Data Mining and Machine Learning - 1 (MSCDAD_C) Project Report

Name: Sushma Suresh Hebbalakar
Student ID:22224122

Master's in Data Analytics

National College of Ireland

03-01-2024

Machine Learning Models Exploration: A Comparative Analysis for Dry Bean Classification, Candidates Identification, and Customer Churn.

Sushma Suresh Hebbalakar
Master's in Data Analytics
National College of Ireland
22224122

***Abstract*—This project encompasses three distinct datasets, each contributing to diverse domains of analysis. Firstly, we delve into Dry Bean Data Classification, employing machine learning techniques in R to accurately classify various types of dry beans based on key attributes. Next, HR Analysis scrutinizes candidate behavior, utilizing predictive modeling in R to identify subtle indicators of job-seeking tendencies. Lastly, Customer Churn Analysis in a banking context employs R's machine learning capabilities to predict and mitigate customer churn, enhancing retention strategies. By navigating through these datasets, we aim to showcase the versatility of R in handling classification tasks across agricultural, human resources, and financial domains, underscoring its efficacy in diverse analytical scenarios.**

I. Introduction

1. Dry Bean Data Classification: This dataset encompasses diverse attributes of dry beans, and objective here is to employ machine learning techniques in R to classify the beans into distinct categories based on their features. By doing so, we aim to enhance agricultural practices and streamline bean categorization processes.

2. HR Analysis - Candidate Job Seeking Prediction: Focused on the human resources domain, this dataset explores various indicators that hint at a candidate's inclination towards job-seeking activities. Using predictive modeling in R, we aim to uncover subtle patterns and signals, aiding HR professionals in proactive talent management and retention efforts.

3. Customer Churn Analysis in Banking: In the context of the banking industry, this dataset delves into customer behavior to predict and mitigate churn. Leveraging R's machine learning capabilities, our goal is to identify key factors contributing to customer churn, empowering banks to implement targeted strategies for customer retention and satisfaction.

II. Related Work

A. Dataset 1

In [10] by S. Shriya, V. Kumar and P. S. Singh Aydav, they are trying to classify the dry bean using ensemble model which is obtained by combining Random Forest and XG Boost.They have concluded in their paper that Ensemble model is best model to classify the dry bean I have tried to implement booth in my findings as well but didn't use just use just two of the models have explored other models as well.

In [12] by P. Suksomboon and A. Ritthipakdee, they have done performance comparison of KNN and Randomforest on different datasets for classification and have came to conclusion that if the dataset length is small KNN will perform better while if the dataset is large then Random forest will perform better.Since I have used the dataset of 20k instance I didn't find much difference between KNN and Randomforest. They both have performed better comparatively Random forest has performed well.

In [4] by S. Hammad, S. Alhaddad, H. Yusuf and A. Alqaddoumi, Different values for K were used (from 1 to 10), and for the parallel execution, different values were used for processors 2, 3, 4, 5, 6, and 7. It was found that the larger the dataset, the greater the serial time. Meanwhile, the use of 7 processors in parallel execution using the KNN algorithm needs less time. In addition, as the number of processors increases, the speedup increases and the efficiency decreases.

In [9] by G. Shobana, S. N. Bushra, K. U. Maheswari and N. Subramanian, They have done pre-processing and trained various models. They have tried to explore which model performs better for given dataset by dimensional reduction using PCA. They have tried to train almost 6 models and tried to explore which one would perform better.Even I have implemented the PCA for dimensional reduction in my project since the dataset was refined there was very slight difference performance wise there was no huge difference found before and after dimensional reduction.

B. Dataset 2

In [8] by S. Roshini, S. Prakash, J. Shilpha Dharshini, M. N. Saroja and J. Dhivya, have worked with decision tree analysis and K Nearest Neighbour alogorithm, to understand the correlation between between emplyees attrition, overtime, performance, monthly income and monthly rate. Hence, were able to conclude that there was a significant correlation between the variables chosen and any increase in the independent variable, will have an impact on the dependent variable. And this seems to be true in my findings as well.

In [2] by V. Dhole, P. V. Yadav, P. N. Phule, U. S. Kollimath and S. Dharmadhikari they have tried to demonstrate why dong HR analytics is important. HR analytics enables HR professionals to leverage data to make informed decisions, optimize processes, and improve overall productivity. By using analytics tools and methodologies, HR departments can gain valuable insights into their workforce, align HR strategies with organizational goals, and make proactive, data-driven decisions that enhance productivity and contribute to the success of the organization.

In [1] by Arora, Meenal and Prakash, Anshika and Mittal, Amit and Singh, Swati, tthey have tried to explain how AI implementation to do analysis on data would help the HR department in various ways.How I would help in enhancing the Humn Resource Management.

In [11] paper presents a survey of using CI and AI in HR analytics. A conceptual example in the context of construction job searching platform using career guidance and development service for students and workers is illustrated. Finally, type of HR analytics in training the trainers is provided for illustration the use of CI and AI as support engines. Enhancing career development and training for upskilling and reskilling through private-public partnership is a necessary element to establish the gap of the successful implementation.

C. Dataset 3

In [3] by Galal, Mohamed and Rady, Sherine and Aref, Mostafa, they have used KNN (k-Nearest Neighbors) algorithm, Logistic Regression, AdaBoost model, Gradient_Boosting model and Random Forest model.The research used a dataset contains 10,000 records. The model applied KNN, Logistic Regression, Random Forest, AdaBoost and Gradient_ Boosting classifiers under different conditions for this study. A better result is achieved when using the Gradient_Boosting classifier.I have implemented NN (k-Nearest Neighbors) algorithm, Logistic Regression for m findings and have used other models like linear regression, Naive Bayes, SVM and ensamble model based on majority voting.

In [6] by they have explained the system where the customer data of banking sector is used to predict whether the customer is going to leave the bank or not. For that LSTM model with SMOTE data pre-processing was used. In SMOTE technique, synthetic minority samples are generated for minor class of data. Thus it can overcome the issue of unequal distribution of data. and have it is concluded that the LSTM model with SMOTE can perform in a way better than the standard models.he finding are true and even in my dataset I have used sampling to balance the data and tries to train my models.

In [5] by Hemalatha, Putta and Amalanathan, Geetha Mary, They examined KNN, SVM, Decision Tree, RF classifiers under different conditions for this study. A better result was achieved when using the RF classifier together with oversampling. Even in my project Random forest has performed better with oversampling for minor class.

In [7] by S. Murindanyi, B. Wycliff Mugalu, J. Nakatumba-Nabende and G. Marvin, They have examined 8 diffrent types of models for two different datasets and have concluded that Randomforest has performed better in both the datsets and they are suggesting to use random forest to find the solution.As informed earlier even in my finding Randomforest has performed better comparatively.

III. Methodology

Utilizing a systematic approach, methodology involves preprocessing the datasets by cleaning and normalizing the data. Then implement advanced machine learning algorithms in R for classification tasks in dry bean data, candidate job seeking prediction, and customer churn analysis. Cross-validation techniques are employed to assess model performance, ensuring robustness and generalization.

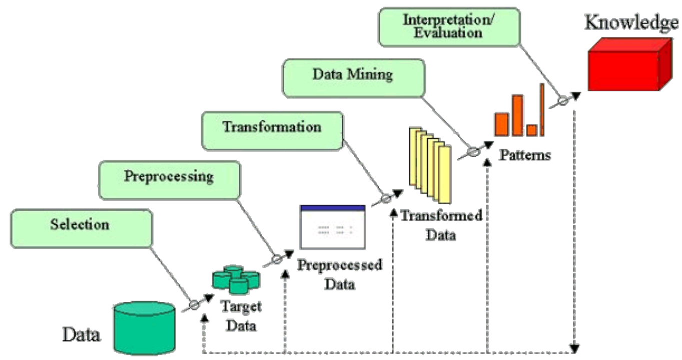

Figure 1: KDD methodology

KDD (Knowledge Discovery in Databases): Embracing the principles of Knowledge Discovery in Databases (KDD), process involves extracting valuable insights from raw datasets. Employ data mining and exploratory data analysis techniques, leveraging R's powerful tools to uncover hidden patterns, trends, and relationships within the datasets. This iterative and comprehensive KDD process is integral to unveiling meaningful knowledge and actionable findings across diverse analytical domains.

A. Data

Three datasets of different domains are from various open source platforms
Dataset 1– Dry Bean Data data from https://archive.ics.uci.edu/ was collected for classifying dry bean data into different classes.
Dataset 2- HR Analytics data fromhttps://www.kaggle.com was collected to check which all candidates are looking for jobs based on various features here.
Dataset 3- Customer churn data fromhttps://www.kaggle.com was collected to looks for the customers who are being churned in bank based on different features.

B. Target

The target variables for all three datsets are as below:
Dataset 1- Class is the target variable and we have Seker, Barbunya, Bombay, Cali, Dermosan, Horoz, and Sira classifiations.
Dataset 2- The target variable has values 0 implying candidate is not looking for job, 1 implying they are looking for job.

Dataset 3- The Target variable has values 0 and 1. O implies the customer is not living the bank and 1 implies customer is living the bank.

C. Pre-Processing Data

Based on requirements there are different kinds of processing steps taken. Like checking if there are null values or empty values are present, if present they have been handled by mutating by different methods based on requirement and findings.
Dataset 1- There were no null values as shown in Fig 2 or empty string present as such and most of the features were numeric type.

```
print(dry_bean_null_count)
           Area       Perimeter MajorAxisLength MinorAxisLength    AspectRation
              0               0               0               0               0
   Eccentricity      ConvexArea   EquivDiameter          Extent        Solidity
              0               0               0               0               0
      roundness     Compactness    ShapeFactor1    ShapeFactor2    ShapeFactor3
              0               0               0               0               0
   ShapeFactor4           Class
              0               0
```

Figure 2: Dry_Bean Data Null count

Dataset 2- There were empty strings, null values present in the dataset as shown in figure 3. There were few columns like 'education_level', 'experience', 'company_size', 'last_new_job' which required order level encoding hence the same has been performed.

```
> print(HR_analytics_null_count)
           enrollee_id                   city city_development_index
                     0                      0                      0
                gender    relevent_experience    enrolled_university
                  4508                      0                    386
       education_level       major_discipline             experience
                   460                   2813                     65
          company_size           company_type           last_new_job
                  5938                   6140                    423
        training_hours                 target
                     0                      0
```

Figure 3: HR Analytics Data Null count

And for column like 'gender' and few more were label encode. Later the null values were filled with mode value based on understanding of the data as all of them were categorical data haven't mutated null values with mean or median. The Numeric columns didn't contain any null values as such.
Dataset 3- The data set had no empty or null values present in it except for the row number column as shown in Figure 4.

```
> print(customer_data_null_count)
      RowNumber      CustomerId         Surname     CreditScore       Geography
              1               0               0               0               0
         Gender             Age          Tenure         Balance   NumOfProducts
              0               0               0               0               0
      HasCrCard  IsActiveMember EstimatedSalary          Exited
              0               0               0               0
```

Figure 4: HR Analytics Data Null count

D. Transformation

Again based on the requirements, findings and analysis different kind of transformation were performed on the datasets.
Dataset 1- Since most of all the 14 features in dataset were numeric as shown in Figure 5, initially the statistics outputs were checked on each of the features, like their min value, max value, mean medians etc., as shown in Figure 6. Since the ranges were different in all of them scaling was performed. And standard scaling is found to be good for the models of choice I have chosen standard scaling to transform data.

```
> str(dry_bean_data)
'data.frame':	13611 obs. of  17 variables:
 $ Area           : num  28395 28734 29380 30008 30140 ...
 $ Perimeter      : num  610 638 624 646 620 ...
 $ MajorAxisLength: num  208 201 213 211 202 ...
 $ MinorAxisLength: num  174 183 176 183 190 ...
 $ AspectRation   : num  1.2 1.1 1.21 1.15 1.06 ...
 $ Eccentricity   : num  0.55 0.412 0.563 0.499 0.334 ...
 $ ConvexArea     : num  28715 29172 29690 30724 30417 ...
 $ EquivDiameter  : num  190 191 193 195 196 ...
 $ Extent         : num  0.764 0.784 0.778 0.783 0.773 ...
 $ Solidity       : num  0.989 0.985 0.99 0.977 0.991 ...
 $ roundness      : num  0.958 0.887 0.948 0.904 0.985 ...
 $ Compactness    : num  0.913 0.954 0.909 0.928 0.971 ...
 $ ShapeFactor1   : num  0.00733 0.00698 0.00724 0.00702 0.0067 ...
 $ ShapeFactor2   : num  0.00315 0.00356 0.00305 0.00321 0.00366 ...
 $ ShapeFactor3   : num  0.834 0.91 0.826 0.862 0.942 ...
 $ ShapeFactor4   : num  0.999 0.998 0.999 0.994 0.999 ...
 $ Class          : chr  "SEKER" "SEKER" "SEKER" "SEKER" ...
```

Figure 5: Structure of Dry Bean Data

```
> summary(dry_bean_data)
      Area          Perimeter      MajorAxisLength MinorAxisLength  AspectRation
 Min.   : 20420   Min.   : 524.7   Min.   :183.6   Min.   :122.5   Min.   :1.025
 1st Qu.: 36328   1st Qu.: 703.5   1st Qu.:253.3   1st Qu.:175.8   1st Qu.:1.432
 Median : 44652   Median : 794.9   Median :296.9   Median :192.4   Median :1.551
 Mean   : 53048   Mean   : 855.3   Mean   :320.1   Mean   :202.3   Mean   :1.583
 3rd Qu.: 61332   3rd Qu.: 977.2   3rd Qu.:376.5   3rd Qu.:217.0   3rd Qu.:1.707
 Max.   :254616   Max.   :1985.4   Max.   :738.9   Max.   :460.2   Max.   :2.430
  Eccentricity      ConvexArea     EquivDiameter       Extent
 Min.   :0.2190   Min.   : 20684   Min.   :161.2   Min.   :0.5553
 1st Qu.:0.7159   1st Qu.: 36715   1st Qu.:215.1   1st Qu.:0.7186
 Median :0.7644   Median : 45178   Median :238.4   Median :0.7599
 Mean   :0.7509   Mean   : 53768   Mean   :253.1   Mean   :0.7497
 3rd Qu.:0.8105   3rd Qu.: 62294   3rd Qu.:279.4   3rd Qu.:0.7869
 Max.   :0.9114   Max.   :263261   Max.   :569.4   Max.   :0.8662
    Solidity        roundness       Compactness      ShapeFactor1
 Min.   :0.9192   Min.   :0.4896   Min.   :0.6406   Min.   :0.002778
 1st Qu.:0.9857   1st Qu.:0.8321   1st Qu.:0.7625   1st Qu.:0.005900
 Median :0.9883   Median :0.8832   Median :0.8013   Median :0.006645
 Mean   :0.9871   Mean   :0.8733   Mean   :0.7999   Mean   :0.006564
 3rd Qu.:0.9900   3rd Qu.:0.9169   3rd Qu.:0.8343   3rd Qu.:0.007271
 Max.   :0.9947   Max.   :0.9907   Max.   :0.9873   Max.   :0.010451
  ShapeFactor2        ShapeFactor3     ShapeFactor4       Class
 Min.   :0.0005642   Min.   :0.4103   Min.   :0.9477   Length:13611
 1st Qu.:0.0011535   1st Qu.:0.5814   1st Qu.:0.9937   Class :character
 Median :0.0016935   Median :0.6420   Median :0.9964   Mode  :character
 Mean   :0.0017159   Mean   :0.6436   Mean   :0.9951
 3rd Qu.:0.0021703   3rd Qu.:0.6960   3rd Qu.:0.9979
 Max.   :0.0036650   Max.   :0.9748   Max.   :0.9997
```

Figure 6: Dry Bean Data Summary

Dataset 2- Since the independent variables here were character type as shown in Figure 7. First I have used chi-square to eliminate features by identifying those that are most likely to be less informative for predicting the target variable.

```
> str(HR_analytics)
'data.frame':	19158 obs. of  14 variables:
 $ enrollee_id           : int  8949 29725 11561 33241 666 21651 28806 402 27107 699
...
 $ city                  : chr  "city_103" "city_40" "city_21" "city_115" ...
 $ city_development_index: num  0.92 0.776 0.624 0.789 0.767 0.764 0.92 0.762 0.92 0.92
...
 $ gender                : chr  "Male" "Male" NA NA ...
 $ relevent_experience   : chr  "Has relevent experience" "No relevent experience" "No
relevent experience" "No relevent experience" ...
 $ enrolled_university   : chr  "no_enrollment" "no_enrollment" "Full time course" NA
...
 $ education_level       : chr  "Graduate" "Graduate" "Graduate" "Graduate" ...
 $ major_discipline      : chr  "STEM" "STEM" "STEM" "Business Degree" ...
 $ experience            : chr  ">20" "15" "5" "<1" ...
 $ company_size          : chr  NA "50-99" NA NA ...
 $ company_type          : chr  NA "Pvt Ltd" NA "Pvt Ltd" ...
 $ last_new_job          : chr  "1" ">4" "never" "never" ...
 $ training_hours        : int  36 47 83 52 8 24 24 18 46 123 ...
 $ target                : num  1 0 0 1 0 1 0 1 1 0 ...
```

Figure 7: Structure of HR Analytics Data

Dataset 3- Here for categorical values, again chi-square was used to eliminate less informative feature/s and for few of categorical feature like 'gender', numeric label of 0 was given to female and 1 for male. For 'geography' the mappings for "Germany", "Spain", "France" was done to 1, 0, 3 respectively. And Standard Scaling was done on "CreditScore", "Age", "Balance" and "EstimatedSalary".And the structure for HR Analytics was as shown in Figure 8.

```
> str(customer_data)
'data.frame':	10000 obs. of  14 variables:
 $ RowNumber      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ CustomerId     : int  15634602 15647311 15619304 15701354 15737888 15574012 15592531 
15656148 15792365 15592389 ...
 $ Surname        : chr  "Hargrave" "Hill" "Onio" "Boni" ...
 $ CreditScore    : int  619 608 502 699 850 645 822 376 501 684 ...
 $ Geography      : chr  "France" "Spain" "France" "France" ...
 $ Gender         : chr  "Female" "Female" "Female" "Female" ...
 $ Age            : int  42 41 42 39 43 44 50 29 44 27 ...
 $ Tenure         : int  2 1 8 1 2 8 7 4 4 2 ...
 $ Balance        : num  0 83808 159661 0 125511 ...
 $ NumOfProducts  : int  1 1 3 2 1 2 2 4 2 1 ...
 $ HasCrCard      : int  1 0 1 0 1 1 1 1 0 1 ...
 $ IsActiveMember : int  1 1 0 0 1 0 1 0 1 1 ...
 $ EstimatedSalary: num  101349 112543 113932 93827 79084 ...
 $ Exited         : int  1 0 1 0 0 1 0 1 0 0 ...
```

Figure 8: Structure of Customer Churn Data

E. Data Mining

In the Knowledge Discovery in Databases (KDD) process, data mining refers to the application of various algorithms and techniques to the transformed data in order to extract valuable patterns, trends, relationships, or knowledge that may be hidden within the dataset. After the data pre-processing and transformation steps, which involve cleaning, normalization, and handling missing values, data mining is the phase where the actual analysis occurs.
The distribution/counts in dataset 1 for all features is as shown in Figure 9:

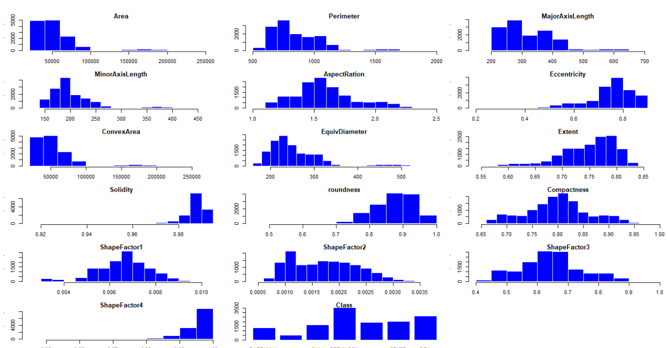

Figure 9: Data distribution for Dry Bean Data

The distribution/counts in dataset 2 for all features is as shown in Figure 10:

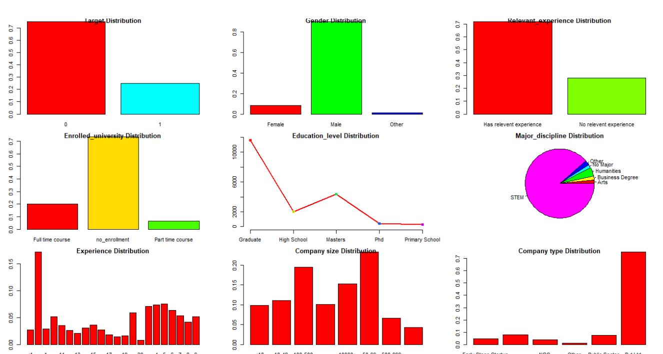

Figure 10: Data distribution for HR analytics

The distribution/counts for dataset 3 for all features is as shown in Figure 11:

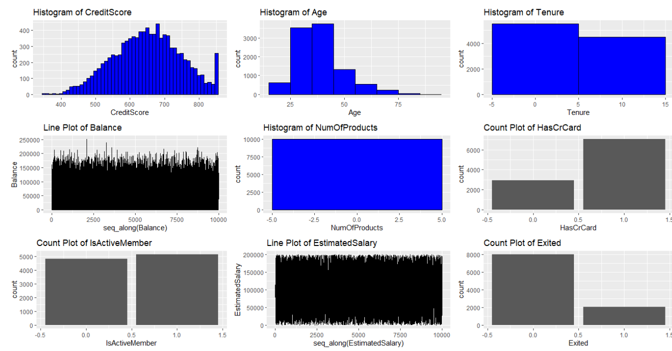

Figure 11: Data distribution for Customer Churn

Figure 12-16 shows the count plot and bar plot for churn rate with respect to different features for Customer Churn for categorical columns. It is evident that except for customer has credit card card or not column, other features has shown some difference in churn rate values. While with respect to Credit card the difference in churn rate value seem to be negligible.

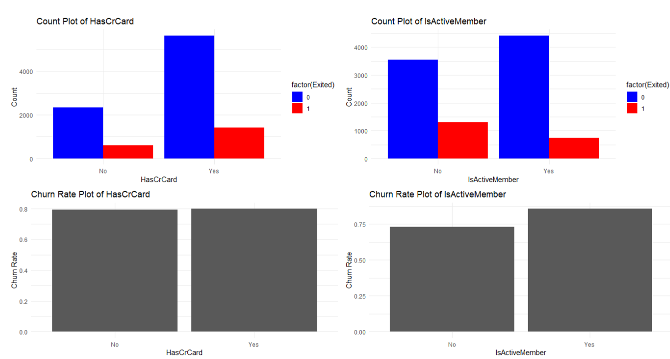

Figure 12: Customer churn and Churn Rate

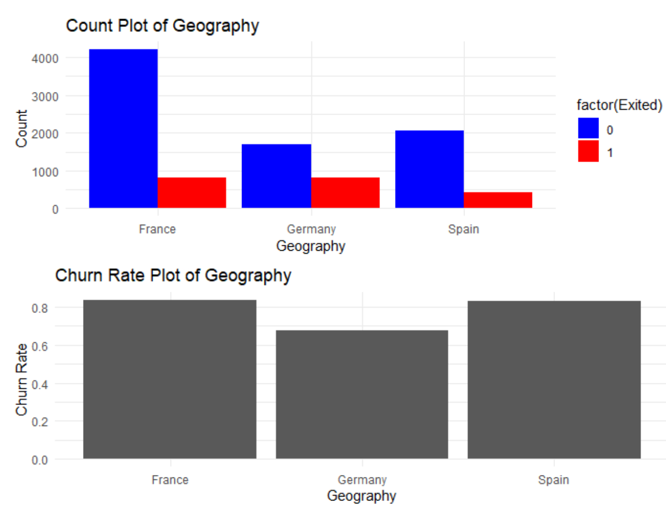

Figure 13: Customer churn and Churn Rate

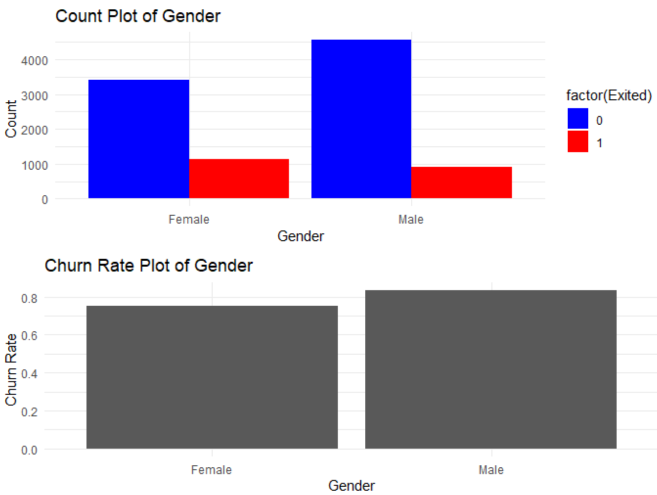

Figure 14: Customer churn and Churn Rate

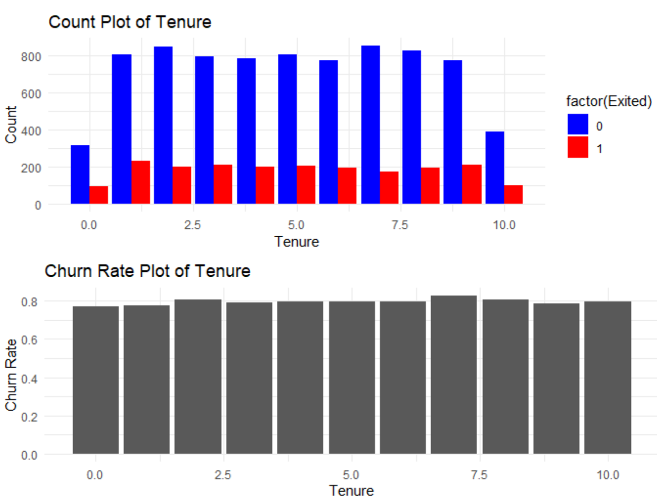

Figure 15: Customer churn and Churn Rate

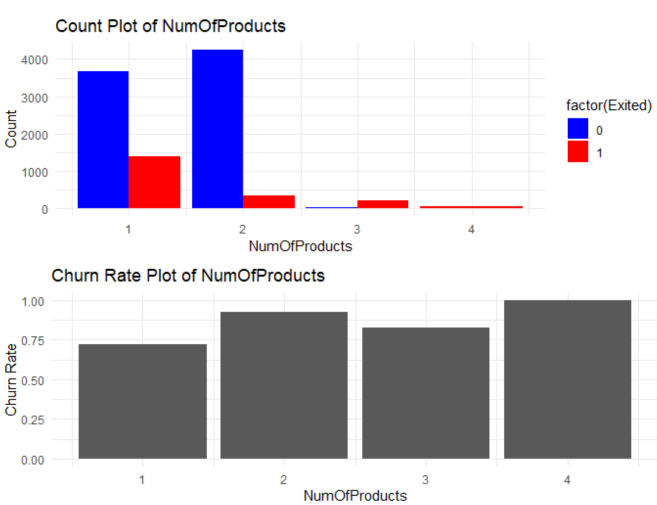

Figure 16: Customer churn and Churn Rate

F. Interpretation

After performing pre-processing and understanding the patterns different models were used to train and evaluate the data which will be explained in details in Evaluation section.

G. Knowledge

The knowledge gained after doing evaluation, which model is performing better based on what and how, Overall accuracy gained, Which is suitable model to get the desired output will be explained in Results section further.

IV. Evaluation

A. Dataset 1

In this Dataset as part of pre-processing first I have tried to remove outliers using DBScan. DBScan is a versatile outlier detection method that relies on density-based principles to automatically identify and separate outliers from the main clusters in classification data. Its ability to handle noise, adapt to varying data densities, and work well in high-dimensional spaces makes it a valuable tool for data pre-processing in classification tasks. Since target variable has 7 classification I have used this method to remove outliers in data.

Then I have done scaling for the numeric columns using Standard Scalar.And for dimensional reduction.I have used PCA for selecting dimensions. Below figure showed first 8 features have more variance in data, as shown in Figure 17 and 18, that can help in classifying the target variables.

```
> summary(pca_result)
Importance of components:
                          PC1    PC2     PC3     PC4     PC5     PC6
Standard deviation     3.0103 2.0447 1.05613 0.90918 0.67503 0.43463
Proportion of Variance 0.5664 0.2613 0.06971 0.05166 0.02848 0.01181
Cumulative Proportion  0.5664 0.8277 0.89739 0.94906 0.97754 0.98934
                           PC7     PC8     PC9    PC10    PC11    PC12
Standard deviation     0.33446 0.22109 0.08509 0.03566 0.03097 0.01401
Proportion of Variance 0.00699 0.00306 0.00045 0.00008 0.00006 0.00001
Cumulative Proportion  0.99633 0.99939 0.99984 0.99992 0.99998 0.99999
                          PC13     PC14     PC15    PC16
Standard deviation     0.01059 0.001853 0.001281 0.00116
Proportion of Variance 0.00001 0.000000 0.000000 0.00000
Cumulative Proportion  1.00000 1.000000 1.000000 1.00000
```

Figure 17: PCA result on dry bean data

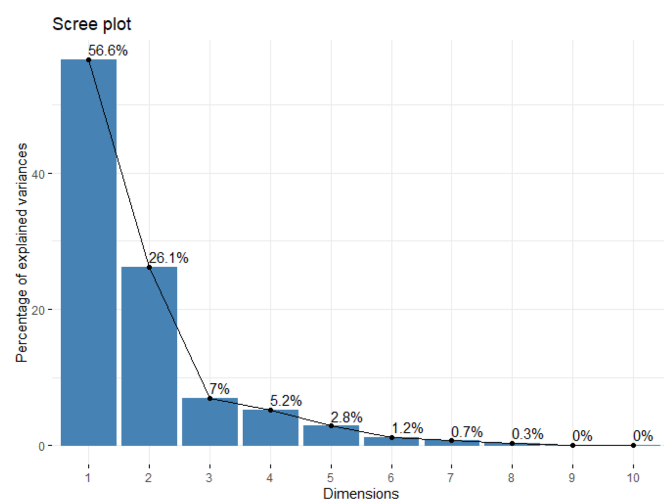

Figure 18: Graphical representation on proportion of variance

After PCA all correlation values between the principal components and original numeric columns are exactly 0 as shown in Figure 19, it signifies the successful orthogonal transformation achieved by PCA, with each principal component representing an independent source of variation in the data.

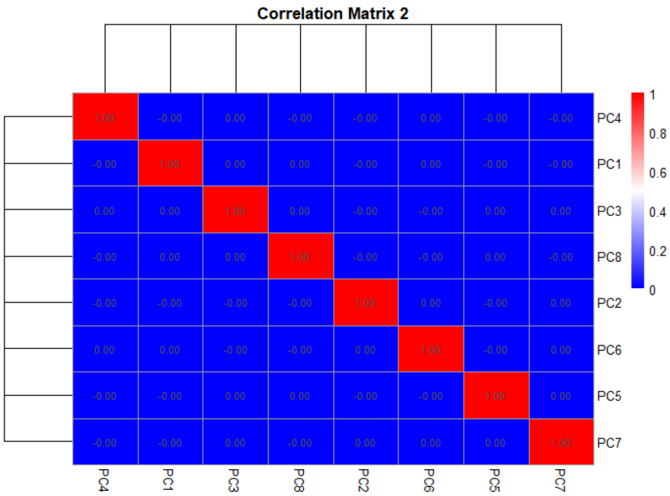

Figure 19: Correlation matrix after dimension reduction

Below are the results of different models on the dataset.

Models/Accuracy	Without PCA	With PCA
Random Forest	92.8	92.73
KNN	91.15	91.25
Naïve Bayes	89.88	89.71
XGBoost	89.88	89.71
Ensemble	91.87	

Figure 20: Comparision of Accuracy before and after PCA

Below are the results on accuracy for each class on different Models.

Class	Random Forest	k-Nearest Neighbors	Naive Bayes	XGBoost
SEKER	0.954	0.917	0.934	0.934
BARBUNYA	1.000	1.000	1.000	1.000
BOMBAY	0.941	0.929	0.917	0.917
CALI	0.918	0.906	0.933	0.933
HOROZ	0.957	0.924	0.925	0.925
SIRA	0.965	0.958	0.966	0.966
DERMASON	0.871	0.853	0.778	0.778

Figure 21: Class-wise accuracy before PCA

Class	Random Forest	k-Nearest Neighbors	Naive Bayes	Ensemble	XGBoost
SEKER	0.979	0.949	0.951	0.952	0.951
BARBUNYA	1.000	1.000	1.000	1.000	1.000
BOMBAY	0.938	0.933	0.913	0.920	0.913
CALI	0.912	0.903	0.929	0.918	0.929
HOROZ	0.973	0.955	0.952	0.954	0.952
SIRA	0.955	0.946	0.960	0.966	0.960
DERMASON	0.858	0.832	0.749	0.836	0.749

Figure 22: Class-wise accuracy after PCA

Below are some other interpretations:
Random Forest: Confidence Interval (CI): (0.9187, 0.9354) This is a relatively narrow interval, suggesting a high degree of precision in estimating the true accuracy. Kappa: 0.9114 A Kappa value of 0.9114 indicates a very high level of agreement beyond what would be expected by chance.

XGBoost: Confidence Interval (CI): (0.8871, 0.9066) This is a reasonably narrow interval, indicating a good level of precision in estimating the true accuracy. Kappa: 0.8749 A Kappa value of 0.8749 suggests a high degree of agreement beyond what would be expected by chance.

kNearest Neighbors (KNN): Confidence Interval (CI): (0.9031, 0.9213) This is a relatively narrow interval, suggesting a high level of precision in estimating the true accuracy. Kappa: 0.8933 A Kappa value of 0.8933 indicates a very high level of agreement beyond what would be expected by chance.

Naive Bayes: Confidence Interval (CI): (0.8847, 0.9042) This is a reasonably narrow interval, indicating a good level of precision in estimating the true accuracy. Kappa: 0.8717 A Kappa value of 0.8717 suggests a high degree of agreement beyond what would be expected by chance.

B. Dataset 2:

In Dataset 2, as preprocessing frst I have corrected the company size column which had wrong format from'10/49' to '10-49'. And then replaced or mutated the null value column with Mode values for character columns.To reduce the dimensions I have used chi-square method for character columns. Below were the results as shown in Figure 23:

```
> print(result)
              variable  chi_square       p_value
1                 city 2998.777229  0.000000e+00
7           experience  690.983270 1.066061e-132
8         company_size  592.964197 7.971606e-124
4  enrolled_university  440.458589  2.267945e-96
3  relevent_experience  315.338577  1.500663e-70
5      education_level  160.454092  1.168254e-33
10        last_new_job  140.620659  1.320494e-28
9         company_type   91.187945  3.782007e-18
6     major_discipline    8.683774  1.223618e-01
2               gender    1.567228  4.567523e-01
```

Figure 23: Chi-square value for character columns in HR Data

While features with higher chi-square values and lower p-values are often considered more relevant. Hence I am dropping "company_type", "major_discipline", "gender" which have low chi-square values. Also I am dropping enrollee_id as it is not an important factor based on general knowledge. Then I have done label encoding wherever necessary either ordinal or manual which has been explained in Pre-processing steps. Target variable count was as shown below in Figure 24:

```
> print(lfjob_counts)

    0     1
10061  3415
```

Figure 24: Imbalanced target columns in HR Data

Using ovun.sample I have tried to over-sample the minority class and below is the target class cunt after sampling.

```
> print(new_lfjob_counts)

    0     1
10061 10061
```

Figure 25: Balanced target columns in HR Data

I have used logistic Regression, Decision Tree and Random Forest to predict the output and below are the observations for balanced dataset:
Figure 26 shows the confusion matrix for logistic Regression, Decision Tee, Random Forest:

Training models for balanced data and their output								
predict_reg			predictions_dt			predictions_rf		
	0	1		0	1	Prediction	0	1
0	3291	1029	0	2534	1786	0	3181	414
1	590	772	1	290	1072	1	1139	948

Figure 26: Confusion Matrix for Balanced data

Below are the observations for unbalanced dataset:
Figure 30 shows the confusion matrix for logistic Regression, Decision Tee, Random Forest:

Training models for unbalanced data and their output								
predict_naive			predictions_lr			predictions_rf		
	0	1		0	1	Prediction	0	1
0	4024	296	0	3893	427	0	3937	848
1	964	398	1	801	561	1	383	514

Figure 27: Confusion Matrix for Unbalanced data

Below are the ROC curves obtained for different models after training them with balanced and unbalanced data.

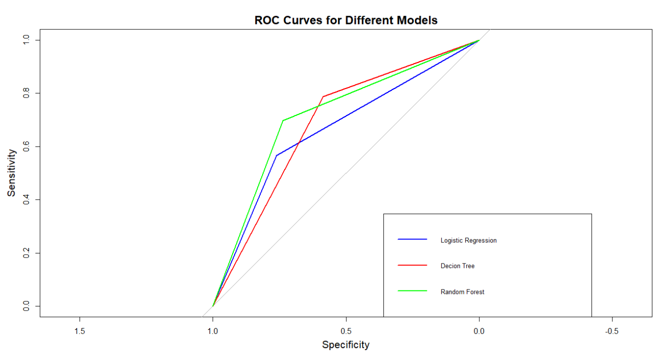

Figure 28: ROC curve for Balanced data

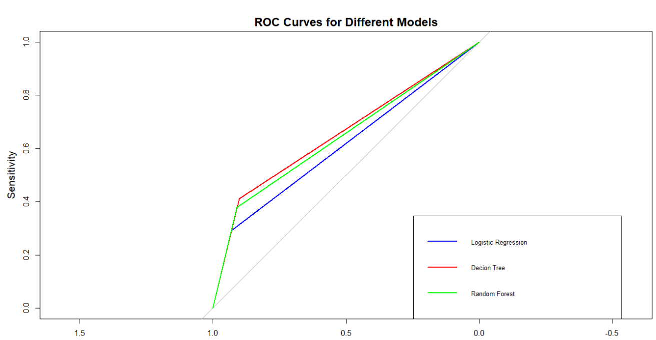

Figure 29: ROC curve for Unbalanced data

Below Figures shows the AUC values for unbalanced and balanced data:AUC can be interpreted as the probability that the model will rank a randomly chosen positive instance higher than a randomly chosen negative instance. For example, an AUC of 0.8 means that if you randomly select a positive instance and a negative instance, the model will correctly rank the positive instance higher than the negative instance 80% of the time.

```
> cat("AUC for Logistic Regression:", auc_lr, "\n")
AUC for Logistic Regression: 0.6643095
> cat("AUC for Random Forest:", auc_rf, "\n")
AUC for Random Forest: 0.7161889
> cat("AUC for Descision Tree:", auc_dt, "\n")
AUC for Descision Tree: 0.686826
```

Figure 30: AUC values for Balanced data

AUC represents the area under the ROC curve. It quantifies the ability of a model to distinguish between positive and negative classes.AUC ranges from 0 to 1, where 0 indicates poor performance (model predicts all negatives as positives or vice versa), and 1 indicates perfect performance (model makes a perfect distinction between positives and negatives). A random classifier would have an AUC of 0.5, as its ROC curve would be a diagonal line.

```
> cat("AUC for Logistic Regression:", auc_lr, "\n")
AUC for Logistic Regression: 0.6118494
> cat("AUC for Random Forest:", auc_rf, "\n")
AUC for Random Forest: 0.6443644
> cat("AUC for Descision Tree:", auc_dt, "\n")
AUC for Descision Tree: 0.6565258
```

Figure 31: AUC values for UnBalanced data

In our case, though the models should higher accuracy for unbalanced data the AUC value were good for those models which were trained on balanced dataset.

Below figre 32 shows the accracies of different models om balanced train data and unbalanced train data.

Models	Balanced Accuracy	Unbalanced Accuracy
Linear_Regression	71.5	77.8
Decision Tree	71.5	78.34
Random_Forest	72.67	77.8

Figure 32: Accuracy of models

Cross-validation is a technique used in machine learning to assess the performance of a model. Instead of relying on a single train-test split, cross-validation involves dividing the dataset into multiple subsets. The model is trained on some of these subsets (folds) and tested on the remaining ones. This process is repeated multiple times, and the performance metrics are averaged. Cross-validation provides a more robust evaluation, helping to ensure that the model's performance is consistent across different parts of the dataset, reducing the risk of overfitting or underfitting. Figure 33 shows the cross-validation report for Random forest.

```
> print(rf_model)
Random Forest

20122 samples
    9 predictor
    2 classes: '0', '1'

Pre-processing: centered (9), scaled (9)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 18110, 18110, 18110, 18110, 18108, 18110, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
  2     0.7518151  0.5036302
  5     0.8841569  0.7683139
  9     0.8889276  0.7778551

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 9.
```

Figure 33: CV report for Random forest

Figure 34 shows the MSE-MEA and F-1 score of all models.

Unbalanced Data:

Evaluation Metric	Linear Regression	Decision Tree	Random Forest
MSE	1.45	1.34	1.38
MAE	1.11	1.06	1.08
F1-score	0.8646	0.8637	0.8647

Balanced Data:

Evaluation Metric	Linear Regression	Decision Tree	Random Forest
MSE	1.13	0.23	1.01
MAE	0.92	0.23	0.86
F1-score	0.8025		0.8028

Figure 34: MSE,MEA,F1-score

C. Dataset 3:

As discussed earlier in Pre-processing section chi-square was calculated for character coluns and below Figure 35 shows the result of chi-square for character columns.

```
> print(result)
        variable   chi_square      p_value
6  NumOfProducts 1503.6293615 0.000000e+00
3      Geography  301.2553368 3.830318e-66
2 IsActiveMember  242.9853416 8.785858e-55
4         Gender  112.9185706 2.248210e-26
5         Tenure   13.9003726 1.775846e-01
1      HasCrCard    0.4713378 4.923724e-01
```

Figure 35: Chi-square for character columns

"Tenure", "HasCrCard" columns which has less ch-square values have be dropped and rest of pre-processing steps carried as discussed earlier.

Below are the evaluation metrics obtained from different models for training them on balanced train_data.

Training models for balanced data and their output														
predict_naive			predictions_lr			predictions_svm			predictions_rf			Ensemble		
	0	1		0	1		0	1		0	1		0	1
0	1825	180	0	1714	195	0	1905	156	0	2086	211	0	2056	217
1	565	430	1	676	415	1	485	454	1	304	399	1	334	393

Figure 36: Confusion matrix by balanced train_data

Training models for unbalanced data and their output														
predict_naive			predictions_lr			predictions_svm			predictions_rf			Ensemble		
	0	1		0	1		0	1		0	1		0	1
0	2328	413	0	2305	471	0	2350	391	0	2329	349	0	2369	433
1	62	197	1	85	139	1	40	219	1	61	261	1	21	177

Figure 37: Confusion matrix by unbalanced train_data

Also the ROC curves fr unbalanced and balaced data is as shown below:

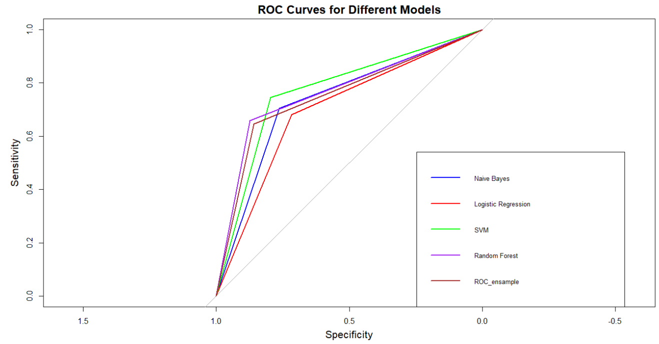

Figure 38: ROC for models that were trained on balanced train_data

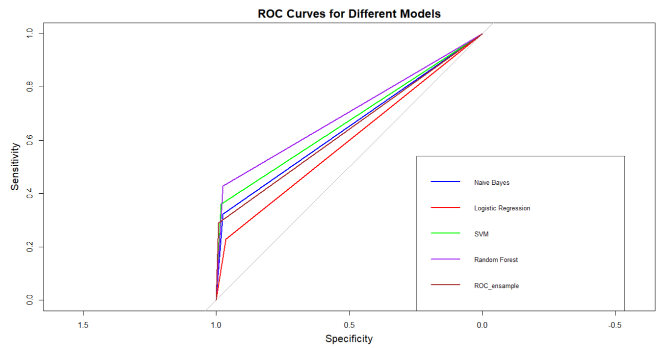

Figure 39: ROC for models that were trained on unbalanced train_data

Even in this dataset the accuracy of models is seen better for unbalanced dataset but the AUC values are good with respect to those models that were trained on balanced data. Below figure shows the AUC values for models trained for balanced and unbalanced train_data.

```
> cat("AUC for Naive Bayes:", auc_nb, "\n")
AUC for Naive Bayes: 0.7342582
> cat("AUC for Logistic Regression:", auc_lr, "\n")
AUC for Logistic Regression: 0.6987413
> cat("AUC for SVM:", auc_svm, "\n")
AUC for SVM: 0.7706667
> cat("AUC for Random Forest:", auc_rf, "\n")
AUC for Random Forest: 0.7634509
> cat("AUC for Ensample:", auc_ensample, "\n")
AUC for Ensample: 0.7522567
```

Figure 40: AUC for models that were trained on balanced train_data

```
> cat("AUC for Naive Bayes:", auc_nb, "\n")
AUC for Naive Bayes: 0.6485047
> cat("AUC for Logistic Regression:", auc_lr, "\n")
AUC for Logistic Regression: 0.596152
> cat("AUC for SVM:", auc_svm, "\n")
AUC for SVM: 0.67114
> cat("AUC for Random Forest:", auc_rf, "\n")
AUC for Random Forest: 0.7011729
> cat("AUC for Ensample:", auc_ensample, "\n")
AUC for Ensample: 0.6406887
```

Figure 41: AUC for models that were trained on unbalanced train_data

Below Figure 39 depicts the accuracy of different models on balanced and unbalanced train_data

Models	Balanced Accuracy	Unbalanced Accuracy
Logestic Regression	70.97	81.4
SVM	78.63	85.6
Random Forest	82.93	86.3
Ensemble	81.6	84.8
Naïve Bayes	75.17	84.17

Figure 42: Accuracy of different models

Below figure shows the Cross-validation result on Random forest for dry bean dataset with 10 folds.

```
> print(rf_model)
Random Forest

11146 samples
    8 predictor
    2 classes: '0', '1'

Pre-processing: centered (9), scaled (9)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 10031, 10032, 10031, 10032, 10032, 10032, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
  2     0.8853402  0.7706809
  5     0.9499371  0.8998742
  9     0.9439257  0.8878517

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 5.
```

Figure 43: Dry_Bean Dataset Cross Validation

Below two figures shows the MSE,MAE and F1-scores.

Unbalanced Data:

Model	MSE	MAE	F1-score
Naïve Bayes	0.24	0.24	0.5358
Linear Regression	0.29	0.29	0.48
SVM	0.21	0.21	0.5861
Random Forest	0.16	0.16	0.61
Ensemble	0.18	0.18	0.5904

Figure 44: MSE, MAE, F1-score

Balanced Data:

Model	MSE	MAE	F1-score
Naïve Bayes	0.15	0.15	0.45
Linear Regression	0.18	0.18	0.33
SVM	0.14	0.14	0.5
Random Forest	0.13	0.13	0.56
Ensemble	0.15	0.14	0.43

Figure 45: MSE, MAE, F1-score

Below figure shows the Cross-validation result on Random forest for dry bean dataset with 10 folds.

```
> print(rf_model)
Random Forest

9069 samples
   8 predictor
   7 classes: 'BARBUNYA', 'BOMBAY', 'CALI', 'DERMASON', 'HOROZ', 'SEKER', 'SIRA'

Pre-processing: centered (8), scaled (8)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 8163, 8162, 8160, 8162, 8163, 8164, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
  2     0.9282165  0.9126339
  5     0.9251281  0.9088789
  8     0.9243584  0.9079333
```

Figure 46: Dry_Bean Dataset Cross Validation

Below two figures shows the MSE,MAE and F1-scores for Dry_Bean dataset with PCA and without PCA application.

Model	MSE	MAE	F1-score
Naïve Bayes	0.87	0.27	1
KNN	0.69	0.23	1
Random Forest	0.53	0.18	1
XGBoost	0.87	0.27	1
Ensemble	0.15	0.14	1

Figure 47: MSE, MAE, F1-score with PCA application

Model	MSE	MAE	F1-score
Naïve Bayes	0.91	0.28	1
KNN	0.68	0.22	1
Random Forest	0.56	0.18	1
XGBoost	0.91	0.28	1
Ensemble	0.702	0.22	1

Figure 48: MSE, MAE, F1-score without PCA application

V. Conclusion and Future work

A. Dataset 1:

In summary, all models have relatively narrow confidence intervals, indicating precise estimates of true accuracy. Additionally, Kappa values for all models are close to 1, signifying a high level of agreement beyond what would be expected by chance. These results suggest that the models are performing well in their respective classifications.And Random forest is performing better amongst all of them. Even ensemble model gave almost similar accuracy and has performed

B. Dataset 2:

With respect to dataset set 2 as explained the accuracies of models are well seen in unbalanced data whereas AUC values are better seen with balanced dataset in both scenarios of training models on balanced and unbalanced dataset Random forest has performed better.The reduction in accuracy at balanced data may be because of oversampling of minority class that has been picked randomly.

C. Dataset 3:

Even with respect to dataset 3 the accuracies is better for the models that are trained on unbalanced train_data whereas with respect to balanced_train data models have better AUC values. Even here the reduction in accuracy for models which are trained under balanced data may be because of oversampling of minority class which were picked randomly.

Overall in all 3 datsets Randomforest has performed better and given better results followed by ensemble model which is based on majority voting of all the models we have trained to predict the target.

As Future work with respect all sets we can try to find more balanced datasets or try to get more accurate datset. Due to restrictions in the R version that was present I was unable to perform oversampling or resampling on dataset 1 which had more that 2 classification data.As R was showing me the warning and errors when I tried to resample them. But I have tried to explore different ways of dimensionality reduction as per my knowledge going ahead we can look for a way to resample the dataset with multiple i.e., more than two class in R 4.3.2.

REFERENCES

[1] Meenal Arora, Anshika Prakash, Amit Mittal, and Swati Singh. Hr analytics and artificial intelligence-transforming human resource management. In *2021 International Conference on Decision Aid Sciences and Application (DASA)*, pages 288–293, 2021.

[2] Vijay Dhole, Pravin Vitthal Yadav, Prashant Namdev Phule, Umesh S. Kollimath, and Sanjay Dharmadhikari. Impact of hr analytics on the productivity of the hr department. In *2023 3rd International Conference on Emerging Smart Technologies and Applications (eSmarTA)*, pages 1–7, 2023.

[3] Mohamed Galal, Sherine Rady, and Mostafa Aref. Enhancing customer churn prediction in digital banking using ensemble modeling. In *2022 4th Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, pages 21–25, 2022.

[4] Salman Hammad, Salman Alhaddad, Husain Yusuf, and Abdulla Alqaddoumi. Parallel implementation of knn algorithm for dry beans dataset. In *2022 International Conference on Data Analytics for Business and Industry (ICDABI)*, pages 72–76, 2022.

[5] Putta Hemalatha and Geetha Mary Amalanathan. A hybrid classification approach for customer churn prediction using supervised learning methods: Banking sector. In *2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*, pages 1–6, 2019.

[6] Jesmi Latheef and S Vineetha. Lstm model to predict customer churn in banking sector with smote data preprocessing. In *2021 2nd International Conference on Advances in Computing, Communication, Embedded and Secure Systems (ACCESS)*, pages 86–90, 2021.

[7] Sudi Murindanyi, Ben Wycliff Mugalu, Joyce Nakatumba-Nabende, and Ggaliwango Marvin. Interpretable machine learning for predicting customer churn in retail banking. In *2023 7th International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 967–974, 2023.

[8] S Roshini, Sanjana Prakash, J Shilpha Dharshini, M N Saroja, and J Dhivya. Decision tree and knn analysis for hr analytics data. In *2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, pages 1–4, 2021.

[9] G. Shobana, S. Nikkath Bushra, K. Uma Maheswari, and Nalini Subramanian. Multivariate classification of dry beans using pipelined dimensionality reduction technique. In *2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, pages 1–6, 2022.

[10] Sakshi Shriya, Vipin Kumar, and Prem Shankar Singh Aydav. Dry beans classification using ensemble learning. In *2023 3rd International Conference on Smart Data Intelligence (ICSMDI)*, pages 327–334, 2023.

[11] Nanta Sooraksa. A survey of using computational intelligence (ci) and artificial intelligence (ai) in human resource (hr) analytics. In *2021 7th International Conference on Engineering, Applied Sciences and Technology (ICEAST)*, pages 129–132, 2021.

[12] Patitta Suksomboon and Amarita Ritthipakdee. Performance comparison classification using k-nearest neighbors and random forest classification techniques. In *2022 3rd International Conference on Big Data Analytics and Practices (IBDAP)*, pages 43–46, 2022.