Capstone Project Data Science Bootcamp

AnomaData (Automated Anomaly Detection for Predictive Maintenance)

Problem Statement

Many different industries need predictive maintenance solutions to reduce risks and gain actionable insights through processing data from their equipment.

Although system failure is a very general issue that can occur in any machine, predicting the failure and taking steps to prevent such failure is most important for any machine or software application.

Predictive maintenance evaluates the condition of equipment by performing online monitoring. The goal is to perform maintenance before the equipment degrades or breaks down.

This Capstone project is aimed at predicting the machine breakdown by identifying the anomalies in the data.

The data we have contains about 18000+ rows collected over few days. The column ‘y’ contains the binary labels, with 1 denoting there is an anomaly. The rest of the columns are predictors.

Steps:

- Loading the dataset
- Exploratory Data Analysis
- Data Preprocessing
- Feature Scaling
- Feature Engineering
- Model Selection
- Model Training
- Hyperparameter tuning
- Model Evaluation

Exploratory Data Analysis:

- After loading the dataset, we observed that there were 63 columns in total.
- We found that: there were many features that were correlated and dataset target column was heavily imbalanced. Also part of the data was either left-skewed or right-skewed.
- There were no null values in the dataset

Data Preprocessing

- First, we dropped the ‘y.1’ and ‘time’ column which were unnecessary for our purpose.
- Then we separated the features X and the target y.
- In order to address the skewness in the data, we applied yeo-johnson transformation as there was both positive as well as negatively skewed data.
- Then, in order to address the imbalance in the dataset, we applied oversampling method SMOTE.

Feature Scaling

- We scaled the features using Standard Scaler method, both on the train and test data.
- We exported the processed train and test data into csv files
- In order to reduce the dimensions of the data, we decide to apply PCA.

Feature Engineering 1

- Using PCA, we found we are able to reach 90% of explained variance ratio using n_components=30. So we fitted the data using PCA to reduce the feature dimensionality.

Model Selection

- We decided to select Support Vector Classifier, Random Forest, Gradient Boosting and XGBoost to build our model and check the performance of each. We found the following scores.

	Model Name	Accuracy	Recall	F1 Score	Cross Val Score
0	Support Vector Classifier	0.982609	0.36	0.219512	0.990184
1	Random Forest	0.994837	0.32	0.457143	0.998940
2	Gradient Boosting Classifier	0.943750	0.52	0.111588	0.965045
3	XGBoost	0.992935	0.32	0.380952	0.997093

Model Selection Contd..

- As we can see, even if the accuracy is very high for the models, **the recall is extremely low**. So we decided to do feature selection using SelectFromModel.

	Model Name	Accuracy	Recall	F1 Score	Cross Val Score
0	Support Vector Classifier	0.982609	0.36	0.219512	0.990184
1	Random Forest	0.994837	0.32	0.457143	0.998940
2	Gradient Boosting Classifier	0.943750	0.52	0.111588	0.965045
3	XGBoost	0.992935	0.32	0.380952	0.997093

Feature Engineering 2 : Feature Selection

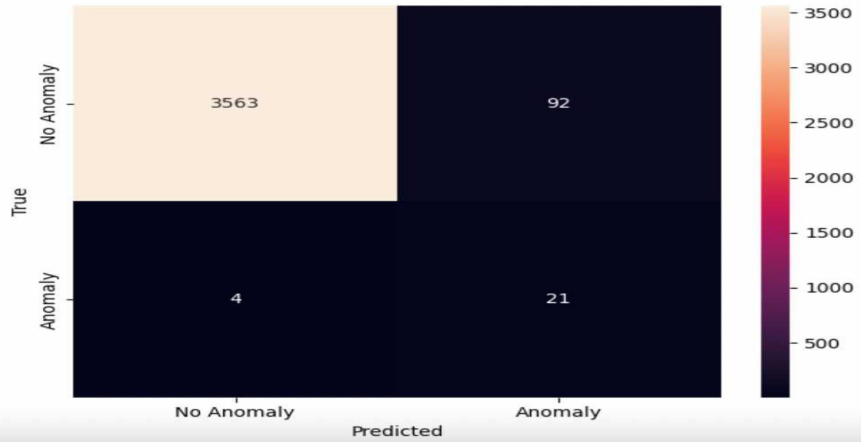
- We applied feature selection using SelectFromModel on 3 models: Gradient Boosting, Random Forest and XGBOOST.

Feature Selection contd..

```
--------------------------------------------------
Binary Classification - Predictictive Maintenance
==================================================
Accuracy: 0.9739130434782609
Classification Report:
              precision    recall  f1-score   support

  No Anomaly       1.00      0.97      0.99      3655
     Anomaly       0.19      0.84      0.30        25

    accuracy                           0.97      3680
   macro avg       0.59      0.91      0.65      3680
weighted avg       0.99      0.97      0.98      3680
```



Using Gradient Boosting:

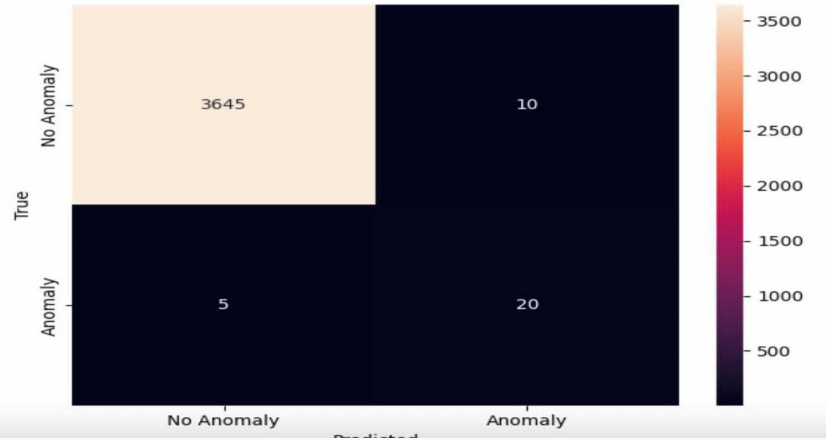
We can see that the recall score as well as accuracy is very good.

Feature Selection contd..

```
==================================================
Binary Classification - Predictictive Maintenance
==================================================
Accuracy: 0.9959239130434783
Classification Report:
              precision    recall  f1-score   support

  No Anomaly       1.00      1.00      1.00      3655
     Anomaly       0.67      0.80      0.73        25

    accuracy                           1.00      3680
   macro avg       0.83      0.90      0.86      3680
weighted avg       1.00      1.00      1.00      3680
```



Using Random Forest:

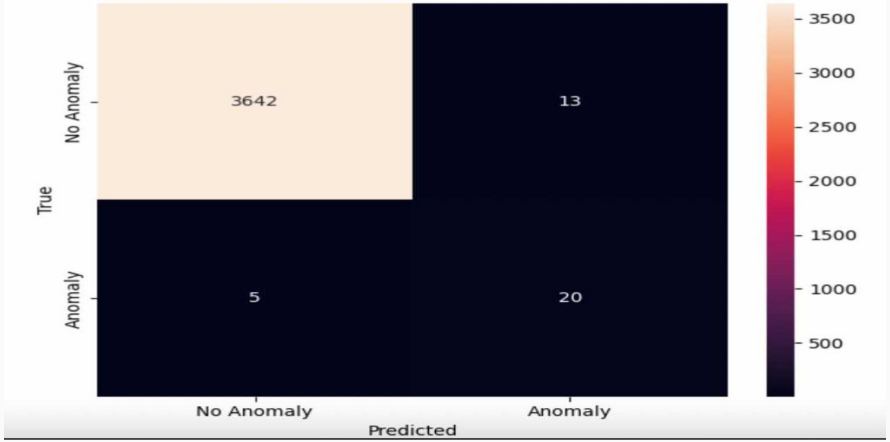
We can see that the recall score is lower than Gradient Boosting but accuracy and f1 score is better.

Feature Selection contd..

```
==================================================
Binary Classification - Predictictive Maintenance
==================================================
Accuracy: 0.9951086956521739
Classification Report:
              precision    recall  f1-score   support

  No Anomaly       1.00      1.00      1.00      3655
     Anomaly       0.61      0.80      0.69        25

    accuracy                           1.00      3680
   macro avg       0.80      0.90      0.84      3680
weighted avg       1.00      1.00      1.00      3680
```



Using XGBOOST:

We can see that f1 score is a little lower compared to Random Forest

.

Hyperparameter Tuning

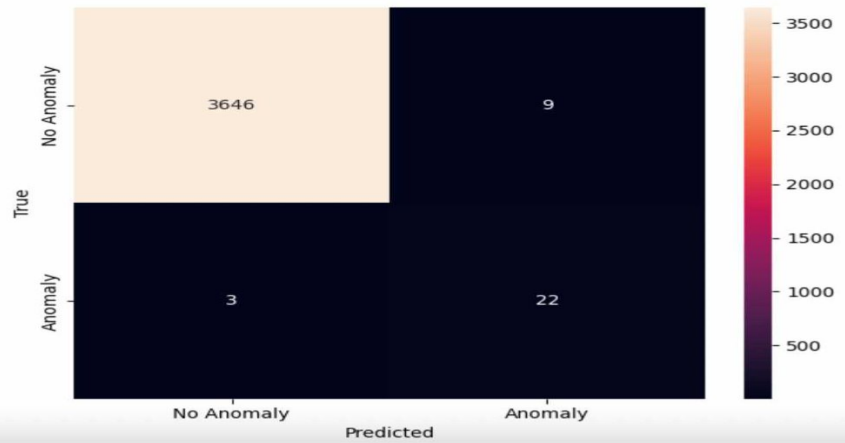
- We apply hyperparameter tuning on the Random Forest model that we trained in the previous step. We choose it because it shows overall best performance among the three.

Tuned Model Performance

```
==================================================
Binary Classification - Predictictive Maintenance
==================================================
Accuracy: 0.9967391304347826
Classification Report:
              precision    recall  f1-score   support

  No Anomaly       1.00      1.00      1.00      3655
     Anomaly       0.71      0.88      0.79        25

    accuracy                           1.00      3680
   macro avg       0.85      0.94      0.89      3680
weighted avg       1.00      1.00      1.00      3680
```



Random Forest model after hyperparameter tuning:

We can see that f1 score and recall of the tuned model has increased significantly compared to untuned Random Forest model.

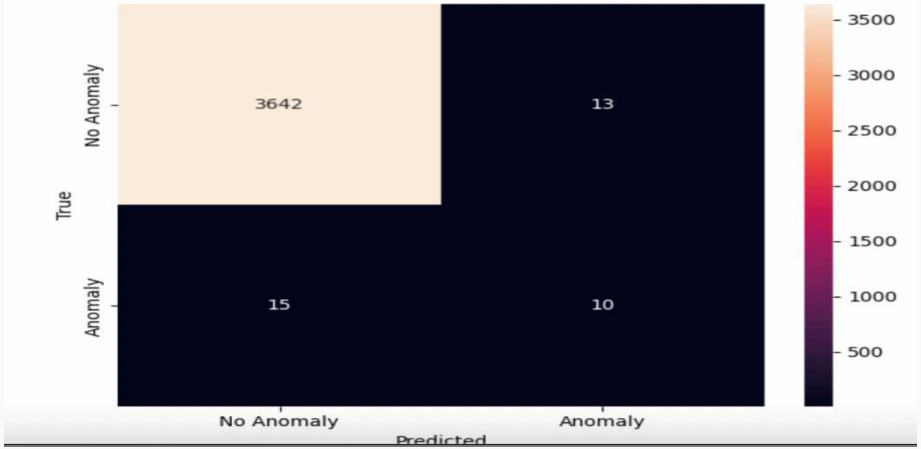
So we choose this as our final model and save to pickle file.

Using Deep Learning model

```
================================================
Binary Classification - Predictictive Maintenance
================================================
Accuracy: 0.9923913043478261
Classification Report:
              precision    recall  f1-score   support

  No Anomaly       1.00      1.00      1.00      3655
     Anomaly       0.43      0.40      0.42        25

    accuracy                           0.99      3680
   macro avg       0.72      0.70      0.71      3680
weighted avg       0.99      0.99      0.99      3680
```



- We apply deep learning model and check its performance.
- We see that recall is extremely low so we reject this model.

Github Link for the Project

https://github.com/saikattal/Capstone_Project

Thank you!