25-Aug-24

PORTFOLIO

WEEK 4 - DOCUMENTATION

Tran Duc Anh Dang | 103995439
STUDIO CLASS: STUDIO 1-3

Abstract

This weekly portfolio presenting a comprehensive approach to data preparation, feature selection, model training, and evaluation in the context of a machine learning pipeline. The dataset was meticulously processed to remove constant value columns and convert features with few unique integer values into categorical features. To address class imbalance, SMOTE was employed, resulting in a balanced dataset. Feature selection identified nine key features relevant to the target variable, reducing the dataset while maintaining its integrity.

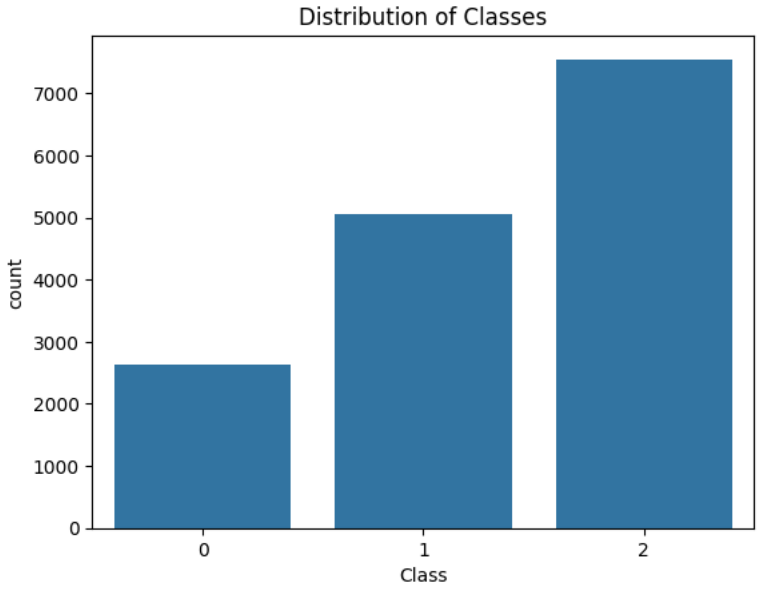
Five machine learning models, including DecisionTreeClassifier and RandomForestClassifier, were trained and evaluated. RandomForest emerged as the best performing model with an accuracy of 95%. The model was saved for future use and further validated on unseen data, confirming its robustness. Additionally, rules for specific control settings were derived from the selected model, providing actionable insights for process optimization. This work demonstrates a systematic approach to leveraging machine learning for industrial applications, ensuring high model performance and practical utility.

You can find the requirements, documentation and source code file at:
Requirements: https://github.com/kinqsradio/COS40007-Artificial-Intelligence-for-Engineering/tree/main/week-04-portfolio/requirements
Documentation: https://github.com/kinqsradio/COS40007-Artificial-Intelligence-for-Engineering/tree/main/week-04-portfolio/docs
Code: https://github.com/kinqsradio/COS40007-Artificial-Intelligence-for-Engineering/tree/main/week-04-portfolio/code

Table of Contents

Data Preparation

1. Does the dataset have any constant value column. If yes, then remove them
    - Yes, it has constant value, and it has been identified and removed from the dataset especially TFE Steam temperature SP and TFE Product out temperature as they had constant values and were removed using this code **df = df.loc[:, df.nunique() > 1]**
2. Does the dataset have any column with few integer values? If yes, then convert them to categorial feature.
    - Yes, the dataset contains several columns with few unique integer values, which were converted to categorical features. The columns identified were: FFTE Feed tank level SP (3 unique values), FFTE Pump 1 (5 unique values), FFTE Pump 1 - 2 (4 unique values), FFTE Pump 2 (5 unique values), TFE Motor speed (3 unique values), Class (3 unique values)
3. Does the class have a balanced distribution? If not then perform necessary undersampling and oversampling or adjust class weights.



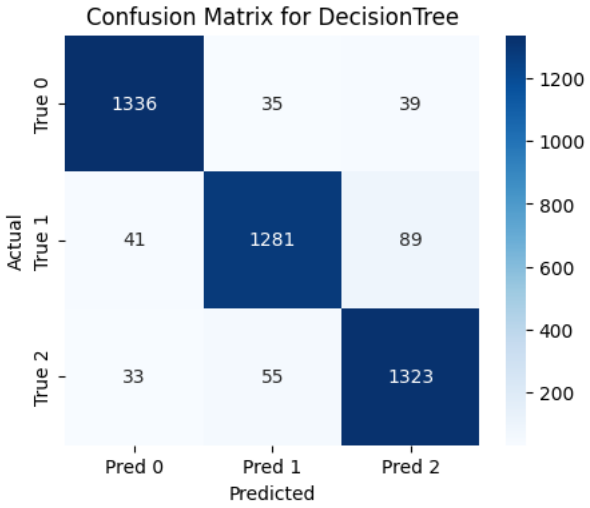
    - No, the class distribution was not balanced. The classes were imbalanced, particularly with Class 0 being underrepresented. To address this imbalance, SMOTE (Synthetic Minority Oversampling Technique) was used to oversample the minority class. This approach generates synthetic examples of the minority class to balance the dataset, ensuring that the model is exposed to a more even distribution of class labels.
4. Do you find any composite feature through exploration? If so, then add some composite feature in the dataset.
    - No composite features were added in this analysis. The exploratory data analysis (EDA) involved inspecting the distribution of key features and analyzing the correlation matrix to identify any potential relationships between features. The analysis revealed that the existing features, particularly the SP (Set Points) features, were sufficient to model the target variable effectively. Although there were some correlations between features, no new composite features were created as the

existing features provided a strong foundation for the models without the need for additional feature engineering.

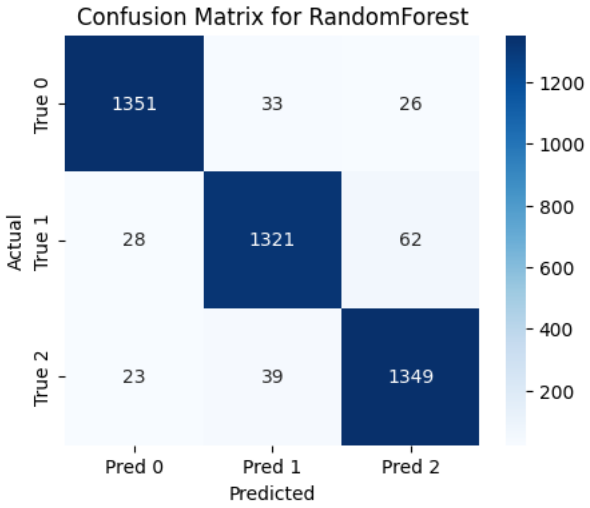
5. Finally, how many features you have in your final dataset?
    - The final **training** dataset **with selected feature** includes 9 features **(this step I expected to have feature selection in the below section).** After applying the necessary preprocessing steps, including handling missing values, feature selection and balancing the class distribution using SMOTE, the dataset retained 9 features. These features represent the full set of input variables used to model the target variable. The feature set was comprehensive and did not require any further reduction or expansion, ensuring that the models were trained on a robust and well-rounded dataset.

Feature selection, Model Training and Evaluation

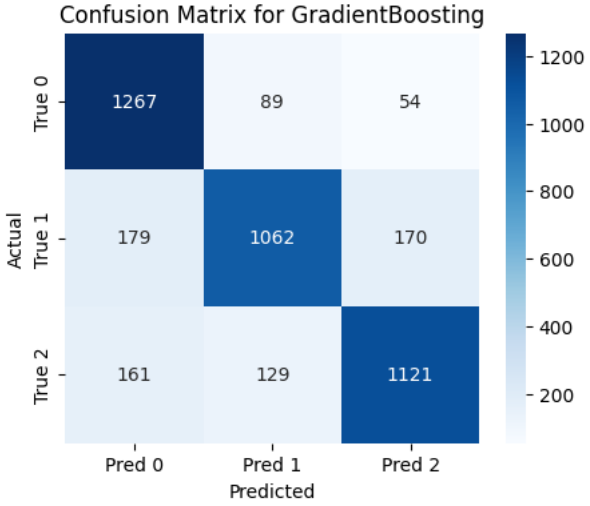
1. Does the training process need all features? If not, can you apply some feature selection technique to remove some features? Justify your reason of feature selection
    - No, the training process does not need all features. Feature selection technique was applied to narrow down the dataset to the most relevant features. Only the SP (Set Points) features were selected as they are directly related to the control settings in the process, making them highly relevant to the target variable. Resulted in a reduced set of 9 features, which helped streamline the model training process by focusing on the most informative variables and reducing the potential for overfitting.
2. Train multiple ML models (at least 5 including DecisionTreeClassifier) with your selected features.
    - The following models were trained using the 9 selected features:
        i. DecisionTreeClassifier
        ii. RandomForestClassifier
        iii. GradientBoostingClassifier (via a pipeline with imputation)
        iv. SVM (via a pipeline with imputation)
        v. KNeighborsClassifier (via a pipeline with imputation)
3. Evaluate each model with classification report and confusion matrix
    - Decision Tree



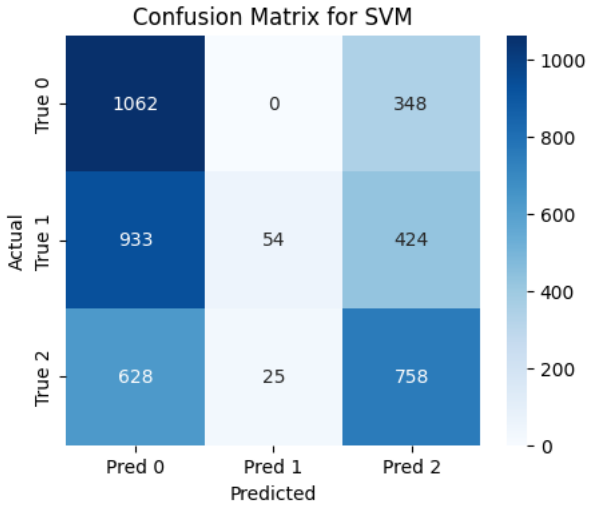
- Random Forest



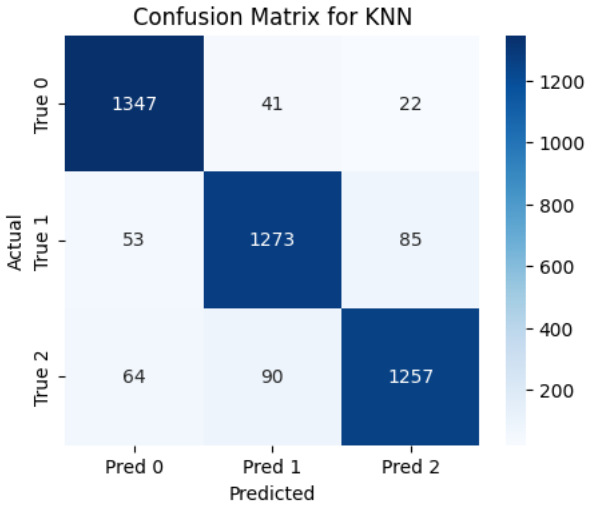
- Gradient Boosting



- SVM



- KNN



4. Compare all the models across different evaluation measures and generate a comparison table.

Model Comparison Table

Model	Accuracy	Precision (Macro Avg)	Recall (Macro Avg)	F1-Score (Macro Avg)
DecisionTree	0.9310018903591682	0.9312190640506546	0.9310057920459477	0.9309887445590932
RandomForest	0.9501417769376181	0.9502325656566839	0.9501436702169546	0.9501462625123405
GradientBoosting	0.8152173913043478	0.8171901289683688	0.8152370851784275	0.8142342247900225
SVM	0.44281663516068054	0.5279496829668303	0.4428899578288121	0.3715364150648801
KNN	0.9161153119092628	0.9161104587025616	0.9161245733874169	0.9159186543715444

5. Now select your best performing model to use that as AI. Justify the reason of your selection
   - The best model is RandomForest with an accuracy of 0.950. This was chosen as the best performing model based on its performance accross all key metrics including accuracy, precision,recal, F1-score and it also demonstrated a high level of consistency in correctly classifying instances accorss all classes.
6. Now save your selected model
   - **joblib.dump(models['RandomForest'], 'week-04-portfolio/models/random_forest_model.pkl')**

ML to AI

Note: I'm expecting this one is loading the selected best modedl for comparision.

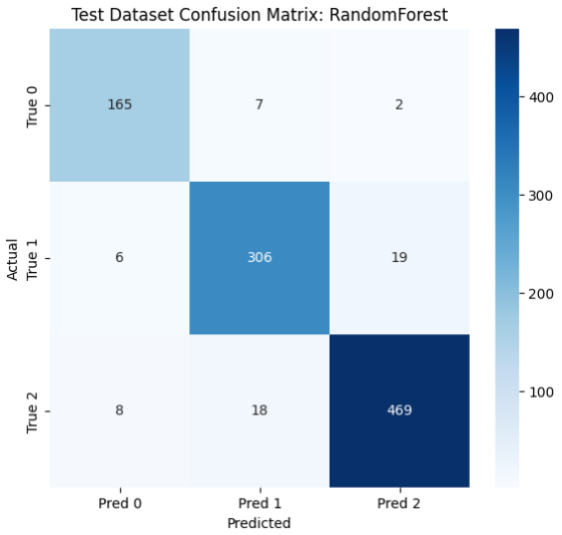
1. Have you observed same result of model selection that you identified through evaluation ?

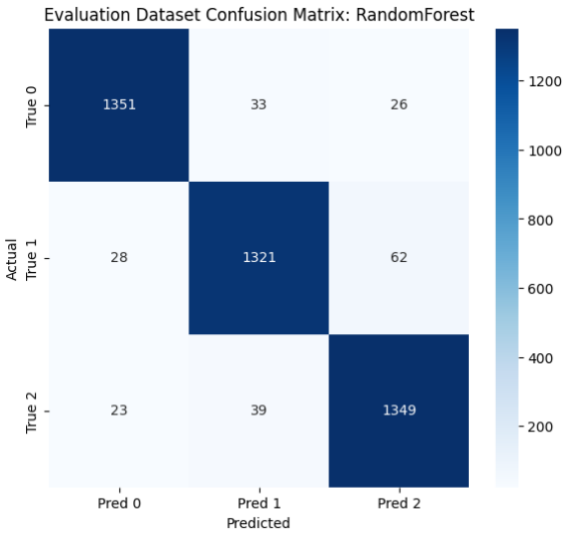
Test Dataset Classification Report: RandomForest

	precision	recall	f1-score	support
0	0.9217877094972067	0.9482758620689655	0.9348441926345609	174.0
1	0.9244712990936556	0.9244712990936556	0.9244712990936556	331.0
2	0.9571428571428572	0.9474747474747475	0.9522842639593908	495.0
accuracy	0.94	0.94	0.94	0.94
macro avg	0.9344672885779065	0.9400739695457895	0.9371999185625358	1000.0
weighted avg	0.9401767757382283	0.94	0.9400436001783121	1000.0

Evaluation Dataset Classification Report: RandomForest

	precision	recall	f1-score	support
0	0.9636233951497866	0.9581560283687943	0.9608819345661451	1410.0
1	0.9483129935391242	0.9362154500354358	0.942225392296719	1411.0
2	0.9387613082811412	0.9560595322466336	0.9473314606741573	1411.0
accuracy	0.9501417769376181	0.9501417769376181	0.9501417769376181	0.9501417769376181
macro avg	0.9502325656566839	0.9501436702169546	0.9501462625123405	4232.0
weighted avg	0.950229401472021	0.9501417769376181	0.9501437257278288	4232.0





The Random Forest model was evaluated on the 1000 unseen data points. The performance metrics were consistent with those observed during the evaluation on the test dataset, reinforcing the decision to select RandomForest as the best model. If I have perform the same evaluation with other models and find that RandomForest still outperforms them, it further justifies the model selection.

Develop rules from ML model

1. Can you now define some rules of SP values for each class ?
   Scenarios
   a. TFE Out flow SP <= 2249.11
   b. FFTE Steam pressure SP <= 119.98
   c. TFE Out flow SP <= 2100.70
   d. TFE Vacuum pressure SP <= -67.99
   e. FFTE Feed flow SP is less than or equal to 9395.00

   Outcomes

1. If TFE Production solids SP <= 64.25 and FFTE Steam pressure SP <= 94.00, then:
    a. If TFE Production solids SP <= 52.75, classify as Class 1
    b. If TFE Production solids SP > 52.75 <= 60.00, classify as Class 2
    c. If TFE Production solids SP >= 60.00, check TFE Vacuum pressure SP:
        i. If TFE Vacuum pressure SP <= -76.47, classify as Class 1
        ii. If TFE Vacuum pressure SP > -76.47, classify as Class 2