UNIVERSITY OF HOUSTON

COSC 6342 MACHINE LEARNING

Dr. Ricardo Vilalta

PREDICTING BREAST CANCER FROM GENE EXPRESSION PROFILES (METABRIC) USING MACHINE LEARNING ENSEMBLE MODELS

Final Project

Manikanth Reddy Police
mpolice@cougarnet.uh.edu
2185952

Sai Vardhan Reddy Pogalla
spogalla@cougarnet.uh.edu
2203794

Amala Yeruva
ayeruva@cougarnet.uh.edu
2202754

ABSTRACT- Breast cancer is a widespread and potentially life-threatening disease, primarily impacting women, with significant rates of death. The substantial enhancement in survival rates, increasing from 50% to 80% when detected in the initial stages, highlights the vital importance of early identification. The objective of this work is to forecast breast cancer by analyzing gene expression profiles extracted from the METABRIC dataset. By conducting a thorough analysis, several machine learning models, such as Random Forest[7], K-Nearest Neighbors (KNN)[9], CatBoost[6] and Logistic Regression[6], were utilized to identify patterns and create prediction models. To tackle the issue of imbalanced classes and improve the performance of the model, the Synthetic Minority Over-sampling Technique (SMOTE) was utilized. Moreover, the study investigates the performance of hybrid models employing ensemble techniques, specifically with a Voting Classifier. The incorporation of many algorithms and ensemble techniques seeks to use the unique advantages of each model, enhancing the overall accuracy of predictions. This involves a comprehensive evaluation of model performance, integrating measurements such as precision, recall, F1-score, and the area under the ROC curve. The research findings provide valuable insights into the possible application of machine learning in predicting breast cancer using gene expression profiles. The versatility and adaptability of machine learning in tackling complex biomedical difficulties, as evidenced by a range of algorithms and ensemble approaches, open opportunities for the creation of more accurate and dependable predictive models in the domain of breast cancer detection.

1 LITERATURE SURVEY:

1.1 Unsupervised Methods:

Many studies have explored the METABRIC dataset with one common goal of grouping the patients using unsupervised learning methods. In a study focused on grouping breast cancer patients by subtype, the authors put significant emphasis on genome-based stratification, utilizing diverse genomic data such as copy number aberrations, gene expression, and mutation information to distinctly identify subgroups. With the help of this subgroup, we were able to provide target medication to the patients [1]. In a different study, to understand how breast cancer relapses over time, especially late recurrences, the authors categorized the patients using spatial-temporal patterns of breast cancer relapse [2]. This helped to monitor the patients with a high rate of further complications after breast cancer is cured. In a different study, to group the patients and know their probable outcomes of the treatment, the authors studied the relationship between genetic changes (copy number) and gene activity to group them. This study revealed new subgroups of breast cancer with different outcomes, including a high-risk group and a favorable prognosis group. This helped the hospitals know who had a better chance of successful treatment or longer survival compared to other subgroups [3].

1.2 Supervised methods:

There are also other sets of studies that were aimed at predicting the survival chance of a patient using supervised learning methods.

1.2.1 Breast Cancer treatment outcome with SVM.

In one such study the authors aimed to use gene expression data to predict survival outcomes in breast cancer patients treated using hormone (HT) and chemotherapy (CT) agents. Here, the authors primarily identified the genes related to the agents they were testing. Then a copy number of these genes in breast cancer cell lines were analyzed using multiple factor analysis of GI 50 values. Then SVM is applied to classify them into resistant or

sensitive categories. This SVM model then classifies patients into categories of treatment response using parameters such as misclassification cost and flexibility control. This study was aimed to provide which type of treatment would yield a better response in the patients [4].

1.2.2 Identification of different subtypes of breast cancer using Ensemble Methods and CM1 Score:

In a study to classify different breast cancer subtypes, the authors used CM1 score to identify discriminative probes for each group and utilized an ensemble learning technique with 24 different classifiers.

CM1 Score:

The CM1 score is a supervised method employed to assess the variation in gene expression levels between samples of different breast cancer intrinsic subtypes. Specifically designed for luminal A, luminal B, HER2-enriched, basal-like, and normal-like subtypes, the CM1 score evaluates each of the 48,803 probes and selects the ten most discriminative ones for a given subtype. This selection process is performed iteratively during the refinement process, aligning with each new label assigned to a sample by the classifiers.

After calculating the CM1 score of the samples. An ensemble learning technique with 24 classifiers is used, incorporating these discriminative probes as features, to assign labels to the samples. The iterative refinement process involves recomputing the CM1 score and selecting discriminative probes in each iteration, with the classifiers influencing subsequent iterations. The refined labels obtained through this process showed enhanced consistency and better alignment with clinicopathological markers and overall survival compared to the original labels provided by the PAM50 method. This iterative approach contributed to a more accurate characterization of breast cancer intrinsic subtypes in the METABRIC dataset [5].

1.3 Our Approach:

In this project, our goal was to predict patient survival likelihood by employing various classification algorithms, like CatBoost, Logistic Regression, KNN, and Random Forest Classifier. After preprocessing the dataset, we assessed model accuracy, performed hyperparameter tuning, and explored ensemble methods to combine these classifiers for improved predictive performance.

1.4 Dataset:

The dataset that is used in the project is called The Molecular Taxonomy of Breast Cancer International Consortium (METABRIC) this dataset was collected by collaboration between Professor Carlos Caldas from Cambridge Research Institute and Professor Sam Aparicio from the British Columbia Cancer Centre in Canada who together published this dataset on Nature Communications. This dataset comprises targeted sequencing data obtained from 1,980 primary breast cancer samples. The genetic component of the dataset includes z-scores for

mRNA levels of 331 genes and information on mutations in 175 genes. There are 693 different attributes in the dataset, and below are a few of the prominent features and their meaning.

Attribute Name	Attribute Type	Description
patient_id	Object	Unique Id of a Patient
age_at_diagnosis	Int	Age of the Patient at start of their diagnosis.
type_of_breast_surgery	Object	Breast surgery type 1-Mastectomy, 2-Breast Conserving
Cellularity	Object	Post-chemotherapy cancer cellularity: tumor cells and arrangement.
Chemotherapy	Int	Whether the patient had chemotherapy or not.
er_status	Object	Cancer cells exhibit estrogen receptor positivity or negativity.
hormone_therapy	Int	Whether the patient received hormonal treatment (yes/no).
mutation_count	Float	Number of gene that has relevant mutations
overall_survival	Object	Target variable: patient's status (alive or dead).

Above are a few of the columns from the dataset.

2 METHODOLOGY:

2.1 Exploratory Data Analysis:

Before preprocessing the dataset let us do some exploratory data analysis.

Dataset Statistics:

There are 1904 rows in the dataset with 693 columns.

Datatype	Column Number
float64	498
Object	190
Int64	5

Missing Data:

	Missing Count	Missing Percentage
tumor_stage	501	26.313025
3-gene_classifier_subtype	204	10.714286
primary_tumor_laterality	106	5.567227
neoplasm_histologic_grade	72	3.781513
cellularity	54	2.836134
mutation_count	45	2.363445
er_status_measured_by_ihc	30	1.575630
type_of_breast_surgery	22	1.155462
tumor_size	20	1.050420
cancer_type_detailed	15	0.787815
tumor_other_histologic_subtype	15	0.787815
oncotree_code	15	0.787815

Figure 1. Missing count and percentages of columns

When doing data analysis of a dataset. We always look for the datapoints that are either missing or have NaN by checking the number of missing data, we can decide on what model we need to use or whether we need to fill those missing values in the dataset. From our dataset analysis below is percentage of missing values. From figure 1 we can see that almost 25% of the data in tumor_stage is missing and almost 10% of the data is missing for the 3- gene_classifier_subtype.

Target Class Distribution:

As we are doing classification it makes sense to check the distribution of the target class. In our dataset our target class overall_survival is binary variable (1-Yes and 0-No). From the count plot in Figure 2 we can see the number of dead patients is more compared to the people who survived indicating a little class imbalance.

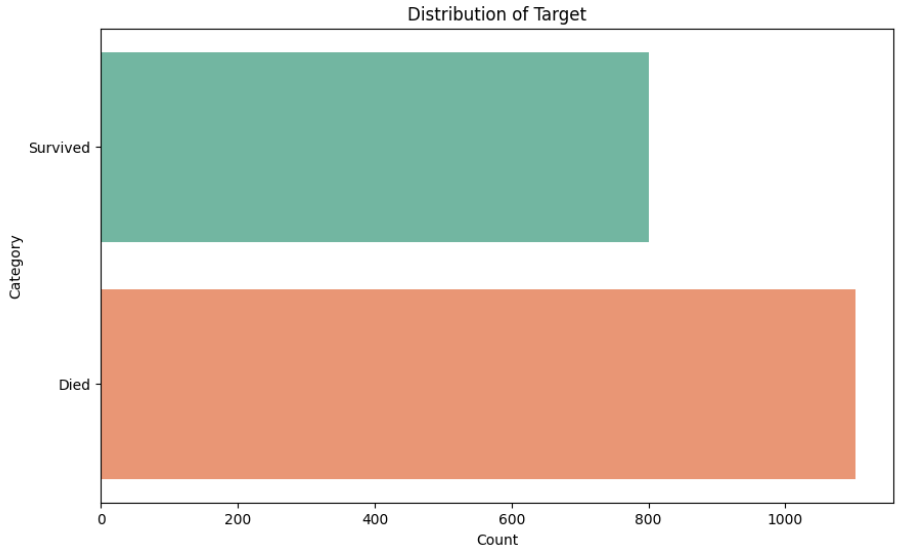


Figure 2. Count Plot of Target Variable

Outlier Analysis:

Outlier analysis is essential in data exploration as it helps identify anomalies, and data quality issues. Outliers distort summary statistics and impact the accuracy of models. Outliers can also provide insight into anomalies and patterns in the data. One popular method to visualize the outliers in data is box plots. From box plots we can consider the datapoint above the upper whisker and lower than lower whisker as outliers. From Figure 3, we can see the attributes lumph_nodes_examined_positive, mutation_count, tumor_size, brca1 have a lot of outliers whereas the attributes Nottingham_prognostic_index, age_at_diagnosis have minute to no outliers.

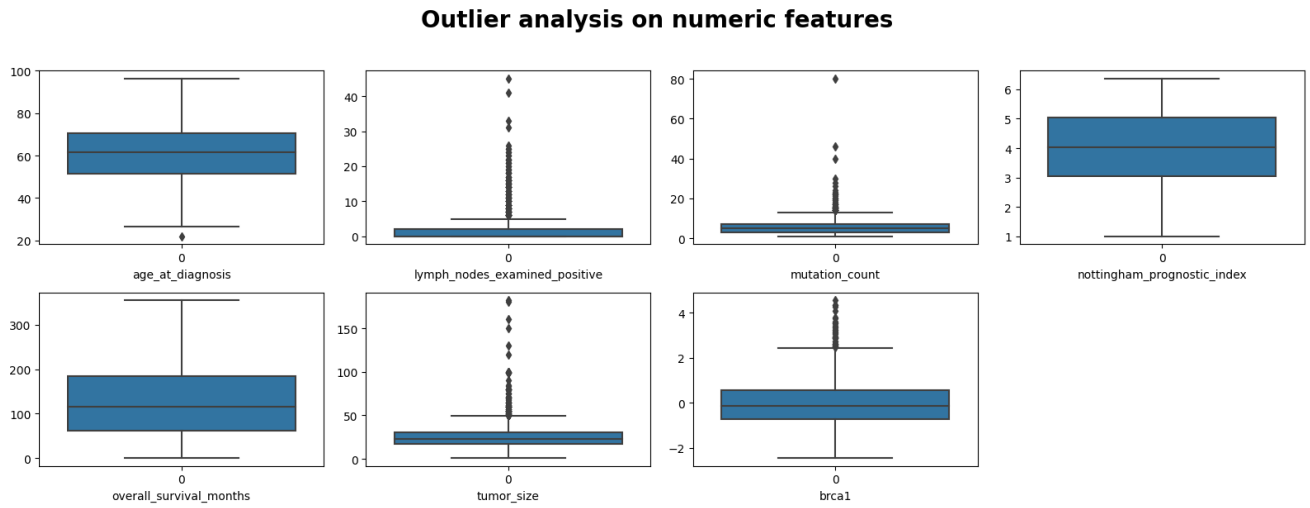


Figure 3. Box Plots of few Numerical Attributes

Correlation Analysis:

Correlation analysis helps uncover relationships and dependencies between variables. By quantifying the strength and direction of associations, correlation analysis provides insights into how variables may interact, aiding in feature selection for modeling. From the below figure 4 we can see the attributes overall_survival_months, inferred_menopausal_state, radio_therapy is positively correlated and the attributes age_at_diagnosis, tumor_size is negatively correlated with overall_survival.

```
Correlation Matrix:
                                Correlation with Overall Survival
overall_survival                                         1.000000
overall_survival_months                                  0.384467
inferred_menopausal_state                                0.170915
radio_therapy                                            0.112083
pam50_+_claudin-low_subtype                              0.063790
3-gene_classifier_subtype                                0.061270
primary_tumor_laterality                                 0.055441
chemotherapy                                             0.045625
tumor_other_histologic_subtype                           0.032734
cancer_type                                              0.026900
her2_status_measured_by_snp6                             0.024611
pr_status                                                0.022425
cellularity                                             -0.016356
er_status_measured_by_ihc                               -0.019156
er_status                                               -0.019587
hormone_therapy                                         -0.030401
her2_status                                             -0.033201
cancer_type_detailed                                    -0.034002
oncotree_code                                           -0.034246
brca1                                                   -0.074490
mutation_count                                          -0.076051
integrative_cluster                                     -0.082834
neoplasm_histologic_grade                               -0.091755
nottingham_prognostic_index                             -0.138000
cohort                                                  -0.149645
tumor_stage                                             -0.169503
lymph_nodes_examined_positive                           -0.173437
type_of_breast_surgery                                  -0.187856
tumor_size                                              -0.194419
age_at_diagnosis                                        -0.303710
```

Figure 4. Correlation of attributes against overall_survival

2.2 Data Pre-processing:

Removing Unnecessary Attributes:

We can remove unnecessary attributes from the dataset which are not related to the classification. Doing this reduces the dimensionality of the dataset from 693 to 28.

Filling Missed Values:

As we have seen in the data exploration, a few columns have an exceptionally large number of missing as these can affect the accuracy of the model, we have filled those missing values with the most frequent value in that attribute for categorical and with mean value for the numerical attributes.

Treating Outliers:

For treating the outliers in the dataset, we have used capping method. In this method we will replace the value of the outlier with either maximum or minimum value of the attribute that is not an outlier. If the current outlier is

1000 and the maximum value and minimum value of the attribute is 100 and 0, we replace that outlier value with 100 as 100 is near to 1000 compared to 0.

Label Encoding:

As many machine learning algorithms require numerical input, we will convert our categorical attributes into numerical attributes. This is done by attaching a unique identifier to each category.

2.3 Training models:

In this study, we aim to predict the survival of breast cancer from gene expression profiles using machine learning classification models. The training models for prediction considered are K-Nearest Neighbor (KNN), Logistic Regression (LR), Random Forest (RF), and Catboost Classification model. We are trying to predict the performance of the classification models to see which one gives the better and optimized results. We can see the overview of these models as follows.

2.3.1 K-Nearest Neighbors (KNN) model:

K-nearest neighbors (KNN)[9] is a non-parametric and supervised machine learning algorithm used for classification and regression tasks. In KNN, the classification of a data point is determined by the majority class among its k-nearest neighbors in the feature space. In other words, it assumes that data points with similar features are likely to belong to the same class.

For a given data point, the algorithm identifies the k-nearest neighbors based on a distance metric (such as Euclidean distance) in the feature space. The class label of the majority of these neighbors is then assigned to the target data point. It is considered a lazy learning algorithm, as it doesn't build a model during training but rather makes predictions at the time of testing based on the stored training data.

2.3.2 Logistic Regression (LR) model:

Logistic regression [6] is a statistical method and a type of regression analysis used for predicting the probability of an outcome in binary or categorical data. Despite its name, logistic regression is used for classification rather than regression tasks.

In logistic regression, the dependent variable is binary, meaning it has only two possible outcomes (usually denoted as 0 and 1). The logistic regression model uses the logistic function (also called the sigmoid function) to model the probability that a given input belongs to a particular category.

The linear regression equation would be:

$$y = b_0 + b_1 x \,. \tag{1}$$

Using the sigmoid function (logistic function) whose equation is:

$$P = \frac{1}{1+e^{-y}}. \quad (2)$$

The logistic function ensures that the predicted probabilities lie between 0 and 1.

Taking the y value form Eq (1) and substitute in the Eq (2), we can get the LR function as:

$$\ln(\frac{p}{1-p}) = b_0 + b_1 x. \quad (3)$$

If the probability is above a certain threshold (commonly 0.5), the model predicts one class; otherwise, it predicts the other.

2.3.3 Random Forest (RF) classification model:

Random Forest[7] is a powerful ensemble learning algorithm employed for classification tasks. This algorithm consists of an ensemble, or a collection, of decision trees. Unlike a single decision tree, each tree in the Random Forest is constructed using a random subset of the features at each node. This introduces diversity among the trees, which is crucial for preventing overfitting and enhancing the model's generalization capabilities. During training, the algorithm employs a process called bootstrap sampling, creating multiple subsets of the training data by random sampling with replacement. Each decision tree is then grown to its maximum depth on one of these subsets. When it comes to making predictions, the Random Forest aggregates the outputs of individual trees through a voting mechanism. The class that receives the most votes becomes the final predicted class. One notable strength of Random Forest lies in its ability to automatically perform feature selection, as it focuses on a subset of features at each split. Moreover, the ensemble nature of Random Forest contributes to its robustness and accuracy, often outperforming individual decision trees and mitigating the risk of overfitting on the training data.

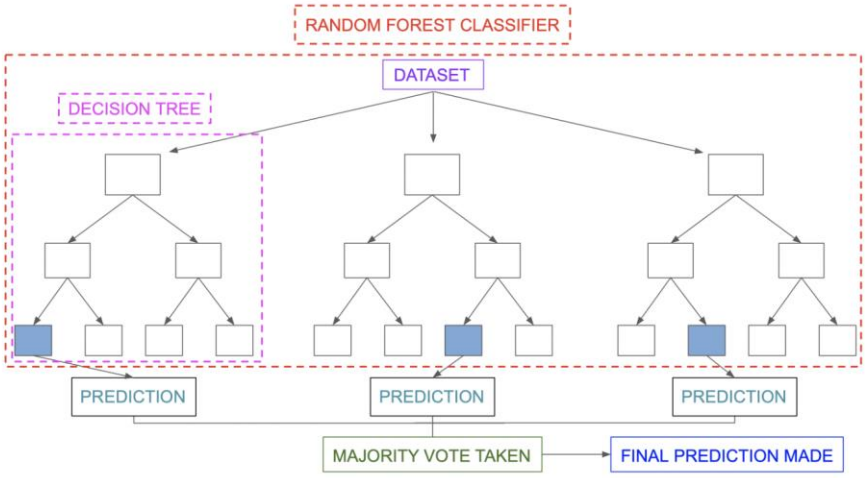


Figure 5. Random Forest Classifier Model Workflow [8]

2.3.4 CatBoost Classification model:

CatBoost, or Categorical Boosting [6], is a machine learning technique that is especially developed for dealing with categorical data in classification problems. It belongs to the boosting algorithm family, which assembles a group of weak learners to form a robust prediction model. CatBoost is distinguished by its ability to handle categorical data effectively without the need for manual preprocessing, such as one-hot encoding. It uses conditional decision trees as its primary predictors. The process of constructing a decision tree model entails recursively dividing the feature space into many tree nodes, using the values of different splitting properties. Binary variables serve as parameters, indicating whether various features surpass a predetermined threshold.

Here, we can observe the results from the trained models with default hyperparameters and discuss them in detail as follows:

	model	train_accuracy	test_accuracy	test_precision	test_recall	test_f1	test_roc_auc
0	LogisticRegression	0.765	0.740	0.689	0.667	0.678	0.804
1	CatBoostClassifier	0.947	0.790	0.753	0.724	0.739	0.859
2	RandomForestClassifier	1.000	0.769	0.750	0.654	0.699	0.832
3	KNeighborsClassifier	0.806	0.659	0.587	0.564	0.575	0.706

Figure 6: *Comparision of default trained models*

KNN – this model obtained a training accuracy of 80.6% and a testing accuracy of 65.9%. Precision, recall, and F1 score on the test set were 58.7%, 56.4%, and 57.5%, respectively. The ROC AUC score was 70.6%. This model shows moderate performance, but it might not be the most suitable for our data features, as it struggles with both precision and recall on the test set.

LR - The Logistic Regression model achieved a training accuracy of 76.5% and a testing accuracy of 74.0%. The precision, recall, and F1 score on the test set were 68.9%, 66.7%, and 67.8%, respectively. The ROC AUC score, measuring the model's ability to distinguish between classes, was 80.4%. The model demonstrates reasonable performance and better one compared to KNN.

RF - The RandomForestClassifier achieved perfect training accuracy (100%), indicating it might have memorized the training data. The testing accuracy was 76.9%. Precision, recall, and F1 score on the test set were 75.0%, 65.4%, and 69.9%, respectively. The ROC AUC score was 83.2%. The model appears to have overfit the training data, as indicated by the perfect training accuracy and a drop in performance on the test set.

CatBoost - The CatBoostClassifier outperformed other models with a high training accuracy of 94.7%, and the testing accuracy with 79.0%. Precision, recall, and F1 score on the test set were 75.3%, 72.4%, and 73.9%, respectively. The ROC AUC score was notably high at 85.9%. This model exhibits strong overall performance, but there could be a slight indication of overfitting, as the training accuracy is substantially higher than the testing accuracy.

We can increase the performance of our trained prediction models with the hyperparameter tuning to find the best hyperparameters set which gives the optimal results among others.

2.4 Hyperparameter Tuning:

Hyperparameter tuning involves identifying the most ideal combination of hyperparameters for a machine learning model to attain the highest level of performance. Hyperparameters are predetermined configuration values that must be specified prior to training a model and are not derived from the data. Examples include the learning rate in gradient boosting, the regularization strength in linear models, or the number of layers and neurons in a neural network.

The objective of hyperparameter tuning is to systematically investigate various combinations of hyperparameter values and determine the combination that yields the optimal model performance. The selection of hyperparameters is of utmost importance as it can greatly influence a model's capacity to effectively generalize to new, or unseen data.

There are several methods for hyperparameter tuning, ranging from exhaustive search to more sophisticated optimization algorithms. Among them, we are using Grid Search in this project for obtaining the optimal hyperparameters. Grid search is a simple and direct approach in which a grid of hyperparameter values are defined for exploration. The algorithm subsequently assesses the model's performance for every combination of hyperparameters within the specified grid.

We are implementing the grid search in our project with the help of a python package called ‘GridSearchCV’. The GridSearchCV is instantiated with the model, parameter grid, the number of folds for cross-validation (cv),

and a scoring metric (in this case, 'accuracy'). The fit method is then called to perform the grid search with cross-validation. After performing the grid search, we can get the best hyperparameters values with the attribute 'best_params_'. From this, we have found the optimal parameters for KNN, CatBoost and Logistic Regression algorithms, and then compared the accuracies of these fine-tuned models.

	model	train_accuracy	test_accuracy	test_precision	test_recall	test_f1	test_roc_auc
0	LogisticRegression	0.765	0.740	0.689	0.667	0.678	0.804
1	CatBoostClassifier	0.947	0.790	0.753	0.724	0.739	0.859
2	RandomForestClassifier	1.000	0.769	0.750	0.654	0.699	0.832
3	KNeighborsClassifier	0.806	0.659	0.587	0.564	0.575	0.706
4	Tune-CatBoostClassifier	0.890	0.780	0.743	0.705	0.724	0.857
5	Tune-RandomForestClassifier	0.957	0.777	0.735	0.712	0.723	0.837
6	Tune-KNeighborsClassifier	1.000	0.685	0.630	0.558	0.592	0.736

Figure 7. Comparing scores of Hyper Tuned Models

In the provided figure, it is observed that the training accuracies decreased after tuning the models, indicating a move towards improved generalization and a reduction in overfitting. Conversely, the test accuracies showed improvement after tuning the models. However, it's noteworthy that the test accuracy of the CatBoost model remained almost same even after the tuning process.

2.5 Oversampling

From figure 2 in Target Distribution, we can see that there is a class imbalance, so the models may not adequately capture the patterns and characteristics of the minority class, leading to poor generalization for the underrepresented class. To overcome this, we can use oversampling techniques to balance the target class. There are several oversampling techniques, but we have used Synthetic Minority Over-sampling Technique (SMOTE) for this project.

SMOTE, also known as Synthetic Minority Over-sampling Technique, is an oversampling that is employed in datasets with uneven class distribution to mitigate the problem of under-representation of the minority class. The process involves creating artificial samples for the underrepresented class using existing examples, hence achieving a balanced distribution of classes. SMOTE algorithm generates synthetic instances by interpolating between existing samples of the minority class, thereby addressing class imbalance without duplicating the current data. SMOTE seeks to enhance model performance and mitigate classifier bias towards the majority class by using synthetic data points. This approach improves the accuracy and reliability of predictive models in situations where data distributions are uneven. We have implemented this in our project with the help of a python library called 'imbalanced-learn', which provides an easy-to-use interface for various techniques addressing class imbalance.

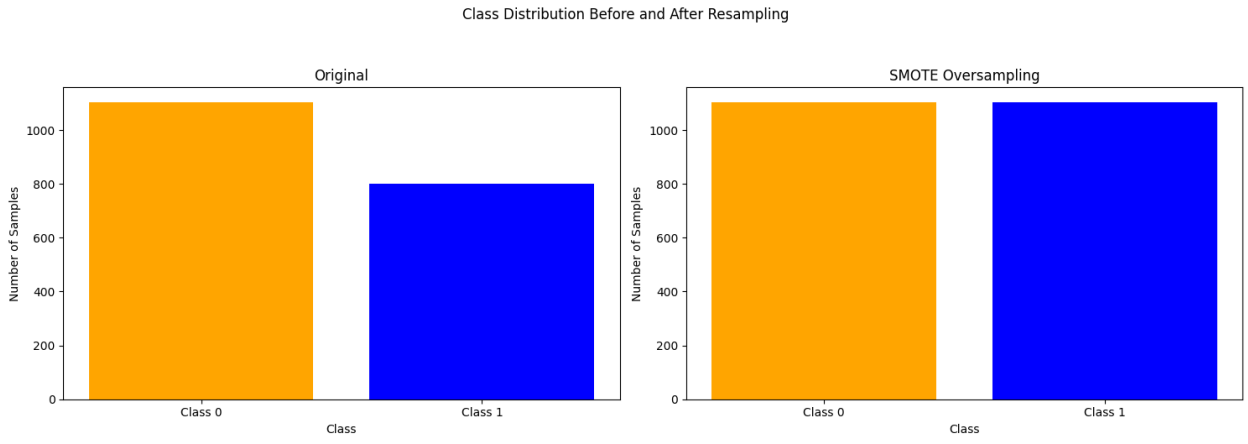


Figure 8. Class distribution before and after resampling

2.6 Ensemble models for Hybrid model:

So far, we have created four classification models and enhanced their accuracy by fine-tuning them. In addition, we have tackled the issue of class imbalance to improve performance. To enhance accuracy, we are now using ensemble methods. Ensemble methods involves the combination of several individual models to construct a more powerful and robust model. Among the several types of ensemble models available, we decided to use a Voting Classifier for this project to enhance the accuracy of our predictions.

A voting classifier is a technique in machine learning that combines the predictions of multiple independent classifiers or models to make an overall decision. The concept is to utilize a variety of distinct models to enhance overall performance and generate more robust predictions. We have utilized the 'VotingClassifier' class from the 'scikit-learn' library to build a Voting classifier in this project.

We created two hybrid models using a Voting Classifier. One model is constructed by combining the Random Forest and CatBoost models, while another model is built by combining the Random Forest and Linear Regression. We have employed soft voting in both hybrid models. The following are the performance metrics of the two hybrid models.

	model	train_accuracy	test_accuracy	test_precision	test_recall	test_f1	test_roc_auc
0	Hybrid Model (RF+Catboost)	0.928	0.830	0.821	0.840	0.831	0.907
1	Hybrid Model (RF+LR)	0.869	0.814	0.810	0.817	0.814	0.892
2	CatBoostClassifier	0.947	0.790	0.753	0.724	0.739	0.859
3	Tune-CatBoostClassifier	0.890	0.780	0.743	0.705	0.724	0.857
4	Tune-RandomForestClassifier	0.951	0.774	0.754	0.667	0.707	0.835
5	RandomForestClassifier	1.000	0.769	0.750	0.654	0.699	0.832
6	LogisticRegression	0.765	0.740	0.689	0.667	0.678	0.804
7	Tune-KNeighborsClassifier	1.000	0.685	0.630	0.558	0.592	0.736
8	KNeighborsClassifier	0.806	0.659	0.587	0.564	0.575	0.706

Figure 9. *Comparing scores of Hybrid Models*

From the above figure, we can observe both the hybrid models performed well and have given better results compared to all the other models built. Among the two hybrid models, the one formed by combining Random Forests and CatBoost showed superior results, achieving a test accuracy of 0.83 and a test ROC AUC score of 0.907.

2.7 Performance Metrics:

1. **Accuracy**:

   Accuracy measures the proportion of correctly classified instanced out of total instances. This provides a general view of how well the model is performing across both classes. A higher accuracy score indicates better model performance. The mathematical formula can be given as below.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2. **Precision:**

   Precision measures the proportion of true positive predictions among all positive predictions made by the model. This is important when the cost of false positives is high. The mathematical formula for Precision can be given as.

$$Precision = \frac{TP}{TP + FP}$$

3. **Recall:**

   Recall measures the proportion of true positive predictions among all actual positive instances. This is important when the cost of false negatives is high. A higher recall score indicates fewer false negatives. The mathematical formula for recall can be given as.

$$Recall = \frac{TP}{TP + FN}$$

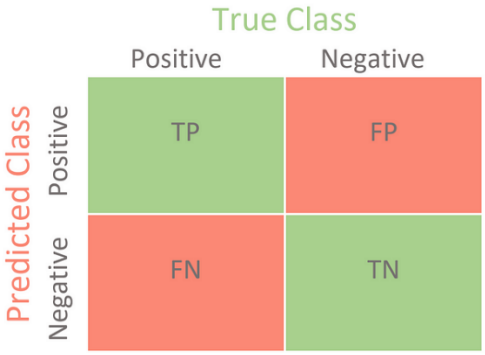
4. **F1 Score:**

   F1-Score is the harmonic means of precision and recall, providing a balance between these two metrics. This is useful when there is an uneven class distribution. A higher F1-Score indicates a better balance between precision and recall. The mathematical formula for F1-Score is given by.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

5. **Confusion Matrix:**

   A Confusion Matrix is a table that summarizes the performance of a classification algorithm, it shows counts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions. This matrix helps us visually understand model's strengths and weaknesses in terms of number of correctly and incorrectly classified instances. Confusion matrix looks like the below figure for a binary classification task.



6. **ROC Curve:**

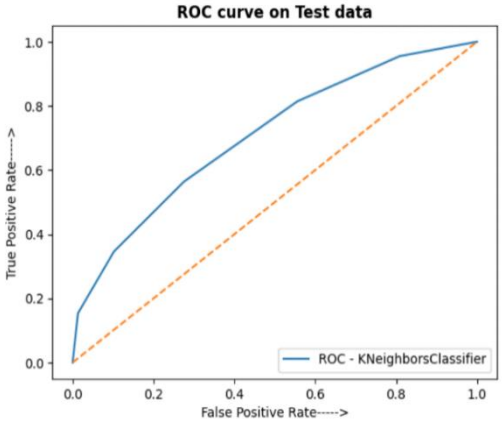
   The ROC (Receiver Operating Characteristic) curve is a graphical representation of the model's ability to discriminate between the positive and negative classes at various classification thresholds. It plots the True Positive Rate (Sensitivity) against the False Positive Rate (1 - Specificity). This can be used to evaluate the trade-off between sensitivity and specificity at different classification thresholds. The Area Under the Curve (AUC) quantifies the overall discriminatory power of the model. An AUC of 0.5 suggests no discrimination (like random guessing), while an AUC of 1.0 indicates perfect discrimination.

3 RESULTS & DISCUSSION:

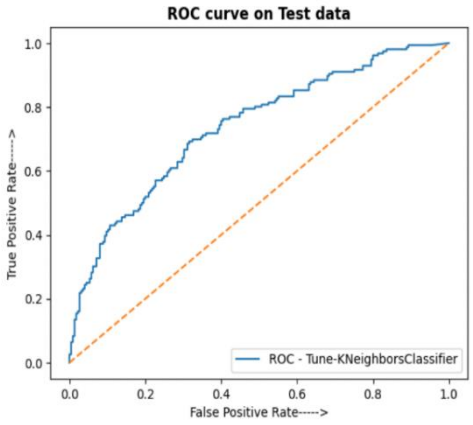
In order to implement it, we used the python coding language on the Google Collaborator Platform. The advantage of this platform is that we have machine learning models as inbuilt libraries. We can discuss the results of the trained models in this study with the performance metrics discussed in the above section.

The ROC plots depict the performance of various models in terms of their ability to discriminate between positive and negative classes. In the Hybrid Model (RF+Catboost), the ROC plot demonstrates robust performance with an area under the curve (AUC) of 0.933, indicating a high true positive rate across different classification

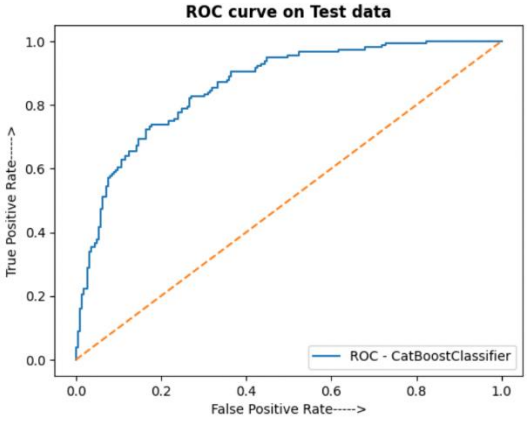
thresholds. Similarly, the Hybrid Model incorporating Random Forest and Logistic Regression components demonstrates consistent and competitive ROC performance with 89.1%, suggesting a synergistic effect of combining these models. The CatBoostClassifier, being a standalone model, displays a commendable AUC of 0.859, indicating its ability to distinguish between classes. The tuned versions of CatBoostClassifier and RandomForestClassifier also perform well, with AUC values of 0.857 and 0.837, respectively. On the other hand, the standalone Logistic Regression and KNeighborsClassifier models show comparatively lower AUC values, highlighting the advantage of ensemble models in achieving superior classification performance.



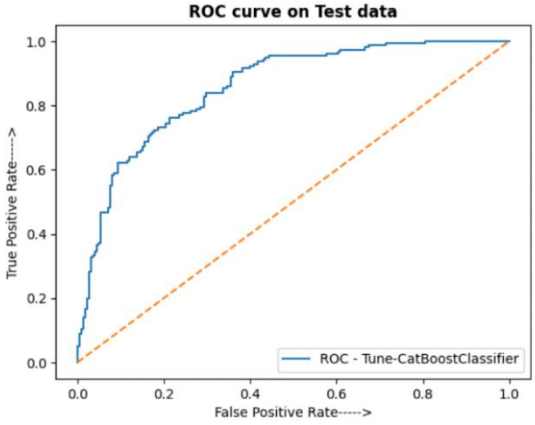
a. KNN model



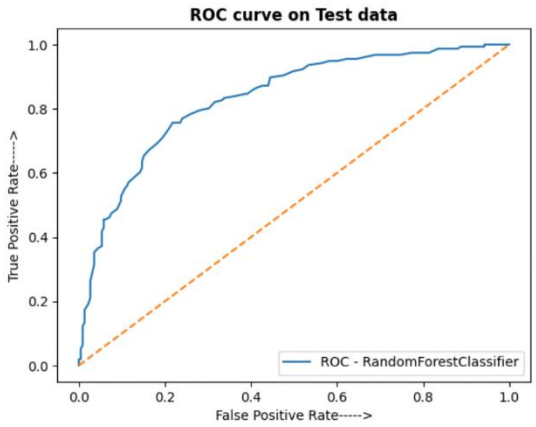
b. Tuned – KNN model



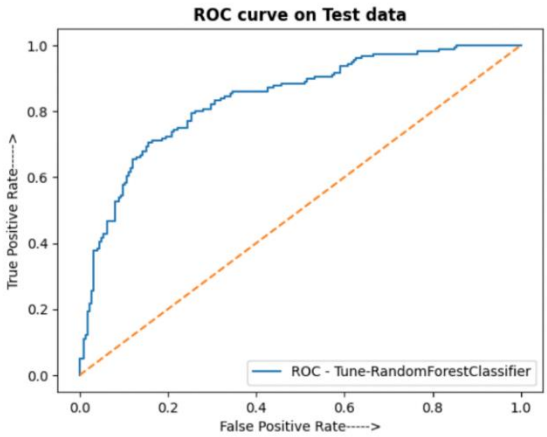
c. CatBoost Classifier



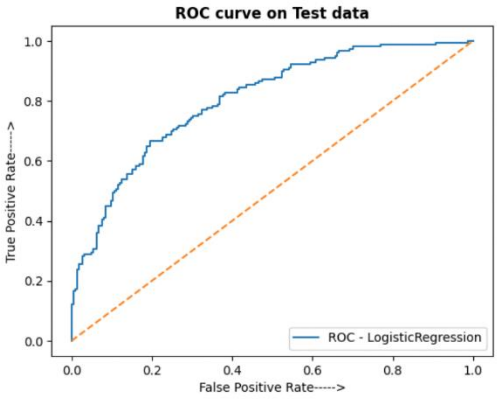
d. Tuned – CatBoost Classifier



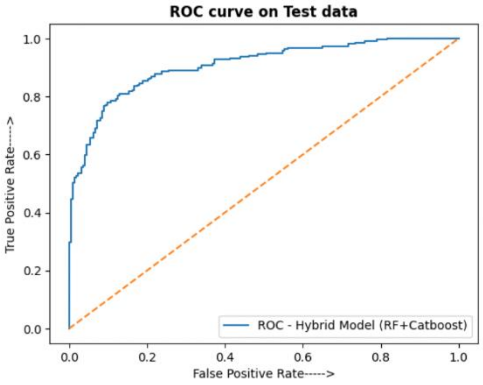
e. RandomForestClassifier Model



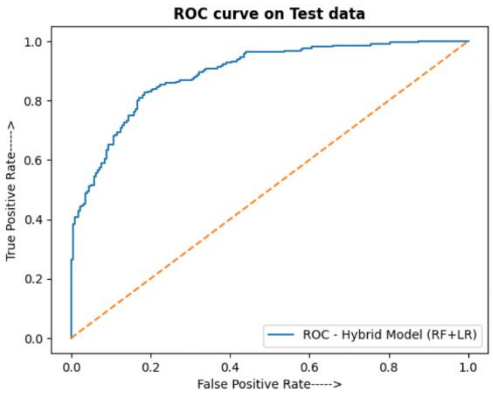
f. Tuned – RandomForestClassifier Model



g. Logistic Regression Model



h. Hybrid Model (RF + CatBoost)



i. Hybrid Model (RF+LR)

Figure 10: ROC curves of all the trained models on breast cancer diagnosis.

The bar graph provides a visual representation of the models' performance in the study. Notably, the Hybrid Model, which integrates Random Forest and CatBoost models, stands out with the highest performance, indicating its effectiveness in the breast cancer survival diagnosis. As we move along the horizontal axis, representing different models, there is a gradual decrease in performance. Examining the training accuracies, it becomes evident that both the Tuned KNN and Random Forest models exhibit the highest values among all models.

However, it is important to note that such high training accuracies may be indicative of overfitting, where the model may excel in capturing the training data patterns but could potentially struggle with generalization to unseen data.

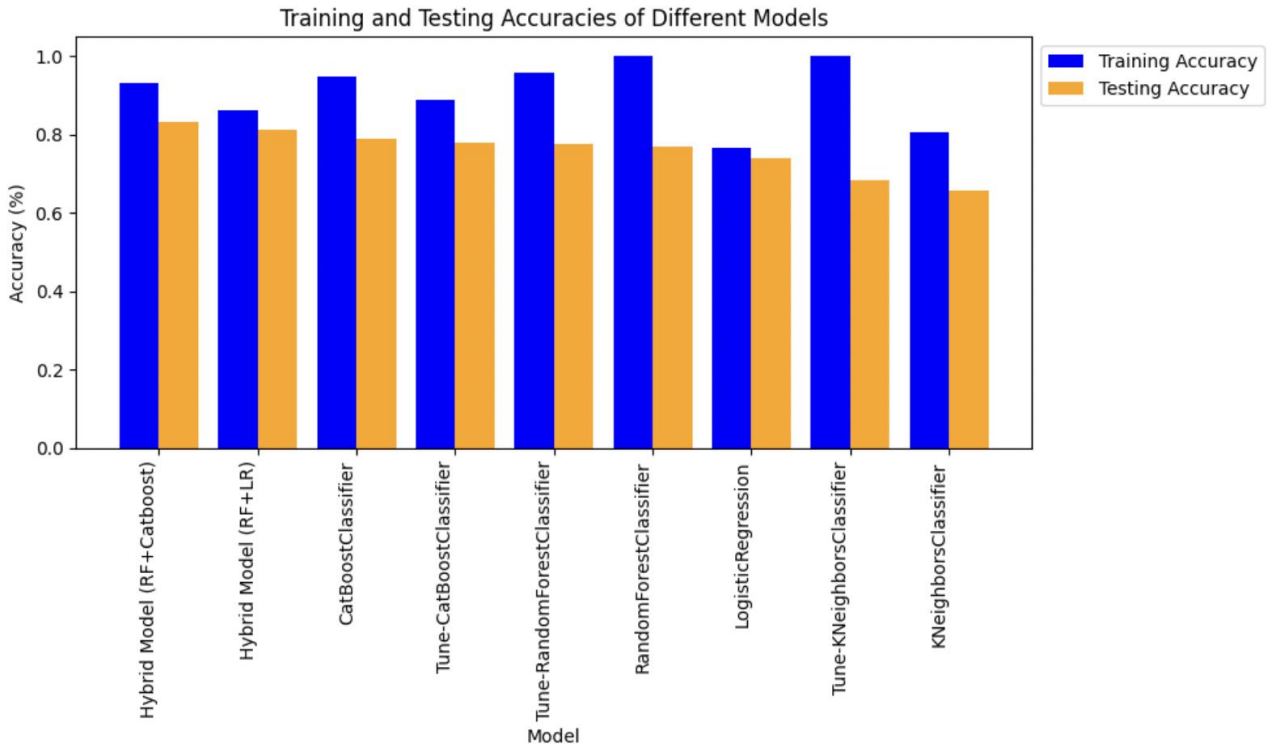


Figure 11: Bar graph for the training and testing accuracies of all models

4 CONCLUSION:

In summary, our project employed various classification algorithms, including Catboost, Random Forest Classifier, and logistic regression, both in their standard and tuned versions. Among these models, Catboost algorithm demonstrated the highest accuracy at 79%, followed by a tuned Random Forest Classifier with an accuracy of 77.4%, surpassing the performance of other models. Further, we came up with a hybrid model in a combination of Random Forest and catboost which accuracy of 83%. We have observed that this hybrid model consists of random forest and catboost outperformed all other models.

REFERENCES:

1. Pereira B, Chin SF, Rueda OM, Vollan HK, Provenzano E, Bardwell HA, Pugh M, Jones L, Russell R, Sammut SJ, Tsui DW, Liu B, Dawson SJ, Abraham J, Northen H, Peden JF, Mukherjee A, Turashvili G, Green AR, McKinney S, Oloumi A, Shah S, Rosenfeld N, Murphy L, Bentley DR, Ellis IO, Purushotham A, Pinder SE, Børresen-Dale AL, Earl HM, Pharoah PD, Ross MT, Aparicio S, Caldas C. “The somatic mutation profiles of 2,433 breast cancers refines their genomic and transcriptomic landscapes.” Nat Commun. 2016 May 10; 7:11479. doi: 10.1038/ncomms11479. PMID: 27161491; PMCID: PMC4866047.
2. Rueda OM, Sammut SJ, Seoane JA, Chin SF, Caswell-Jin JL, Callari M, Batra R, Pereira B, Bruna A, Ali HR, Provenzano E, Liu B, Parisien M, Gillett C, McKinney S, Green AR, Murphy L, Purushotham A, Ellis IO, Pharoah PD, Rueda C, Aparicio S, Caldas C, Curtis C. “Dynamics of breast-cancer relapse reveal

late-recurring ER-positive genomic subgroups." Nature. 2019 Mar;567(7748):399-404. doi: 10.1038/s41586-019-1007-8. Epub 2019 Mar 13. PMID: 30867590; PMCID: PMC6647838.

3. Curtis C, Shah SP, Chin SF, Turashvili G, Rueda OM, Dunning MJ, Speed D, Lynch AG, Samarajiwa S, Yuan Y, Gräf S, Ha G, Haffari G, Bashashati A, Russell R, McKinney S; METABRIC Group; Langerød A, Green A, Provenzano E, Wishart G, Pinder S, Watson P, Markowetz F, Murphy L, Ellis I, Purushotham A, Børresen-Dale AL, Brenton JD, Tavaré S, Caldas C, Aparicio S. "The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups." Nature. 2012 Apr 18;486(7403):346-52. doi: 10.1038/nature10983. PMID: 22522925; PMCID: PMC3440846.
4. Mucaki EJ, Baranova K, Pham HQ, Rezaeian I, Angelov D, Ngom A, Rueda L, Rogan PK. "Predicting Outcomes of Hormone and Chemotherapy in the Molecular Taxonomy of Breast Cancer International Consortium (METABRIC) Study by Biochemically-inspired Machine Learning." F1000Res. 2016 Aug 31;5:2124. doi: 10.12688/f1000research.9417.3. PMID: 28620450; PMCID: PMC5461908.
5. Milioli, H.H., Vimieiro, R., Tishchenko, I. et al. Iteratively refining breast cancer intrinsic subtypes in the METABRIC dataset. BioData Mining 9, 2 (2016). https://doi.org/10.1186/s13040-015-0078-9
6. Harshit Gupta, Pritam Kumar, Subham Saurabh, et al. "CATEGORY BOOSTING MACHINE LEARNING ALGORITHM FOR BREAST CANCER PREDICTION." Rev. Roum. Sci. Techn.– Électrotechn. et Énerg.Vol.66, 3, pp. 201–206, Bucarest, 2021.
7. Weiwei Lin, Ziming Wu, Longxin Lin, et al. "An Ensemble Random Forest Algorithm for Insurance Big Data Analysis." Digital Object Identifier 10.1109/ACCESS.2017.2738069.
8. "Introduction to Random Forest in Machine Learning." https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/
9. Seyyid Ahmed Medjahed, Tamazouzt Ait Saadi, Abdelkader Benyettou. "Breast Cancer Diagnosis by using k-Nearest Neighbor with Different Distances and Classification Rules." International Journal of Computer Applications (0975 - 8887) Volume 62 - No. 1, January 2013.