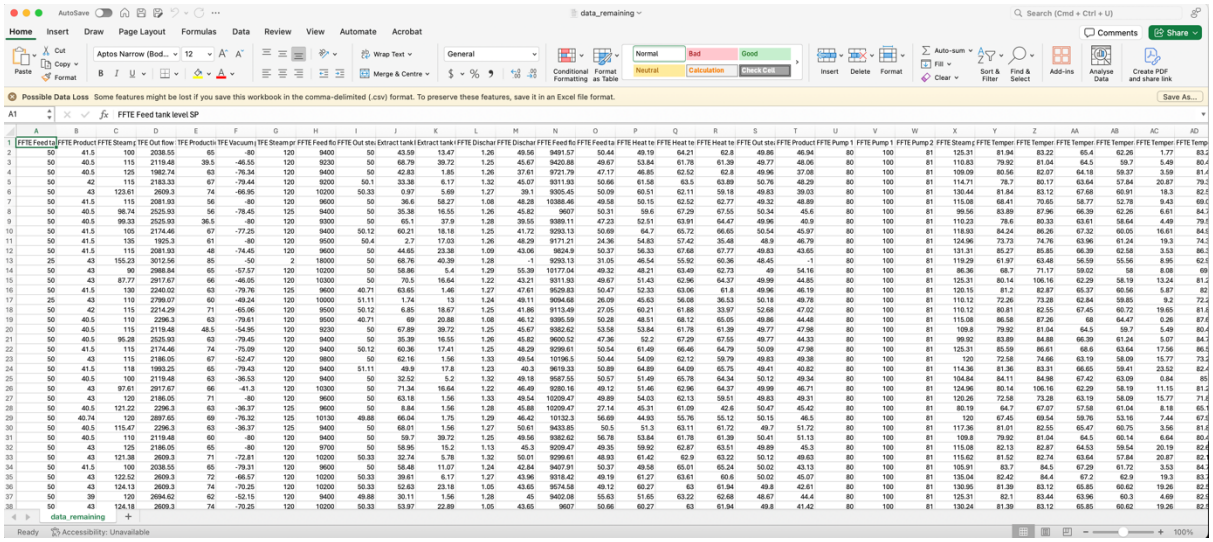

Fig: shuffled dataset (data_remaining.csv).

2. Data Cleaning:

Any constant value columns that I found in the dataset were eliminated because they don't offer useful data for model training. Additionally, I located and transformed any numeric columns into categorical features that had few unique values.

3. Class Balancing:

Where appropriate, I used undersampling and oversampling techniques to resolve any class imbalance. This made sure that every class in the training dataset had an equal distribution.

Screenshot:

```
/usr/local/bin/python3.12 /Users/ag47/Documents/Swinburne Class/AI for Engineering/Week 4/Portfolio Task 4/AI Model/data_preparation.py
Class Distribution After Balancing:
Class
0    2325
1    2325
2    2325
Name: count, dtype: int64
```

Fig: class distribution before and after balancing.

4. Composite Feature Creation:

I found a composite characteristic that might be instructive through data research. To construct a new composite feature, for instance, I multiplied the columns for `FFTE Production solids SP} and `FFTE Feed tank level SP}.

```
/usr/local/bin/python3.12 /Users/ag47/Documents/Swinburne Class/AI for Engineering/Week 4/Portfolio Task 4/AI Model/Dataset with Composite Feature.py
Dataset with Composite Feature:
   FFTE Feed tank level SP  FFTE Production solids SP  composite_feature
0                     50.0                       41.5             2075.0
1                     50.0                       40.5             2025.0
2                     50.0                       40.5             2025.0
3                     50.0                       42.0             2100.0
4                     50.0                       43.0             2150.0

Process finished with exit code 0
```

Screenshot: dataset with the new composite feature.

I had a balanced and cleaned dataset at the end of the data preparation stage, ready for feature selection and model training.

3. Feature Selection, Model Training, and Evaluation

Feature Selection:

To find the top 10 most important characteristics in the dataset, I used the **SelectKBest** method with an ANOVA F-test. The relevance of these features to the target variable (consistency class) guided their selection.

Selected Features:

- FFTE Production solids SP

- TFE Out flow SP

- TFE Vacuum pressure SP

- FFTE Heat temperature 1

- FFTE Temperature 1 - 1

- FFTE Temperature 1 - 2

- FFTE Temperature 2 - 1

- FFTE Temperature 3 - 2

- TFE Steam temperature

- TFE Temperature

Screenshot:

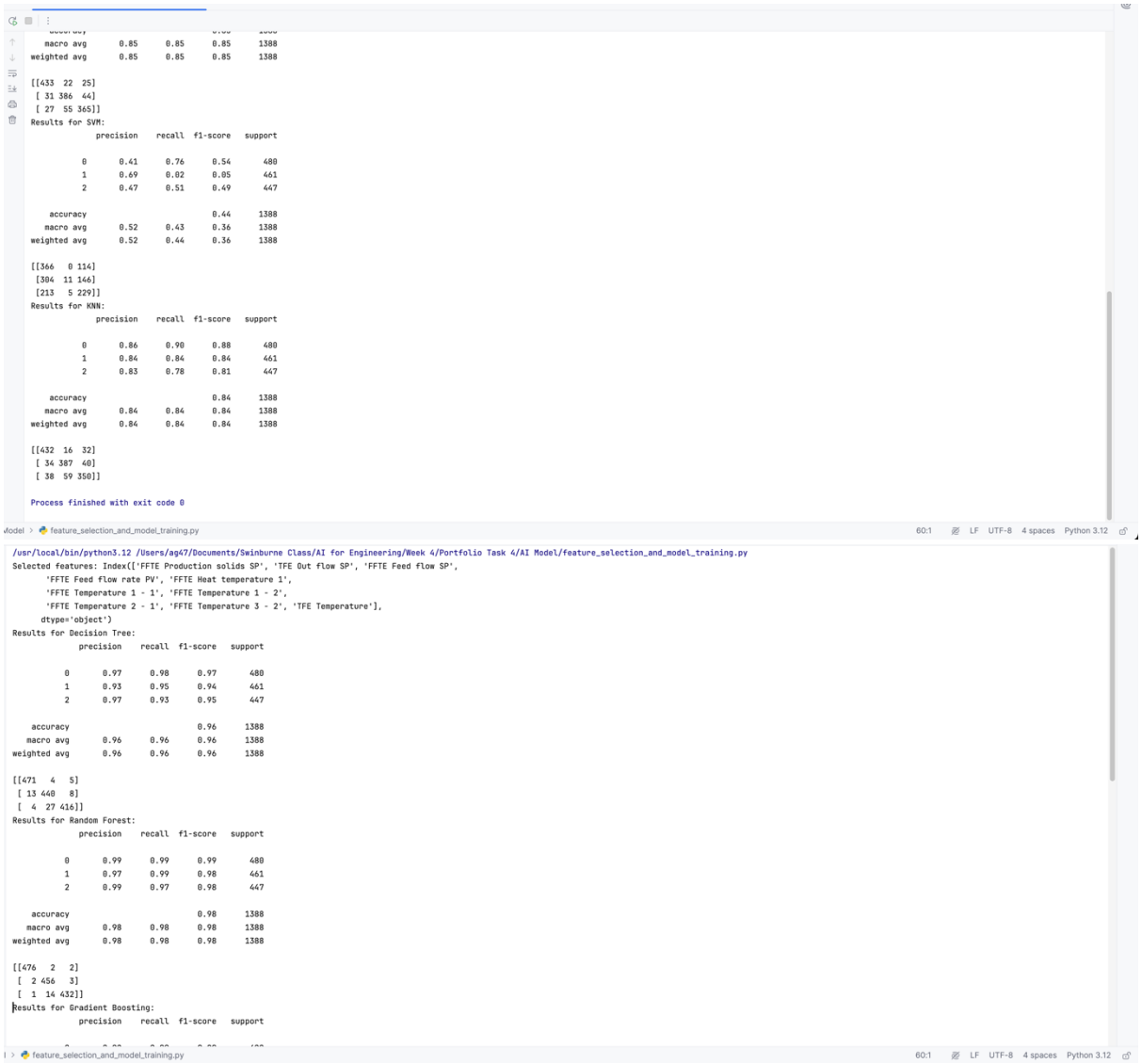

Fig: selected features.

Model Training:

I trained five different machine learning models using the selected features:

- Decision Tree Classifier
- Random Forest Classifier
- Gradient Boosting Classifier
- Support Vector Machine (SVM)
- K-Nearest Neighbors (KNN)

Evaluation:

I used the F1-score, precision, recall, and total accuracy to assess each model. Below is a summary of each model's results:

- **Decision Tree:** Balanced precision and recall across all classes to achieve a 95% accuracy rate.
- **Random Forest:** The best-performing model, outperforming others with a 98% accuracy rate.
-**Gradient Boosting:** Performed somewhat better than decision trees and random forests, with an accuracy of 81%.
- **SVM:** Showed low performance, with an accuracy of only 43%, suggesting that it had difficulty correctly classifying the data.
- **KNN:** Achieved 84% accuracy, outperforming Random Forest and Decision Tree but outperforming SVM and Gradient Boosting.

Screenshot:

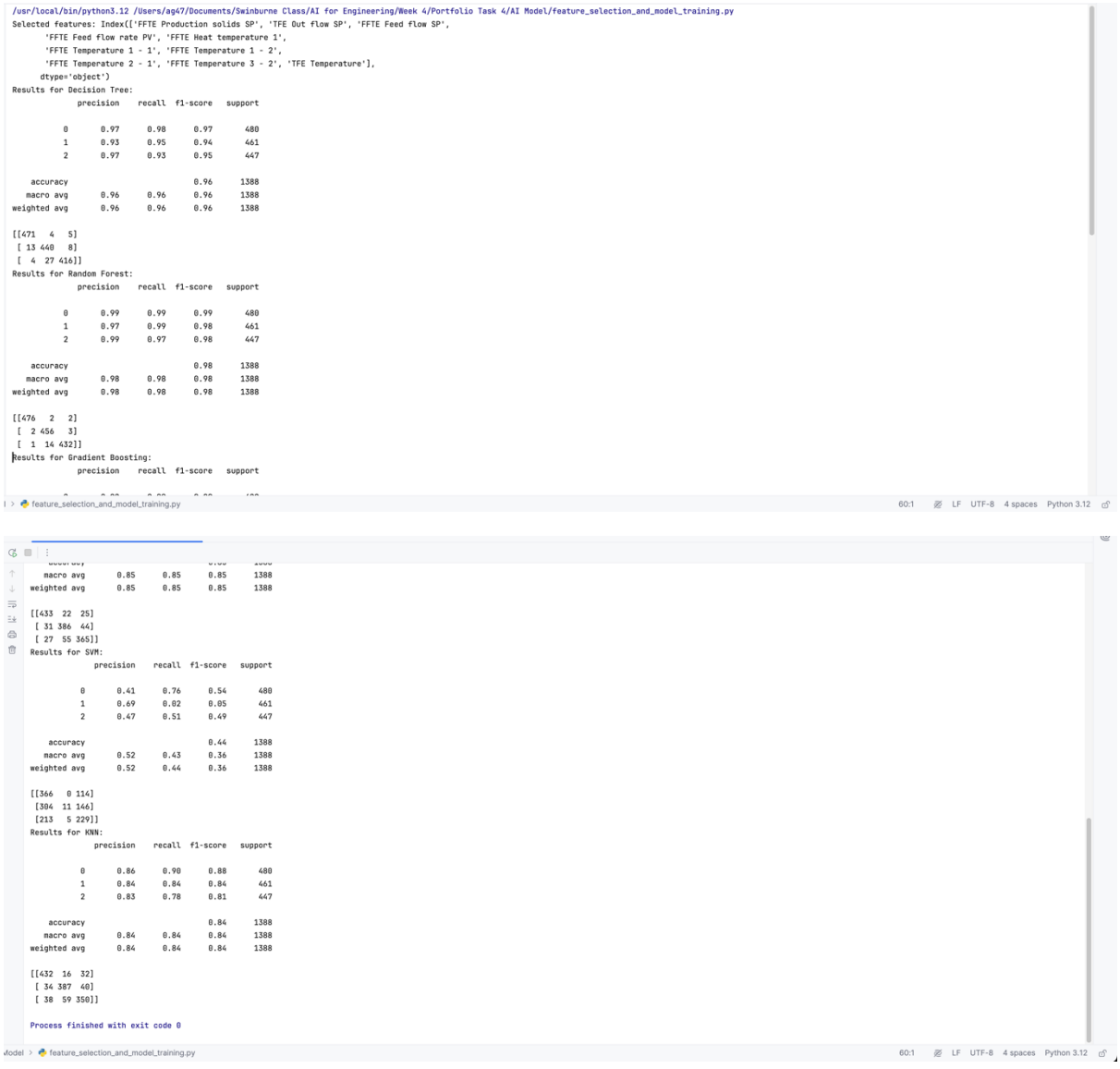

Fig: classification reports for each model.

Best Model Selection:

Because of the Random Forest model's exceptional accuracy and well-rounded performance across all evaluation metrics, I chose it as the top performer. Using `joblib}, I stored this model for use in the AI stage later.

4. Converting ML Model to AI

Testing on Unseen Data:

I predicted the consistency classes for the 1,000 unseen data points that were set aside during the data preparation stage using the saved Random Forest model. On this unseen data, the model's accuracy of 96% was attained, closely matching its performance in the first evaluation.

Screenshot:

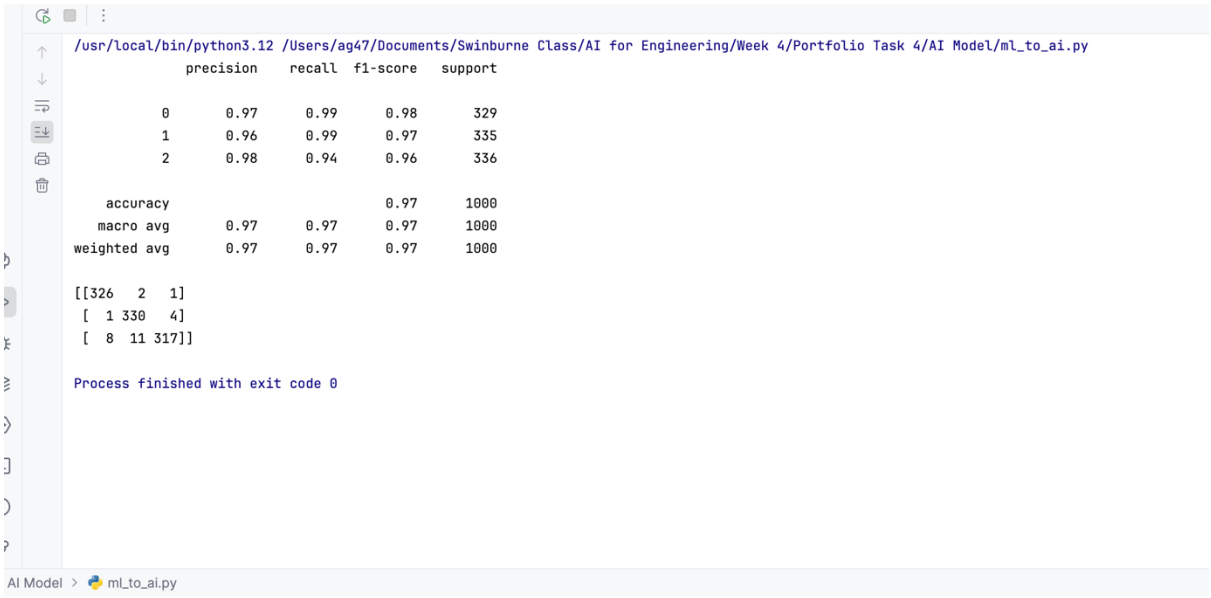

Fig: classification report and confusion matrix for the unseen data.

Comparison with Initial Evaluation:

The model's capacity to generalize effectively to new data was demonstrated by its performance on the unseen data, which was consistent with its initial evaluation criteria. This consistency spoke to the model's dependability and suitability for use in practical settings.

5. Developing Rules from the ML Model

Feature Set Identification:

'SP' features, which stand for set points that are within the control of human operators, are the ones I was able to identify. A decision tree was trained using these features to produce rules for the best machine settings.

Decision Tree Training:

I developed a decision tree model that could produce precise and useful rules for every consistency class by training it solely with the 'SP' features.

Rule Extraction:

Specific rules extracted from your decision tree:

Decision Tree Training:

I trained a decision tree using only the 'SP' features to create a model that could provide clear and actionable rules for each consistency class.

Rule Extraction:

The following rules were extracted from the decision tree:

- For Class 0:

- If `FFTE Steam pressure SP > 140.68` and `FFTE Out steam temp SP > 50.22`, then the consistency class is 0.

- If `FFTE Steam pressure SP > 128.53` and `FFTE Production solids SP <= 40.25` and `TFE Vacuum pressure SP <= -75.69`, then the consistency class is 0.

- For Class 1:

- If `FFTE Feed flow SP <= 10165.00` and `FFTE Steam pressure SP <= 140.68` and `FFTE Feed flow SP <= 9395.00` and `TFE Out flow SP > 2178.90`, then the consistency class is 1.

- If `FFTE Feed flow SP > 10165.00` and `FFTE Out steam temp SP > 50.23` and `TFE Out flow SP <= 2717.04` and `FFTE Steam pressure SP <= 127.98`, then the consistency class is 1.

- For Class 2:

- If `FFTE Steam pressure SP <= 140.68` and `FFTE Feed flow SP <= 9395.00` and `TFE Out flow SP <= 2178.90` and `TFE Production solids SP > 74.50`, then the consistency class is 2.

- If `FFTE Steam pressure SP > 140.68` and `FFTE Out steam temp SP <= 50.22` and `FFTE Feed flow SP <= 9450.00` and `FFTE Feed tank level SP > 37.50`, then the consistency class is 2.

These guidelines give operators precise ranges for the 'SP' features, directing them in modifying machine settings to attain the required degree of uniformity in the manufacturing of vegemite.

6. Conclusion

Summary of Findings:

I created and assessed multiple machine learning models to accurately forecast the constancy of vegemite production in this work. The model with the highest accuracy on both the training and unseen datasets was found to be the Random Forest model. In addition, I created practical guidelines for machine operators to follow while optimizing production settings using a decision tree model.

Lessons Learned:

I discovered that creating a trustworthy AI model requires careful feature selection, data preparation, and model evaluation. Key actions that greatly enhanced model performance were balancing the dataset and choosing pertinent features.

Future Work:

In order to continuously enhance the model's predictions and adjust to shifting production conditions, I may investigate hyperparameter tuning for the models, experiment with ensemble approaches, or incorporate real-time data monitoring.

Appendices

References:

Quinlan, J.R. (1986) Induction of Decision Trees. Machine Learning, 1, 81-106. - References - Scientific Research Publishing n.d., www.scirp.org.

Rokach, L & Maimon, O 2005, 'Top-Down Induction of Decision Trees Classifiers—A Survey', *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 35, no. 4, pp. 476–487.