Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IMPACT FACTOR: 7.056

IJCSMC, Vol. 10, Issue. 9, September 2021, pg.1 – 11

Applying Ensemble Approach on U.S. Census Data Classification

Raj Kumar Pal; Jugal Chaturvedi; V. Sai Teja
Data Science and Machine Learning, PES University, Bengaluru, Karnataka
raajkumarpal@yahoo.com; chaturvedijugal@gmail.com; babbulusai7@gmail.com
Leena Shibu
Data Science and Machine Learning, Great Learning, Bengaluru, Karnataka
leena@greatlearning.in

DOI: 10.47760/ijcsmc.2021.v10i09.001

Abstract— During this paper, we have a tendency to examine the adult financial gain dataset obtainable at the UC Irvine Machine Learning Repository. To aim predict whether or not associate individual's financial gain are going to be bigger than $50,000 per annum victimization completely, different boosting and bagging strategies and compare models supported many attributes from the census information.

Keywords- Precision, Recall, Specificity, F1 Score, Accuracy, Ture Positive (TP), True Negative(TN), False Positive(FP), False Negative(FN), Recursive Feature Elimination(RFE) , KNN, Random Forest(RF), Ada Boost, Gradient Boost, Extreme Gradient Boost (XGBoost).

I. INTRODUCTION

The United States adult census dataset could be a repository of forty-eight thousand eight hundred forty-two (48842) entries row wise and total 15 columns extracted from the 1994 United States census info. In initial section, to explore the information as to grasp the trends and representations of sure demographics within the corpus, then this information in section 2 it consisting of 2 stages, the first one is feature engineering and prediction. Feature engineering contains of feature extraction and have choice that is feature selection .In the third section, we glance into some algorithms like RFE[5], Random Forest[6], KNN[4] & boosting strategies[1,2,3] to seek out fee what strategies they're victimization to achieve insight on constant information. finally, within the fourth section, compare all models moreover as that of others so as to seek out fee what options area unit of significance, what strategies area unit best, associated gain an understanding of a number of the intuition behind the numbers.

II. EXPLORATORY ANALYSIS

The Census financial gain dataset has forty-eight thousand eight hundred forty-two (48842) entries row wise and total 15 columns. every entry contains the subsequent information.

.. Data set

- **age**: the age of a private
- **work class:** a general term to represent the use standing of a private

- **fnlwgt**: final weight. In different words, this can be the number of individuals the census believes the entry represents.

education: the very best level of education achieved by a private.

- **education-num:** the very best level of education achieved in numerical kind.
- **marital-status:** legal status of a private. Married-civ-spouse corresponds to a civilian relation whereas Married-AF-spouse could be a relation within the soldiers.
- **occupation:** the overall style of occupation of a private
- **relationship:** represents what this individual is relative to others. for instance, a private can be a Husband. every entry solely has one relationship attribute and is somewhat redundant with legal status. we'd not build use of this attribute in the least.
- **race:** Descriptions of associate individual's race
- **sex**: the biological sex of the individual
- **capital-gain:** capital gains for a private
- **capital-loss:** financial loss for a private
- **hours-per-week:** the hours a private has reported to figure per week
- **native-country:** country of origin for a private

the label: whether or not or not a private makes over $50,000 annually**.**

The original dataset contains a distribution of 24.1% entries labelled with greater than 50k and 75.9% entries labelled with less than equal to 50k. Now split the dataset into coaching and check sets whereas maintaining the distribution and the subsequent graphs, statistics pertain to the coaching set hopes to spot options that give very little info so as to modify the model's quality and runtime. The distribution information was shown below

Table 1: Salary Percentage

Label	Number	Percentage
<= 50k	7841	24.1
> 50k	24720	75.9

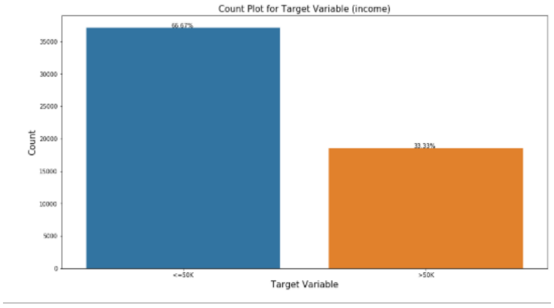


Figure 1: Income Distribution

A. Distribution of Variable

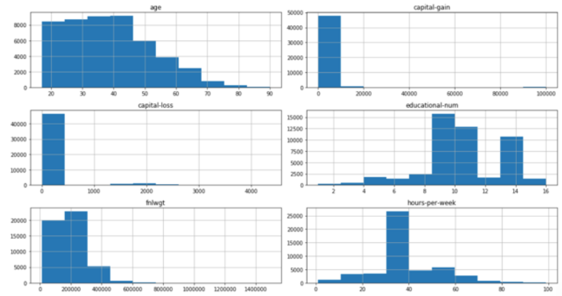


Figure 2:Distribution of Variables

Age, capital-gain, capital-loss, educational-num, fnlwgt, hours-per-week these all parameters are mostly right skewed.

B. Feature Engineering

This data set contains Education column with - 'Bachelors', 'HS-grad', 'Masters', 'Some-college', 'Prof-school', '10th', 'Assoc-acdm', '11th', '9th', '7th-8th', 'Doctorate', '1st-4th', '5th-6th', '12th', 'Assoc-voc', 'Preschool' these many unique values. With the help of feature engineering these values converted to total 6 category which are **doctorate**, **masters**, **bachelors**, **college**, **highschoolgrad** and **dropout**. Likewise for 'marital-status' converted to **NotMarried**, **Married**, **Separated** and **Widowed**.

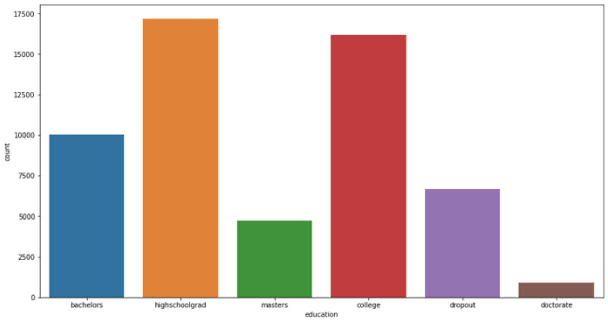


Figure 3: Count plot for 'education'

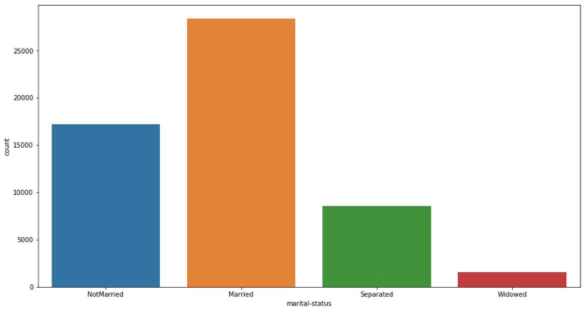


Figure 4: Count plot for 'marital-status'

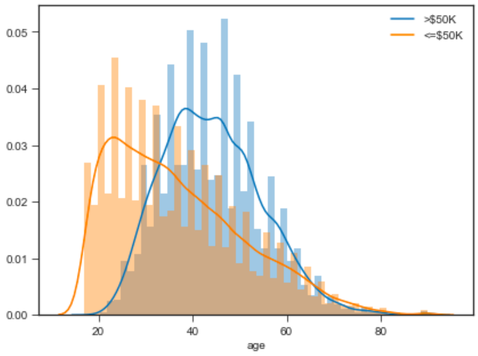


Figure 3: Distribution plot for age vs salary

Age of 40 to 44 getting the 50k and above salary concluded from the above distribution plot.

III. PRELIMINARIES

In this section we give brief description of various methodologies with their mathematical notations.

A. Confusion Matrix

This is kind of a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known. It is also known as an Error Matrix. The Confusion matrix gives a lot of information, but sometimes it may prefer you a more concise metric. These metrices are PRECISION, RECALL, SPECIFICITY, F1 SCORE and ACCURACY[8].

- **Precision**: This metric quantifies the number of positive class predictions that actually belong to the positive class.
  Formula: *TP / (TP + FP)*

- **Recall**: Recall quantifies the number of positive class predictions made out of all positive examples in the dataset.
  Formula: *TP / (TP / FN)*

- **Specificity**: It is ration of true negatives to the total actual negative observation.
  Formula: *TN / (TN + FP)*

- **F1 Score**: This metric provides a single score that balances both the concerns of precision and recall in one number.
  Formula: *2*(precision*recall)/(precision+recall)*

- **Accuracy**: It is the ratio of correct predictions i.e. True Negative + True Positive.
  Formula: *(TN+TP) / (TN+FP+FN+TP)*

B. Kappa Score: It is a measure of inter-rater reliability.

C. Youden's Index: Youden's Index is the classification cut-off probability for which the (Sensitivity + Specificity - 1) is maximized.

Youden's Index = max(Sensitivity + Specificity - 1)
= max(TPR + TNR - 1)
= max(TPR - FPR)

D. ROC Curve: Receiving Operating Characteristic Curve is kind of graph showing the performance of a classification model at all classification thresholds[7]. This curve plots two parameters: a.) True Positive Rate(TPR) b.)False Positive Rate(FPR).

- TPR = TP / TP + FN
- FPR = FP / FP + TN

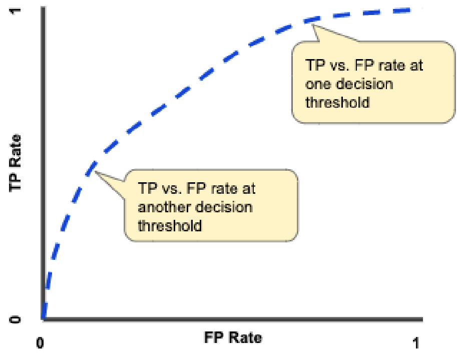


Figure 4: ROC Curve

E. AUC: Area Under the ROC Curve is measures the entire two-dimensional area underneath the entire ROC curve from (0,0) to (1,1)[7].

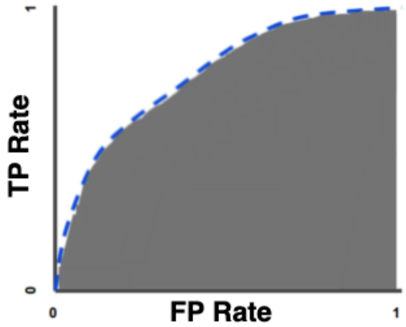


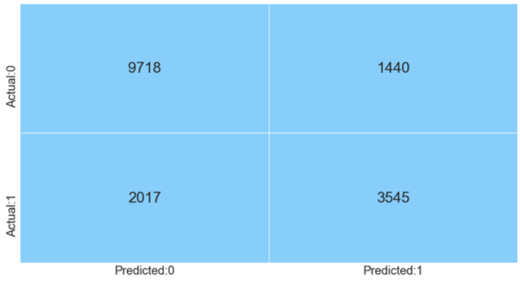
Figure 5: AUC Curve

IV. Classification Model

In this section we divided the data train-test-split part into 70-30 ratio (train data size is 70% and test data size is 30%) and apply various algorithm to compare their accuracy.

Algorithm 1: Method - Logistic Regression (LR)

Using LR algorithm we divided the data into train and test part in 70:30 ratio, and we built a basic model



	Predicted:0	Predicted:1
Actual:0	9718	1440
Actual:1	2017	3545

Figure 6: Confusion matrix

True Positive (TP) = 3545; meaning 3545 positive class data points were correctly classified by the model

True Negative (TN) = 9718; meaning 9718 negative class data points were correctly classified by the model

False Positive(FP) = 1440; meaning 1440 negative class data points were incorrectly classified as belonging to the positive class by the model

False Negative(FN) = 2017; meaning 2017 positive class data points were incorrectly classified as belonging to the negative class by the model

Table 2: Score Comparison

From the above table we can infer that accuracy is 79% without replacing unknown value and 85%.

	precision	**recall**	**f1_score**	**accuracy**
With '?'	0.71	0.63	0.67	0.79
Replaced '?'	0.72	0.60	0.66	0.85

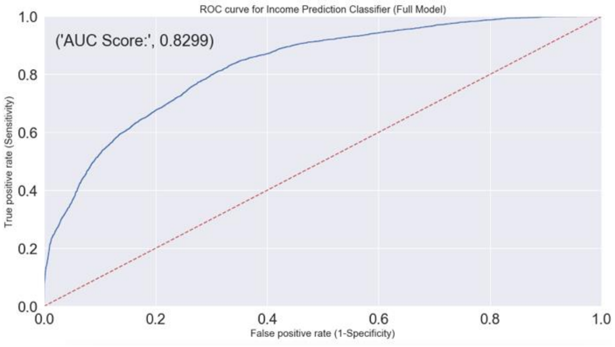


Figure 9: ROC Curve based on LR Model

In **Figure 9** the red dotted line represents the ROC curve of a purely random classifier; a good classifier stays as far away from that line as possible (toward the top-left corner). From the above plot, we can see that our classifier (logistic regression) is away from the dotted line; with the AUC score 0.8299.

Algorithm 2: Method - Recursive Feature Elimination (RFE)

RFE is a wrapper-type feature selection algorithm. This means that a different machine learning algorithm is given and used in the core of the method, is wrapped by RFE, and used to help select features

Table 3: Score Comparison

	precision	**recall**	**F1_score**	**accuracy**
With '?'	0.58	0.86	0.69	0.75
Replaced '?'	0.50	0.85	0.63	0.76

From the above table we can infer that the recall of the positive class is known as sensitivity and the recall of the negative class is specificity and the accuracy is 75% with '?' and 76% of accuracy with replaced '?'.

Algorithm 3: Method – K-Nearest Neighbours (KNN)

Suppose there are two categories, i.e., Category A and Category B, and a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a KNN algorithm.

Table 4: Score Comparison

	precision	**recall**	**f1_score**	**accuracy**
With '?'	0.62	0.62	0.62	0.75
Replaced '?'	0.56	0.33	0.42	0.77

From the above table we can infer. that accuracy is 75% with '?' and 77% of accuracy with replaced '?'

Algorithm 4: Method – Random Forest (RF)

Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction, one big advantage of random forest is that it can be used for both classification and regression problems.

Table 5: Score Comparison

	precision	**recall**	**F1_score**	**accuracy**
With '?'	0.84	0.88	0.86	0.90
Replaced '?'	0.72	0.62	0.67	0.85

From the above table we can infer. that accuracy is 90% with '?' and 85% of accuracy with replaced '?'

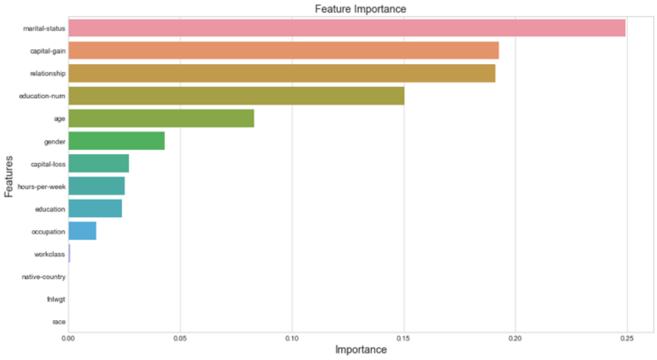


Figure 10: Bar plot for Feature importance

From **Figure 10**, above we can see the 'marital-status', 'capital-gain', 'relationship', 'education-num' - these are the important features followed by 'age', 'gender', 'capital-loss'.

Algorithm 5: Method – Ada Boost

Boosting[13] is one of the techniques that uses the concept of ensemble learning. A boosting algorithm combines multiple simple models (also known as weak learners or base estimators) to generate the final output. We will look at some of the important boosting algorithms in this paper and compare each algorithm.

Let us build the AdaBoost [14] classifier with decision trees. The model creates several stumps (decision tree with only a single decision node and two leaf nodes) on the train set and predicts the class based on these weak learners (stumps). For the first model, it assigns equal weights to each sample. It assigns the higher weight for the wrongly predicted samples and lower weight for the correctly predicted samples. This method continues till all the observations are correctly classified or the predefined number of stumps is created.

	precision	**recall**	**F1_score**	**accuracy**
With '?'	0.76	0.70	0.73	0.83
Replaced '?'	0.75	0.59	0.66	0.85

Table 6: Score Comparison

From the above table we can infer that the recall of the positive class is sensitivity and the recall of the negative class is specificity. We can see that accuracy is 83% without replacing the unknown value and 85% of accuracy by replacing unknown value

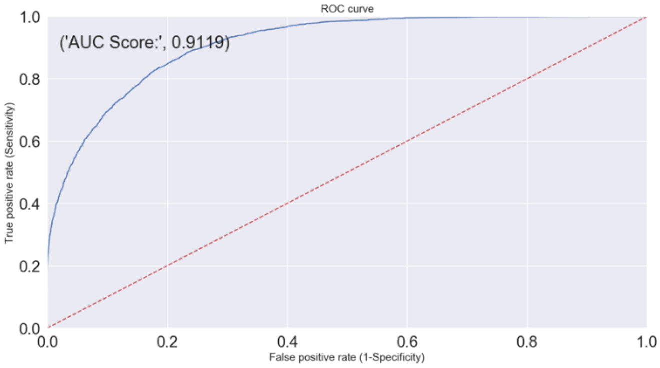


Figure 11: ROC Curve for Ada Boost model

In **Figure 11**, The red line represents the ROC curve of a strictly random classifier. A decent classifier stays as distant from that line as doable (toward the top-left corner). From the on top of plot, we will see that the AdaBoost model is removed from the dotted line with the AUC score 0.9128.

Algorithm 6: Method – Gradient Boosting

This Gradient Boosting [15] technique optimizes the differentiable loss, perform by building the number of weak learners (decision trees) consecutive. It considers the residuals from the previous model and fits consequent model to the residuals. The rule uses a gradient descent technique to reduce the error.

	precision	**recall**	**f1_score**	**accuracy**
With'?'	0.81	0.78	0.79	0.86
Replaced '?'	0.77	0.64	0.70	0.86

Table 7: Score Comparison

The classification report shows that the model is 86% correct for both the cases whether it replaced with '?' mark or not.

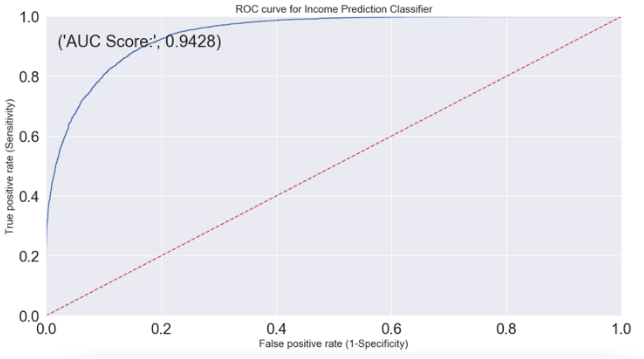


Figure 12: ROC Curve for Gradient Boosting

In **Figure 12**, the AUC score is 0.9428

Algorithm 7: Method – Extreme Gradient Boosting

XGBoost [16] is an alternative form of gradient boosting method. This method generally considers the initial prediction as 0.5 and build the decision tree to predict the residuals. It considers the regularization parameter to avoid overfitting.

	precision	**recall**	**f1_score**	**accuracy**
With '?'	0.82	0.82	0.82	0.88
Replaced '?'	0.75	0.66	0.70	0.86

Table 8: Score Comparison

From above table we can infer that 88% is correct when it is not replaced and 86%, we it is replaced.

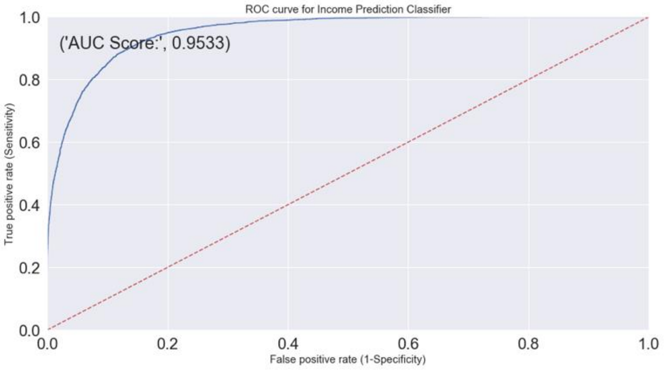


Figure 13: ROC Curve for XGBoost

In **Figure 13**, the AUC score is 0.9533.

Algorithm 8: Method –XGBoost using GridSearchCV

In this section we try to tune the hyperparameters[17] for XGBoost classifiers for more accurate result. Here we get the best parameters for XGBoost classifier are - 'gamma': 0, 'learning_rate': 0.1, 'max_depth': 12

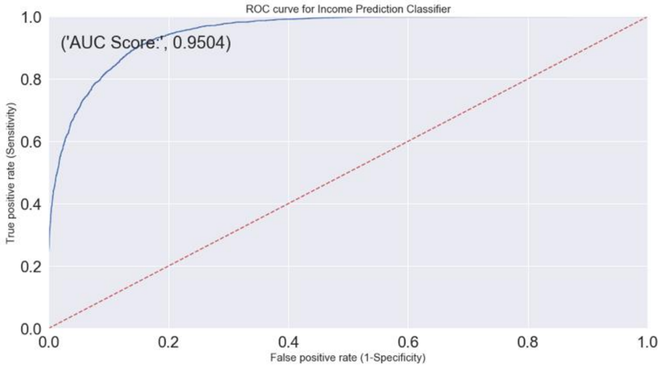


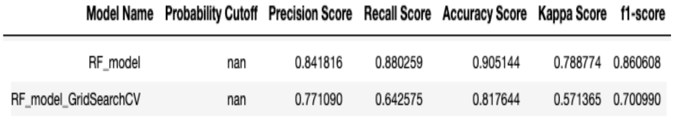
Figure 14: ROC Curve for XGBoost tuned parameters

The AUC score is showing 0.9504 from the above graph.

Algorithm 9: Method –Random Forest using GridSearchCV

Here we tune the hyperparameters for Random Forest using GSCV[18] and get the best parameters - 'criterion': 'gini', 'max_depth': 6, 'max_features': 'sqrt', 'max_leaf_nodes': 11, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 14

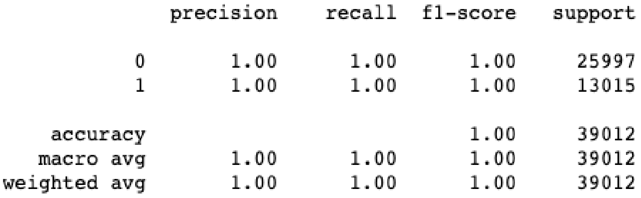
Below table is showing the result for both RF and tuned RF model



Model Name	Probability Cutoff	Precision Score	Recall Score	Accuracy Score	Kappa Score	f1-score
RF_model	nan	0.841816	0.880259	0.905144	0.788774	0.860608
RF_model_GridSearchCV	nan	0.771090	0.642575	0.817644	0.571365	0.700990

Based on the Random Forest train accuracy is 100%, so it is clearly overfitted model on the other hand the test report shows that it is 91% accurate, so again we can say the Random Forest classifier overfitting.

Train Report

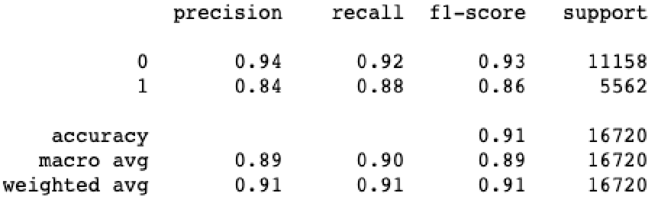


```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     25997
           1       1.00      1.00      1.00     13015

    accuracy                           1.00     39012
   macro avg       1.00      1.00      1.00     39012
weighted avg       1.00      1.00      1.00     39012
```

Test Report

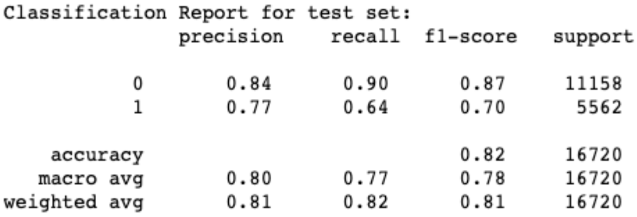


```
              precision    recall  f1-score   support

           0       0.94      0.92      0.93     11158
           1       0.84      0.88      0.86      5562

    accuracy                           0.91     16720
   macro avg       0.89      0.90      0.89     16720
weighted avg       0.91      0.91      0.91     16720
```

With hyperparameter tuning the accuracy sore decreases for both train and test. Now it is 82%. So, this tuning model is not overfitted.



```
Classification Report for test set:
              precision    recall  f1-score   support

           0       0.84      0.90      0.87     11158
           1       0.77      0.64      0.70      5562

    accuracy                           0.82     16720
   macro avg       0.80      0.77      0.78     16720
weighted avg       0.81      0.82      0.81     16720
```

V. MODEL COMPARISON

Basically, in this paper we trying to work on two different methodology, the 1st one is analysis the classification model and comparison the score without replacing the '?' mark and the 2nd approach is analysis based on '?'.

Model	Accuracy with '?'	Accuracy without '?'
Logistic Regression	79%	85%
Recursive Feature Elimination	75%	76%
K Nearest Neighbours	75%	77%
Random Forest	90%	85%
AdaBoost	83%	85%
Gradient Boosting	86%	86%
XGBoost	88%	87%

The accuracy is high in Random Forest when unknown value is not replaced, but in train and test reports there was a huge difference in them, so it is considered that the RF model is an overfitted model.

Coming next highest accuracy is XGBoost model, it shows 88% of accuracy without replacing the unknown value, and with replacing unknown value it gives 87% accuracy, so in both cases XGBoost model performs good.

VI. CONCLUSION

For two different cases, '?' mark were treated as the value and not any error/missing part in the data. Now the '?' mark were treated as the missing value and were replaced with respective methods (Applying frequency count or mode). We saw various classification methods, like LR, RFE, KNN etc. so, these are the different models and hyperparameter tuning models comes into the picture for getting the best model in terms of highest accuracy score, and here for both the cases we have "XGBoost-Hyperparameter tuning" as the best model which gives the accuracy score around 88%.

REFERENCES

[1]. Leo Breiman (1996). "BIAS, VARIANCE, AND ARCING CLASSIFIERS" (PDF). TECHNICAL REPORT. Archived from the primary (PDF) on 2015-01-19. Retrieved nineteen Jan 2015. Arcing [Boosting] is further successful than sacking in variance reduction.

[2]. *Zhou Zhihua, (2008).* "On the margin explanation of boosting algorithm" *(PDF). In: Proceedings of the 21st Annual Conference on Learning Theory (COLT'08): 479–490.* On the margin explanation of boosting algorithm.

[3]. Schapire, Robert E. (1990). "The Strength of Weak Learnability" (PDF). Machine Learning. 5 (2): 197–227. CiteSeerX 10.1.1.20.723. doi:10.1007/bf00116037. S2CID 53304535. Archived from the primary (PDF) on 2012-10-10. Retrieved 2012-08-23..

[4]. Altman, Naomi S. (1992). "An introduction to kernel and nearest-neighbor nonparametric regression" (PDF). *The American Statistician.*

[5]. https://blockgeni.com/recursive-feature-elimination-rfe-in-python/

[6]. Ho TK (1998). "The Random Subspace Method for Constructing Decision Forests" (PDF). *IEEE Transactions on Pattern Analysis and Machine Intelligence*

[7]. https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc#:~:text=An%20ROC%20curve%20(receiver%20operating,False%20Positive%20Rate

[8]. Sofia Visa, Brian Ramsay, Anca Ralescu, Esther van der Knaap "Confusion Matrix-based Feature Selection".

[9]. Wright, R. E. (1995). *Logistic regression.* In L. G. Grimm & P. R. Yarnold (Eds.), *Reading and understanding multivariate statistics* (p. 217–244). American Psychological Association.

[10]. Pablo M.Granitto, Cesare Furlanellob, Franco Biasioli, Flavia Gasperi – "Recursive feature elimination with random forest for PTR-MS analysis"

[11]. Debo Cheng, Shichao Zhang, Zhenyun Deng, Yonghua Zhu, Ming Zong – "kNN Algorithm with Data-Driven k Value"

[12]. Angshuman Paul, Dipti Prasad Mukherjee, Senior Member, IEEE, Prasun Das, Abhinandan Gangopadhyay, Appa Rao Chintha and Saurabh Kundu – "Improved Random Forest for Classification"

[13]. Boosting classifier for predicting protein domain structural class Kai-Yan Feng a , Yu-Dong Cai b , Kuo-Chen Chou c,* a Imaging Science and Biomedical Engineering, Medical School, The University of Manchester, Manchest

[14]. Jingjing Cao, Sam Kwong n , Ran Wang – " A noise-detection based AdaBoost algorithm for mislabeled data"

[15]. P Prettenhofer, G Louppe - 2014 - orbi.uliege.be – "Gradient boosted regression trees in scikit-learn"

[16]. Blagus, R., Lusa, L., Gradient boosting for high-dimensional prediction of rare events. Computational Statistics and Data Analysis (2016), http://dx.doi.org/10.1016/j.csda.2016.07.016

[17]. Sayan Putatunda, Kiran Rama – "A Comparative Analysis of Hyperopt as Against Other Approaches for Hyper-Parameter Optimization of XGBoost"

[18]. Ranjan G S K, Amar Kumar Verma, Sudha Radhika – "K-Nearest Neighbors and Grid Search CV Based Real Time Fault Monitoring System for Industries"