Coursework Summary

Vignesh Krishnakumar

230435614

1. Introduction:

This summary of the coursework provides a more detailed outlook on the code work used in the jupyter file for the building a predictive pipeline model on determining linear B-cell epitopes from protein sequences. Exploratory data analysis was carried out along with extensive data pre-processing to make sure data is refined to the finest. Feature reduction was a crucial step carried out getting the right features and different classifier machine learning models were tried upon to get the best classifier. Hyper Parametric tuning was implemented to tune the model increasing their discriminative ability, and the performance metric used was ROC curve with area under the curve as the main metric for selecting best classifier model.

2. Exploratory Data Analysis:

In EDA first we have imported the required libraries such as Pandas, NumPy, Matplotlib etc. First step was to check whether data had any missing values. The result - there were no missing values present. We proceeded further in checking for NaN values which also turned out to be zero. Since there was no missing or NaN values present, we didn't require to Impute the data to fill the missing cells in the dataset.

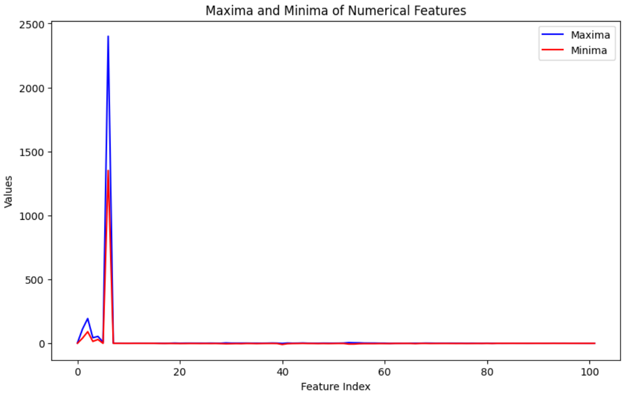
2.1 Visualisation:

We have used two different types of graphs to visualise data to understand the range of values with which we would be working with.

One was line graph which is depicted in Figure: 2, wherein you get to know ascent and descent in features maximum and minimum value. This provides the insight on the undulations present in data which requires standardisation in pre-processing stage.

Other way was bar graph (Figure: 3) to depict the same data giving a more defined outlook on features which had higher range difference to the one's with lower range difference in values.

By these visualisations we understand that the data must be standardised to get them in a similar range to give a more refined outlook to the data.



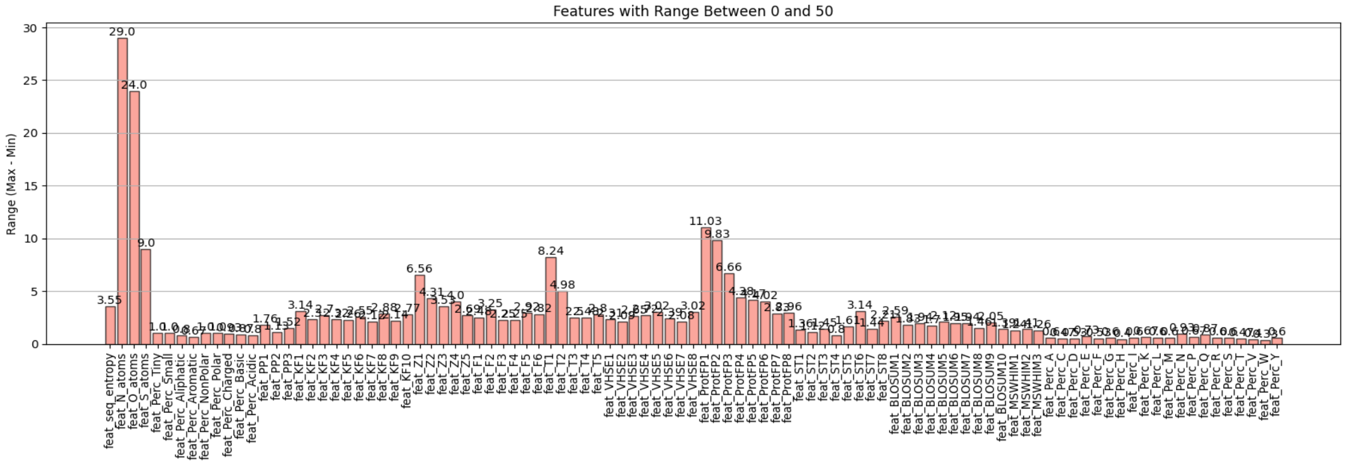


Figure 2 & 3: Line Graph and Bar Graph of 102 features from dataframe respectively.

2.2 Class Distribution:

Class distribution is a type of visualisation process to study the target class upon which the data is being trained to identify on unseen data.

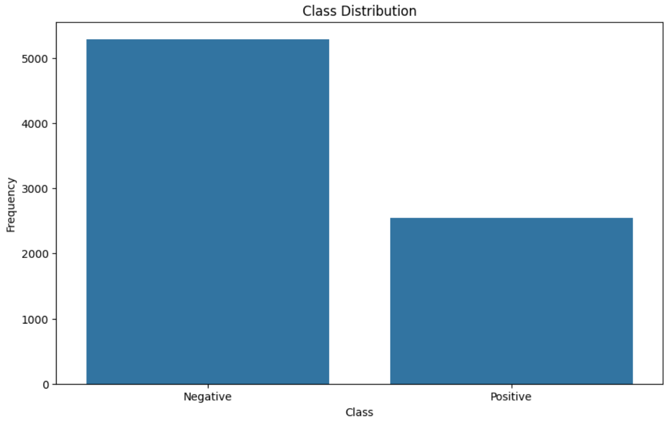


Figure 4: Class Distribution

We could see that the class is heavily inclined towards negative instances. This increases bias of the model towards negative instances rather than giving a non-biased outcome.

To remove this bias, we have done class re-balancing in data pre-processing to make sure model predicts non-biased outcomes.

2.3 Outlier Detection:

Outliers are datapoints present in dataset which are significantly away from the rest of the observations in dataframe [6].

First create a copy of dataframe to identify the number of outliers present in the dataframe since the treatment of them comes under section 2. We have implemented Isolation Forest [5] method to handle outliers.

Isolation Forest is the best model to use since the dimensionality of the data is high, along with that it is an unsupervised learning model meaning it can identify outliers without any labelled data present in them. Adding on, the assumptions made are minimal compared to statistical methods wherein statistical methods stick to particular data distribution [5]. These reasons make this model superior compared to other models.

Figure 5 is an example of one column “feat_Z2” wherein yellows are the outliers for this feature and the purples are grouped together showing them as non-outlier's.

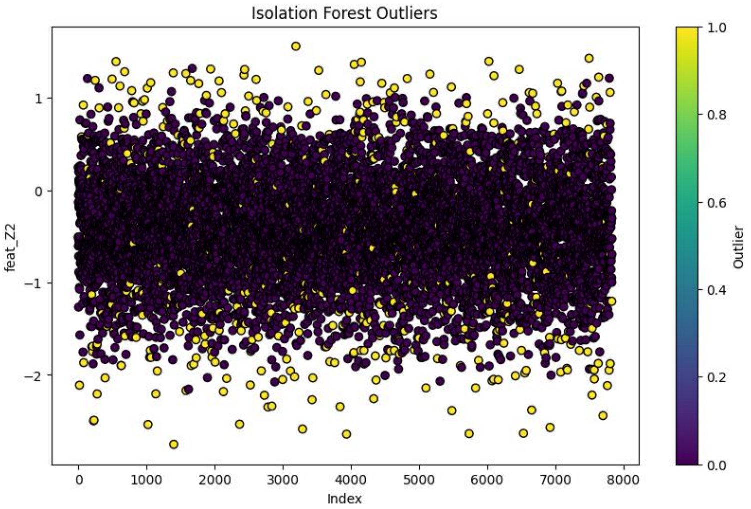


Figure 5: Oultier detection on feature feat_Z2

3. Data Pre-Processing:

As a first step duplicates are removed from the dataframe to increase the integrity, reduces the bias and provides a balanced representation of the dataframe.

Best practices to follow is to split the data at the start of pre-processing stage [7] preventing test data to be seen during rest of the pre-processing stage, allowing for unbiased evaluation which ultimately makes models generalise better to new unseen data. This helps ensure reliability and validity of the data models.

On trained data check for null values. Since no null values are present imputation of the dataframe is not required.

Now outliers will be treated using Isolation Forest model on the 80% train data.

After outlier treatment the dataframe is scaled using StandardScaler() [8] function. This is done to scale the data using mean and standard deviation to lower the variance of data which helps in improving the efficiency and effectiveness of the model.

Label encoder [9] is used to transform the class column with variables positive and negative to zero’s & one’s for model compatibility. For example, Logistic regression model is used which requires in binary format and making the model more efficient. It was used at necessary places to convert the data accordingly.

4. Feature Selection:

Feature selection is used to isolate the best features in dataset so that dataframe becomes more simpler, model performs more effectively and more efficiently. This increases accuracy and removes the dimensionality curse by reducing the dimensions of data, making it more suitable for analysis.

Three different feature selection techniques were considered:

- Information Gain [10]
- Chi-square test [10]
- Fishers score [10]

Fishers score was the best technique due to the following reasons:

1. This has high discriminative power to distinguish between each class based on variance, making it highly effective on identifying epitopes.
2. It is well suited for high dimensional dataset such as the one used here proving its effectives.
3. It considers every feature individually which makes it more useful in a biological dataset.

5. Class Rebalancing:

As previously seen from section 2.3 class rebalancing was required. SMOTE (Synthetic Minority Oversampling Technique) is an oversampling method which was used to address the class imbalance.

SMOTE mainly works on increasing the synthetic samples of the minority class so that the bias is eliminated towards one single instance, either positive or negative in our case. SMOTE is preferred over ADASYN and down-sampling methods for the following reasons:

- Loss of information is being prevented since down-sampling techniques focuses on reducing the majority class to address the class imbalance.[1]
- SMOTE decreases the false negatives but at the cost of increase in false positives.[1]
- ADASYN mainly works on the data points on the boundary line leading to increase in noise which is avoided in SMOTE. Also, when ADASYN was implemented for this dataset, it could still be seen there was imbalance in class due to the noise present. This shows SMOTE is more robust and simpler to use.

Figure 6 below shows the rebalanced class after using SMOTE.

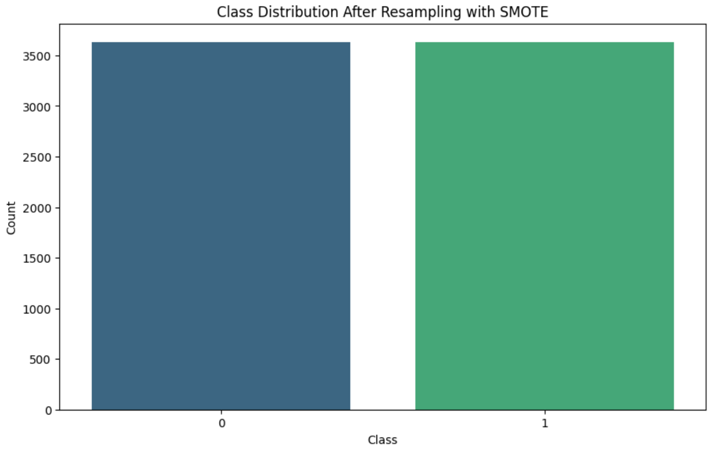


Figure 6: Class Distribution

6. Model Selection:

StratifiedKFold [3] is the technique used to split the data in more orderly manner than random sampling to increase the accuracy of the models.

Three classification models used namely:

- Logistic Regression (Base-line model)
- Random Forest Classifier
- eXtreme Gradient Boosting (XGboost) [2]

Random Forest proved to be the best performing model with mean AUC (Area Under the Curve) score of 0.804 on training data compared to XGboost and logistic regression. Same was applied on test data giving out an AUC score of 0.46 showing it required hyper-parametric tuning.

Hyper parametric tuning was conducted on Random Forest classifier by introducing four new parameters namely n_estimators, max_features, max_depth, max_leaf_nodes. All parameters carried their respective variables/values to find out the best set of values upon which the test data was fitted again.[12]

All four parameters had a base line set and tuning was done by taking considering each parameter separately leading to set of values which provided with an increased AUC score of 0.56 in the unseen test data.

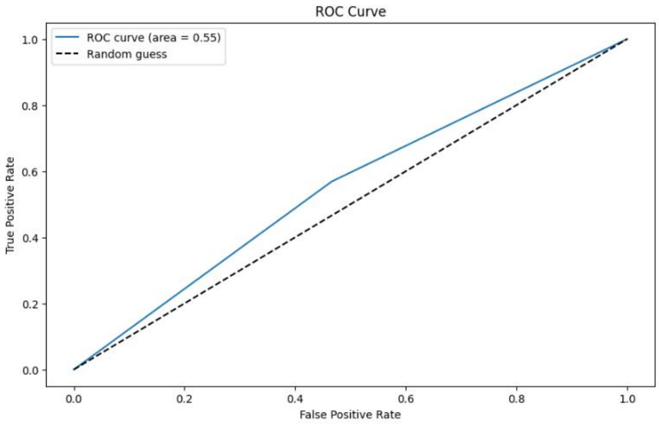


Figure 6: AUC score of Random Forest classifier

7. Pipeline:

In pipeline full dataset was loaded wherein all the pre-processing steps were applied.

The results showed that random forest classifier was still the best model providing its effectiveness on unseen data with an AUC [4] score of 0.46.

Performance evaluation along with hyper-parametric tuning was done on the test data which showed an increase in AUC score of 0.56 as shown in the below figure.

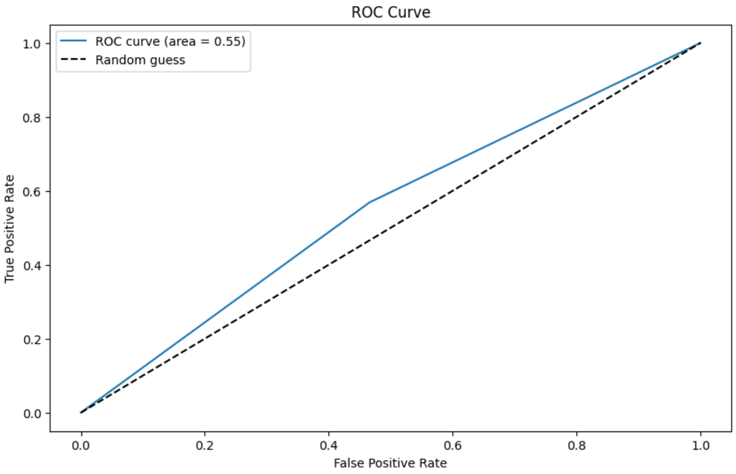


Figure 7: AUC score on the test data by Random Forest Classifier

8. Conclusion:

In this project we have achieved by developing a predictive pipeline to identify linear B-cell epitopes from protein sequence. Isolation Forest being most suitable outlier method and Fisher score being the best feature selection technique giving the best features from the dataset. The best model turned out to be Random Forest Classifier providing an ROC [4] score of 0.56 showing its predictability and reliability on unseen data.

9. References:

1. SMOTE | Towards Data Science
2. ML | XGBoost (eXtreme Gradient Boosting) - GeeksforGeeks
3. Stratified K Fold Cross Validation - GeeksforGeeks
4. AUC ROC Curve in Machine Learning - GeeksforGeeks
5. An Introduction to Isolation Forests (r-project.org)
6. 2201.00382 (arxiv.org)
7. Tutorial CS4850/AM41UD - Data Preprocessing (blackboard.com)
8. What is StandardScaler? - GeeksforGeeks
9. Scikit-Learn's preprocessing.LabelEncoder in Python (with Examples) | PythonProg
10. Feature Selection Techniques in Machine Learning (analyticsvidhya.com)
11. Guide to AUC ROC Curve in Machine Learning (analyticsvidhya.com)
12. Random Forest Hyperparameter Tuning in Python - GeeksforGeeks