Hepatitis-C Classification Problem (Healthcare Data)

Milestone: Performance Evaluation and Interpretation

Student 1: Aditya Gorakh Velapurkar

Student 2: Sahil Sushil Mahajan

Phone No: 617-602-0993
Phone No: 339-231-9594

velapurkar.a@northeastern.edu
mahajan.sah@northeastern.edu

Percentage of Effort Contributed by Aditya Gorakh Velapurkar: 50%
Percentage of Effort Contributed by Sahil Sushil Mahajan: 50%

Signature of Student 1: Aditya Gorakh Velapurkar
Signature of Student 2: Sahil Sushil Mahajan

Submission Date: 04:23:2023

Problem Setting:

The Hepatitis C virus (HCV) is responsible for the deadly medical disorder known as hepatitis C, which attacks the liver. Numerous symptoms, such as exhaustion, fever, nausea, and abdominal discomfort, may result from it. In some instances, it can also result in liver cancer, cirrhosis, and chronic liver disease. So that prompt action can be taken to stop the spread of the illness and its repercussions, we need an algorithm that can predict the possibility that a person has Hepatitis C.

Problem Definition:

The current task is managing chronic hepatitis C (CHC) patients by foretelling the stage of liver fibrosis using a supervised learning methodology. A labeled dataset of CHC patients with common laboratory data and their corresponding stage of liver fibrosis (severe fibrosis or cirrhosis) should be used to train the algorithm. This can be accomplished by using common supervised learning methods including logistic regression, decision trees, random forests, and neural networks. The aim of our project is to train and test the classification models and find the perfect model which will help in classification of the data so that whenever there is any new data added in this dataset, we will be able to classify the blood donors into correct categories with highest accuracy possible. The model's performance can be measured using measures like accuracy, precision, recall, and F1-score.

Data Sources:

The data for this project Hepatitis-C Classification Problem has been taken fromUCI Machine Learning Repository.

https://archive.ics.uci.edu/ml/datasets/HCV+data

Data Description:

The dataset we are going to use is HCV data set It contains 614 records and 14 columns at present, our target variable will be category I.e.(0,0s,1,2,3) and others are the dependent variables. Following are the attributes:

- Category: Patients with various stages of CHC or risk levels, for example, could be represented by distinct categories or groups in this column. Categorized into five types: 0 = Blood Donor, 0s = Suspect Blood Donors, 1 = Hepatitis, 2 = Fibrosis, and 3 = Cirrhosis.
- gender: The gender of the patient, which may be either male or female, is represented by this column.
- Age: The patient's age is shown in years in this column.
- ALB: The amount of albumin in the patient's blood is shown in this column. The liver produces albumin, a protein that aids in preventing circulatory fluids from seeping into neighboring tissues.
- ALP: The amount of alkaline phosphatase in the patient's blood is shown in this column.

  The liver, bones, and intestines are just a few of the body's tissues that contain the enzyme

  alkaline phosphatase.
- ALT: The amount of alanine aminotransferase in the patient's blood is shown in this column. High levels of the enzyme ALT, which is mostly present in the liver, in the blood can indicate liver damage.
- AST: The amount of aspartate aminotransferase in the patient's blood is shown in this column. Although AST is an enzyme that is present in many bodily tissues, elevated blood levels may be a sign of liver disease.
- BIL: The amount of bilirubin in the patient's blood is shown in this column. High amounts of the yellow pigment bilirubin, which is produced when red blood cells degrade, in the blood can signify liver issues.
- CHE: The amount of cholinesterase in the patient's blood is shown in this column. Low levels of the cholinesterase enzyme, which helps control nervous system function, in the blood can signify liver disease.
- CHOL: The amount of cholesterol in the patient's blood is shown in this column. Although high blood cholesterol levels can increase the risk of developing heart disease, cholesterol is a form of fat that is necessary for numerous body processes.
- CREA: The patient's blood creatinine levels are shown in this column. A waste product called creatinine is created by the muscles and removed from the blood by the kidneys. High levels in the blood may be a sign of kidney disease.

- GGT: The levels of gamma-glutamyl transferase in the patient's blood are shown in this column. GGT is an enzyme that is present in many bodily tissues, however having excessive amounts inthe blood can indicate liver impairment.
- PROT: The amount of total protein in the patient's blood is shown in this column. Low levels of total protein, which includes albumin and other blood proteins, may indicate liver or renal issues.

Categorical Attributes = 2

Numerical Attributes =12

Number of Null values = 1 (ALB),

18(ALP), 1(ALT),1(PROT)

Number of Outliers = 0

- Following is how the initial dataset looks like:

	Unnamed: 0	Category	Age	Sex	ALB	ALP	ALT	AST	BIL	CHE	CHOL	CREA	GGT	PROT
0	1	0=Blood Donor	32	m	38.5	52.5	7.7	22.1	7.5	6.93	3.23	106.0	12.1	69.0
1	2	0=Blood Donor	32	m	38.5	70.3	18.0	24.7	3.9	11.17	4.80	74.0	15.6	76.5
2	3	0=Blood Donor	32	m	46.9	74.7	36.2	52.6	6.1	8.84	5.20	86.0	33.2	79.3
3	4	0=Blood Donor	32	m	43.2	52.0	30.6	22.6	18.9	7.33	4.74	80.0	33.8	75.7
4	5	0=Blood Donor	32	m	39.2	74.1	32.6	24.8	9.6	9.15	4.32	76.0	29.9	68.7

- Following are the data types of the Attributes of the dataset:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 615 entries, 0 to 614
Data columns (total 14 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Unnamed: 0  615 non-null    int64  
 1   Category    615 non-null    object 
 2   Age         615 non-null    int64  
 3   Sex         615 non-null    object 
 4   ALB         614 non-null    float64
 5   ALP         597 non-null    float64
 6   ALT         614 non-null    float64
 7   AST         615 non-null    float64
 8   BIL         615 non-null    float64
 9   CHE         615 non-null    float64
 10  CHOL        605 non-null    float64
 11  CREA        615 non-null    float64
 12  GGT         615 non-null    float64
 13  PROT        614 non-null    float64
dtypes: float64(10), int64(2), object(2)
memory usage: 67.4+ KB
```

Data Collection and Processing:

Following are the steps that we carried out for Preprocessing our data of Blood Donors:

1. First we are using the formula df.isnull().sum().sort values(ascending=False)/len(df)*100 todiscover any missing values in the data set.
2. The percentage of missing values for each column in the Pandas Data Frame df is displayedas a result, sorted by decreasing missing value percentage.
3. In order to proceed with further analysis, it may be necessary to address columns with a higpercentage of missing values, which can be identified using this information.

Number of Null Value in each column:

```
#Checking the dataset whether it contains missing values or not.
df_missing= df.isnull().sum().sort_values(ascending=False)/len(df)*100
df_missing
```

```
ALP           2.926829
CHOL          1.626016
ALB           0.162602
ALT           0.162602
PROT          0.162602
Unnamed: 0    0.000000
Category      0.000000
Age           0.000000
Sex           0.000000
AST           0.000000
BIL           0.000000
CHE           0.000000
CREA          0.000000
GGT           0.000000
dtype: float64
```

4. The number of empty (null) values in each column of a Pandas Data Frame is shown in thisstatement.

```
[ ]  df.isnull().sum()

    Unnamed: 0      0
    Category        0
    Age             0
    Sex             0
    ALB             1
    ALP            18
    ALT             1
    AST             0
    BIL             0
    CHE             0
    CHOL           10
    CREA            0
    GGT             0
    PROT            1
    dtype: int64
```

5. Checked for any duplicate Rows in dataset:

```
df.duplicated().sum()

0
```

6. The following stage entails carrying out two main opera/ons: forward filling and impu/ng missing values.

```
df_mean = df.groupby ('Category Number').mean()

for column in df.columns :
  if df[column].isna().sum() >= 1:
    df[column].fillna(df.groupby('Category Number')[column].transform('mean'), inplace = True)
  else:
    df[column].fillna(method='ffill', inplace=True)
```

```
<ipython-input-11-1f220c2bfd04>:1: FutureWarning: The default value of numeric_only in DataFrameGroupBy.mean
  df_mean = df.groupby ('Category Number').mean()
```

```
df.head()
```

	Sr no	Age	Sex	ALB	ALP	ALT	AST	BIL	CHE	CHOL	CREA	GGT	PROT	Category Number	Category type
0	1	32	1	38.5	52.5	7.7	22.1	7.5	6.93	3.23	106.0	12.1	69.0	0	Blood Donor
1	2	32	1	38.5	70.3	18.0	24.7	3.9	11.17	4.80	74.0	15.6	76.5	0	Blood Donor
2	3	32	1	46.9	74.7	36.2	52.6	6.1	8.84	5.20	86.0	33.2	79.3	0	Blood Donor
3	4	32	1	43.2	52.0	30.6	22.6	18.9	7.33	4.74	80.0	33.8	75.7	0	Blood Donor
4	5	32	1	39.2	74.1	32.6	24.8	9.6	9.15	4.32	76.0	29.9	68.7	0	Blood Donor

7. Removing Column name (Named: 0) from the dataset and replacing m to 1 and f to 0 in &column.

```
df = df.rename(columns={'Unnamed: 0': 'Sr no'})
```

```
import pandas as pd

# Changing m to 1 and f to 0
df['Sex'] = df['Sex'].replace({'m': 1, 'f': 0})
```

```
df.head()
```

	Sr no	Age	Sex	ALB	ALP	ALT	AST	BIL	CHE	CHOL	CREA	GGT	PROT	Category Number	Category type
0	1	32	1	38.5	52.5	7.7	22.1	7.5	6.93	3.23	106.0	12.1	69.0	0	Blood Donor
1	2	32	1	38.5	70.3	18.0	24.7	3.9	11.17	4.80	74.0	15.6	76.5	0	Blood Donor
2	3	32	1	46.9	74.7	36.2	52.6	6.1	8.84	5.20	86.0	33.2	79.3	0	Blood Donor
3	4	32	1	43.2	52.0	30.6	22.6	18.9	7.33	4.74	80.0	33.8	75.7	0	Blood Donor
4	5	32	1	39.2	74.1	32.6	24.8	9.6	9.15	4.32	76.0	29.9	68.7	0	Blood Donor

8. We have finally reached a point where the data processing is finished, and the dataset contains no missing or null values. As a result, we may move on with data visualization and exploration and get new knowledge and insights about the data.

Statistics of our dataset:

```
# print the statistics
print(statistics)
```

```
                Age         Sex         ALB         ALP         ALT         AST  \
count   615.000000  615.000000  615.000000  615.000000  615.000000  615.000000
mean     47.408130    0.613008   41.605338   67.954053   28.448293   34.786341
std      10.055105    0.487458    5.787660   26.086593   25.449016   33.090690
min      19.000000    0.000000   14.900000   11.300000    0.900000   10.600000
25%      39.000000    0.000000   38.800000   52.200000   16.400000   21.600000
50%      47.000000    1.000000   41.900000   66.000000   23.000000   25.900000
75%      54.000000    1.000000   45.200000   80.350000   33.050000   32.900000
max      77.000000    1.000000   82.200000  416.600000  325.300000  324.000000

                BIL         CHE         CHOL        CREA         GGT        PROT
count   615.000000  615.000000  615.000000  615.000000  615.000000  615.000000
mean     11.396748    8.196634    5.363858   81.287805   39.533171   72.040897
std      19.673150    2.205657    1.126647   49.756166   54.661071    5.398832
min       0.800000    1.420000    1.430000    8.000000    4.500000   44.800000
25%       5.300000    6.935000    4.605000   67.000000   15.700000   69.300000
50%       7.300000    8.260000    5.300000   77.000000   23.300000   72.200000
75%      11.200000    9.590000    6.055000   88.000000   40.200000   75.400000
max     254.000000   16.410000    9.670000 1079.100000  650.900000   90.000000
```

Cleaned Dataset:

```
df.isnull().sum()
```

```
Sr no              0
Age                0
Sex                0
ALB                0
ALP                0
ALT                0
AST                0
BIL                0
CHE                0
CHOL               0
CREA               0
GGT                0
PROT               0
Category Number    0
Category type      0
dtype: int64
```

Correlation Matrix of our dataset:

Correlation Graph

```
1 df.corr()
```

```
<ipython-input-33-2f6f6606aa2c>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future v
  df.corr()
```

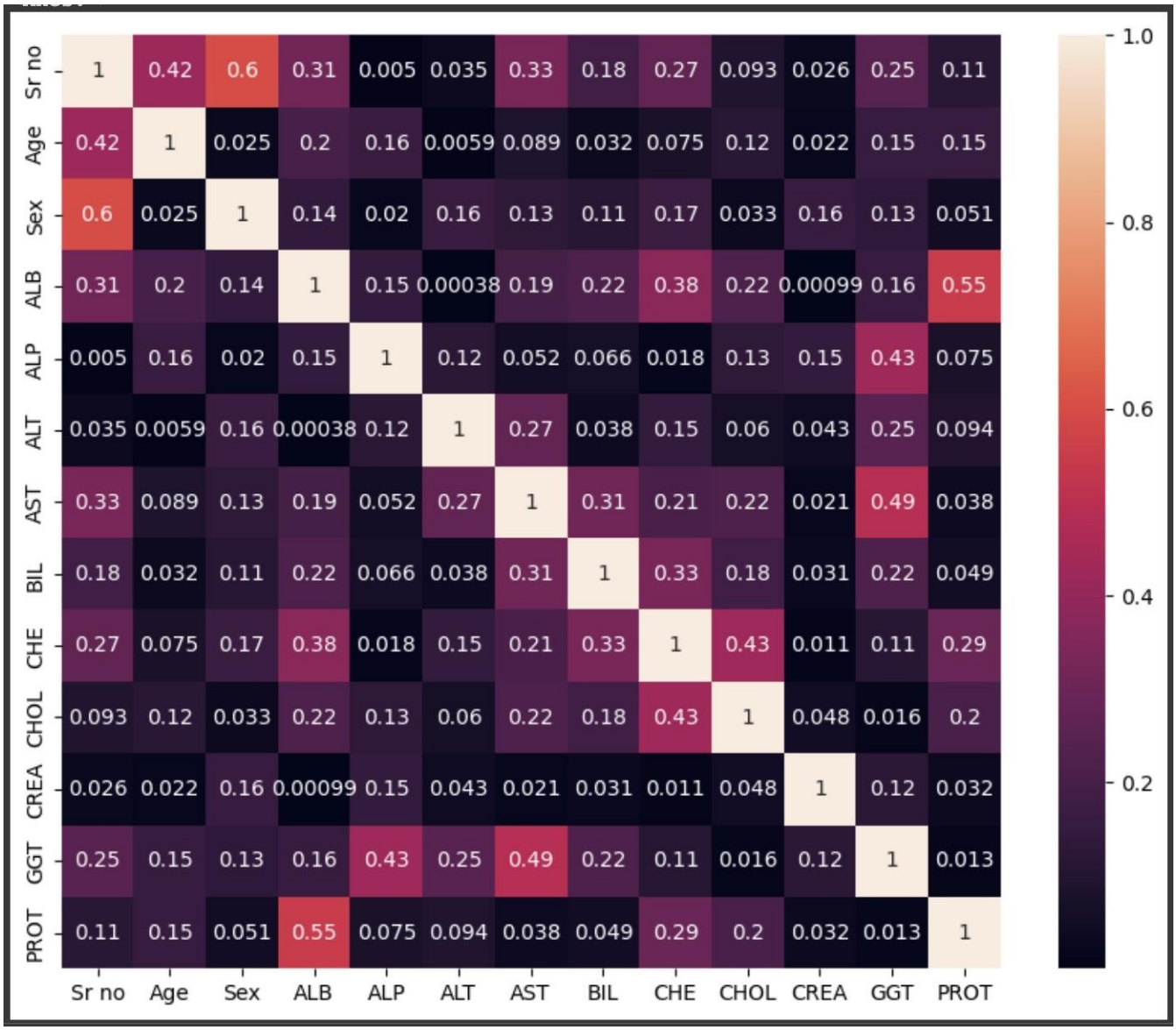
	Sr no	Age	Sex	ALB	ALP	ALT	AST	BIL	CHE	CHOL	CREA	GGT	PROT
Sr no	1.000000	0.420477	-0.598597	-0.313035	0.004965	-0.035126	0.332626	0.181459	-0.270549	-0.093214	-0.025987	0.247781	-0.114680
Age	0.420477	1.000000	-0.024544	-0.201104	0.164186	-0.005924	0.088666	0.032492	-0.075093	0.116949	-0.022296	0.153087	-0.153564
Sex	-0.598597	-0.024544	1.000000	0.143727	-0.020063	0.161706	0.130891	0.111177	0.169111	-0.032749	0.159589	0.133276	0.050767
ALB	-0.313035	-0.201104	0.143727	1.000000	-0.146022	0.000378	-0.194497	-0.221410	0.375898	0.216300	-0.000987	-0.158413	0.547787
ALP	0.004965	0.164186	-0.020063	-0.146022	1.000000	0.124558	0.052177	0.066369	0.017605	0.131276	0.146530	0.427945	-0.075320
ALT	-0.035126	-0.005924	0.161706	0.000378	0.124558	1.000000	0.273329	-0.038446	0.146890	0.059798	-0.043018	0.248122	0.093736
AST	0.332626	0.088666	0.130891	-0.194497	0.052177	0.273329	1.000000	0.312231	-0.208536	-0.215980	-0.021387	0.491263	0.038463
BIL	0.181459	0.032492	0.111177	-0.221410	0.066369	-0.038446	0.312231	1.000000	-0.333172	-0.180758	0.031224	0.217024	-0.048716
CHE	-0.270549	-0.075093	0.169111	0.375898	0.017605	0.146890	-0.208536	-0.333172	1.000000	0.426988	-0.011157	-0.110345	0.294982
CHOL	-0.093214	0.116949	-0.032749	0.216300	0.131276	0.059798	-0.215980	-0.180758	0.426988	1.000000	-0.048244	-0.015936	0.204427
CREA	-0.025987	-0.022296	0.159589	-0.000987	0.146530	-0.043018	-0.021387	0.031224	-0.011157	-0.048244	1.000000	0.121003	-0.032085
GGT	0.247781	0.153087	0.133276	-0.158413	0.427945	0.248122	0.491263	0.217024	-0.110345	-0.015936	0.121003	1.000000	-0.012817
PROT	-0.114680	-0.153564	0.050767	0.547787	-0.075320	0.093736	0.038463	-0.048716	0.294982	0.204427	-0.032085	-0.012817	1.000000

Data Exploration & Visualization:

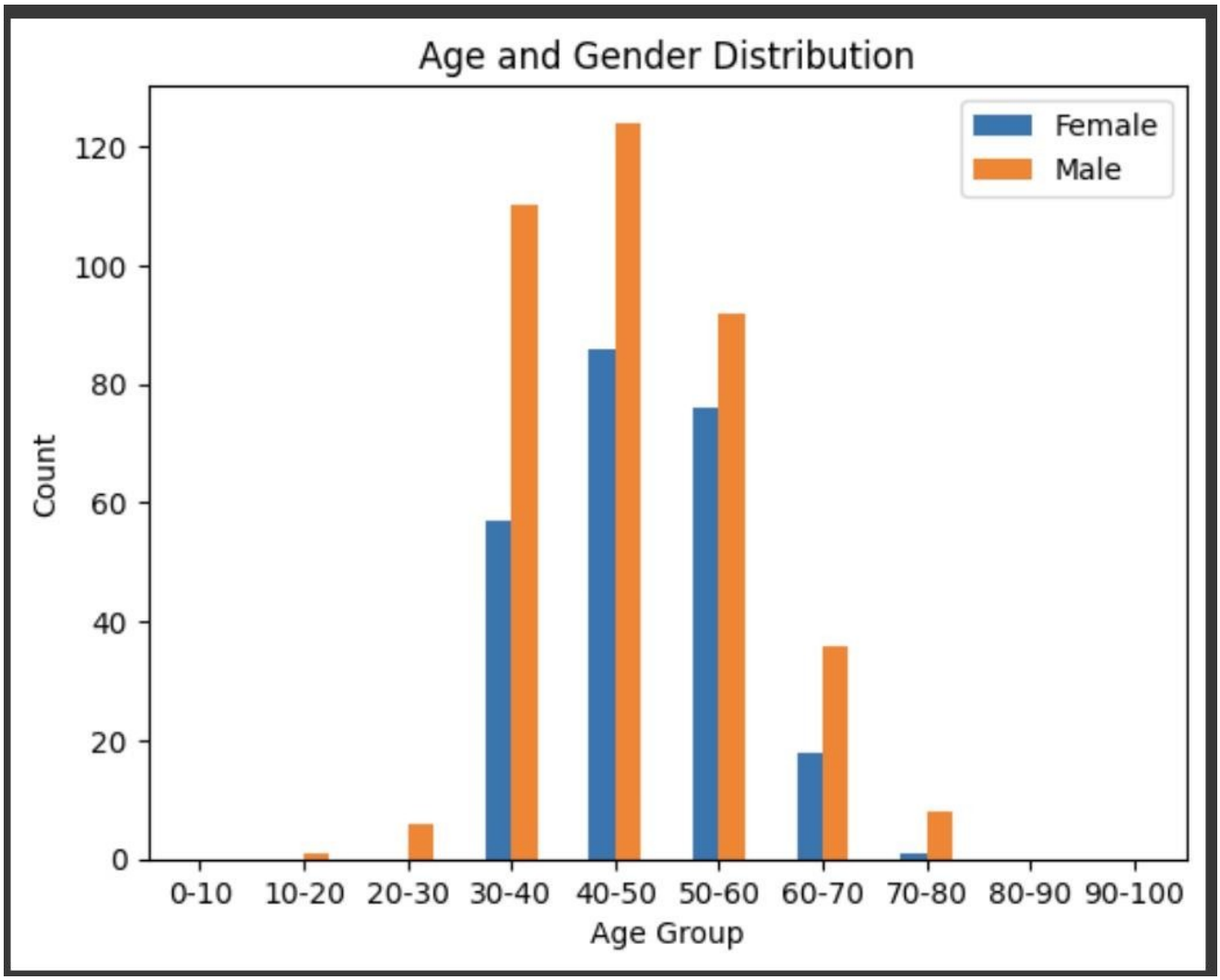
Data visualization plays a crucial role in supervised machine learning by allowing us to explore and understand the data, identify patterns and relationships, and select relevant features to build effective predictive models.

We have created various data visualizations in order to better understand the data and identify the trends and outliers if any.

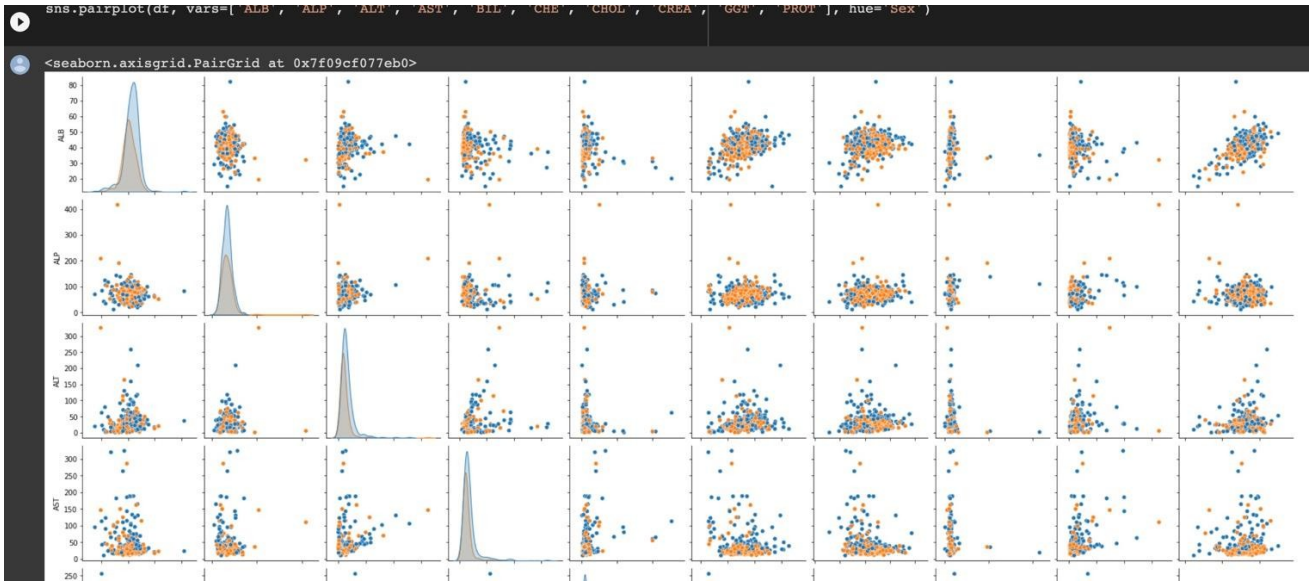
Following is the correlation matrix heat map to find the correlation between all the variables and if any is >=0.9 then can be eliminated or ignored.



1. We have generated a bar chart which shows age groups as well as gender wise distribution of the Blood Donors in our dataset:



2. We have generated multiple Density plots and scatter plots of attributed used in dataset to identify if there are any outliers present:



Data Mining Models / Methods (Classification Problem):

Following are the classification models that were used in the project:

- Random Forest Classifier
- Logistic Regression
- Support Vector Machine Classifier
- Naïve Bayes Classifier
- Decision Tree Classifier
- K-Nearest Neighbors Classifier
- Multi-Layer Perceptron Classifier (Neural Network)

Following are the metrics that were used to evaluate all the classification models:

- Accuracy
- Precision
- Recall / Sensitivity
- Confusion Matrix
- Specificity
- F1 Score
- ROC Curve (AUC Score)
- Gains Chart

Performance Evaluation and Interpretation:

Model 1: Logistic Regression

```
Logistic Regression Accuracy: 0.918918918918919
Logistic Regression Confusion matrix:
 [[152   1   0   0   0]
 [  1   2   0   0   0]
 [  2   0   5   2   1]
 [  3   0   0   5   0]
 [  3   0   0   2   6]]
Logistic Regression Classification report:
              precision    recall  f1-score   support

           0       0.94      0.99      0.97       153
          0s       0.67      0.67      0.67         3
           1       1.00      0.50      0.67        10
           2       0.56      0.62      0.59         8
           3       0.86      0.55      0.67        11

    accuracy                           0.92       185
   macro avg       0.80      0.67      0.71       185
weighted avg       0.92      0.92      0.91       185
```

Model 2: Decision Tree

```
Decision Tree Accuracy: 0.8810810810810811
Decision Tree Confusion matrix:
 [[150   1   2   0   0]
 [  0   2   0   0   1]
 [  3   2   2   3   0]
 [  4   1   1   2   0]
 [  2   0   0   2   7]]
Decision Tree Classification report:
              precision    recall  f1-score   support

           0       0.94      0.98      0.96       153
          0s       0.33      0.67      0.44         3
           1       0.40      0.20      0.27        10
           2       0.29      0.25      0.27         8
           3       0.88      0.64      0.74        11

    accuracy                           0.88       185
   macro avg       0.57      0.55      0.54       185
weighted avg       0.87      0.88      0.87       185
```

Model 3: Random Forest

```
Random Forest Accuracy: 0.9027027027027027
Random Forest Confusion matrix:
 [[153   0   0   0   0]
 [  2   1   0   0   0]
 [  5   0   3   2   0]
 [  5   0   0   3   0]
 [  4   0   0   0   7]]
Random Forest Classification report:
              precision    recall  f1-score   support

           0       0.91      1.00      0.95       153
          0s       1.00      0.33      0.50         3
           1       1.00      0.30      0.46        10
           2       0.60      0.38      0.46         8
           3       1.00      0.64      0.78        11

    accuracy                           0.90       185
   macro avg       0.90      0.53      0.63       185
weighted avg       0.90      0.90      0.89       185
```

Model 4: K-Nearest Neighbors

```
K-Nearest Neighbors Accuracy: 0.8486486486486486
K-Nearest Neighbors Confusion matrix:
 [[153   0   0   0   0]
 [  3   0   0   0   0]
 [  9   0   0   1   0]
 [  7   0   0   1   0]
 [  7   0   0   1   3]]
K-Nearest Neighbors Classification report:
              precision    recall  f1-score   support

           0       0.85      1.00      0.92       153
          0s       0.00      0.00      0.00         3
           1       0.00      0.00      0.00        10
           2       0.33      0.12      0.18         8
           3       1.00      0.27      0.43        11

    accuracy                           0.85       185
   macro avg       0.44      0.28      0.31       185
weighted avg       0.78      0.85      0.80       185
```

Model 5: Support Vector Machine

```
Support Vector Machine Accuracy: 0.8756756756756757
Support Vector Machine Confusion matrix:
 [[153   0   0   0   0]
 [  0   0   0   0   3]
 [  8   0   0   1   1]
 [  4   0   0   3   1]
 [  3   0   0   2   6]]
Support Vector Machine Classification report:
              precision    recall  f1-score   support

           0       0.91      1.00      0.95       153
          0s       0.00      0.00      0.00         3
           1       0.00      0.00      0.00        10
           2       0.50      0.38      0.43         8
           3       0.55      0.55      0.55        11

    accuracy                           0.88       185
   macro avg       0.39      0.38      0.39       185
weighted avg       0.81      0.88      0.84       185
```

Model 6: Multi-Layer Perceptron Classifier Algorithm (Artificial Neural Network)

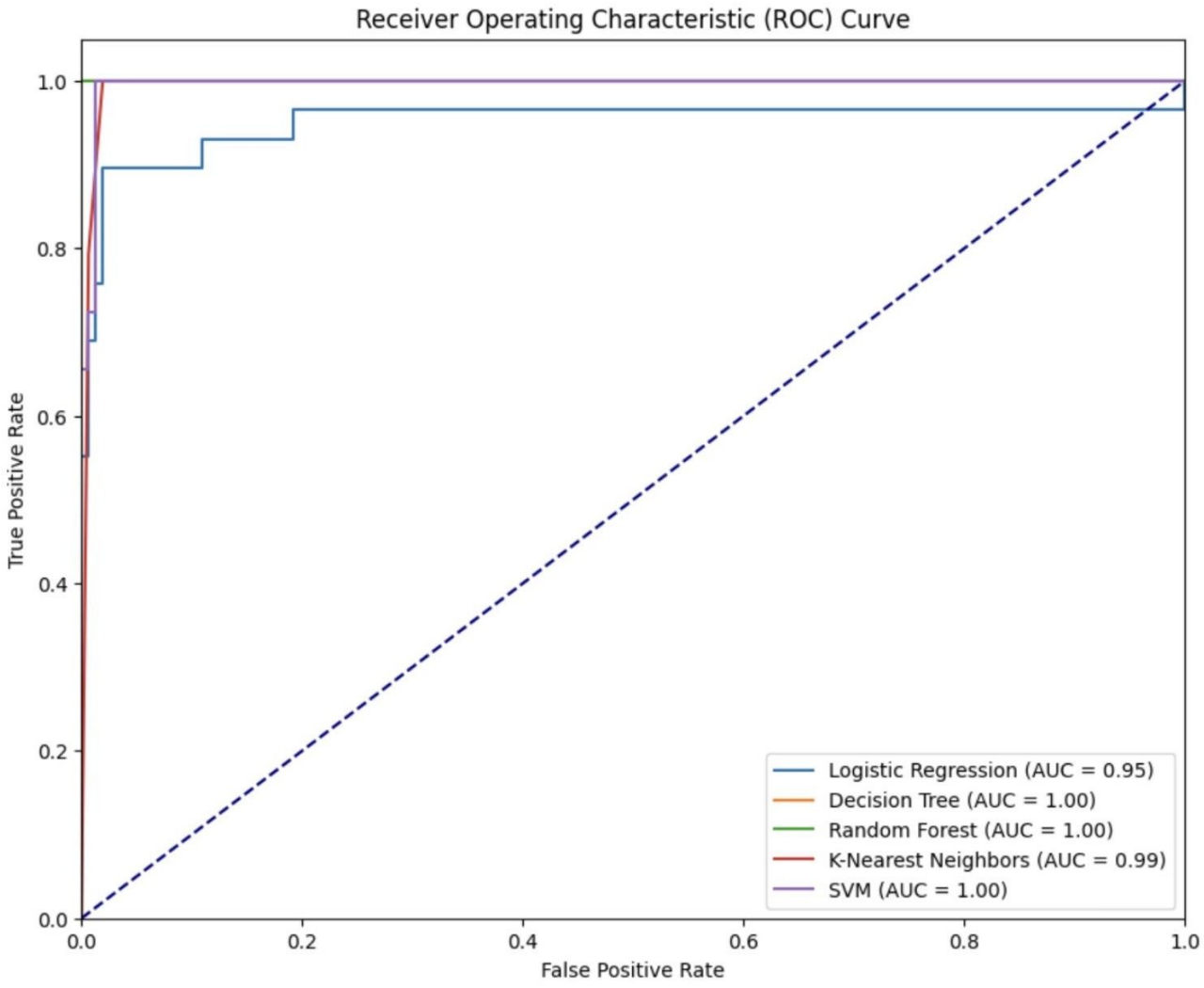
```
Multi-Layer Perceptron Classifier Algorithm (Artificial Neural Network)
Training accuracy: 0.9976744186046511
Testing accuracy: 0.9027027027027027
F1 score: 0.894159070222702
Confusion matrix:
 [[152   1   0   0   0]
 [  0   0   0   0   3]
 [  2   0   5   3   0]
 [  1   0   3   4   0]
 [  3   0   1   1   6]]
Classification report:
              precision    recall  f1-score   support

           0       0.96      0.99      0.98       153
          0s       0.00      0.00      0.00         3
           1       0.56      0.50      0.53        10
           2       0.50      0.50      0.50         8
           3       0.67      0.55      0.60        11

    accuracy                           0.90       185
   macro avg       0.54      0.51      0.52       185
weighted avg       0.89      0.90      0.89       185
```

Model Selection:

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	91.89%	0.92	0.92	0.91
Random Forest	91.00%	0.91	0.91	0.9
KNN	84.86%	0.78	0.85	0.8
SVM	87.56%	0.81	0.88	0.84
Decision Tree	88%	0.89	0.89	0.87
Neural Network	89%	0.87	0.88	0.86

We have compared all the parameters given above and came to conclusion that Logistic Regression model is the best fit for our problem.

ROC Curve with AUC Scores:



Implementation of Final Model (Logistic Regression):

```
Logistic Regression
Training accuracy: 0.9697674418604652
Testing accuracy: 0.918918918918919
F1 score: 0.9126122750701245
Confusion matrix:
 [[152   1   0   0   0]
 [  1   2   0   0   0]
 [  2   0   5   2   1]
 [  3   0   0   5   0]
 [  3   0   0   2   6]]
Classification report:
              precision    recall  f1-score   support

           0       0.94      0.99      0.97       153
          0s       0.67      0.67      0.67         3
           1       1.00      0.50      0.67        10
           2       0.56      0.62      0.59         8
           3       0.86      0.55      0.67        11

    accuracy                           0.92       185
   macro avg       0.80      0.67      0.71       185
weighted avg       0.92      0.92      0.91       185
```

Project Results:

Why Logistic Regression Algorithm performs the best as compared to the other models used in our project?

As compared to the other models the training and testing accuracy, F1-score, Precision and Recall is the highest in case of Logistic Regression therefore this model is best fit for our problem.

- Accommodates non-linear relationships effectively.
- Handles missing values well.
- Ensemble Learning
- Resilient to outliers
- Attribute Importance

Potential Drawbacks / Disadvantages of Logistic Regression Algorithm:

- Can Lead to overfitting of model.
- Inclined towards categorical variables.
- Susceptible to noisy data
- As we have 5 Output variables therefore, we are unable to create Gains Chart to Evaluate performance.

Solutions to the drawbacks of Random Forest Classifier Algorithm:

- Using hyper parameter tuning of Logistic regression model optimum accuracy can be achieved and overfitting of the model can be avoided.
- Regularization techniques like L1 and L2 regularization can be used to penalize the coefficients of the model and prevent overfitting.
- Cross-validation techniques like K-fold cross-validation can be used to assess the performance of the model on independent data.
- One-hot encoding can be used to convert categorical variables into binary variables, which can help in reducing the bias towards categorical variables.
- Outlier detection and removal techniques can be used to remove noisy data points from the dataset.
- Robust statistical measures like median and interquartile range can be used instead of mean and standard deviation to reduce the effect of outliers on the model.

Impact of the project outcomes:

Our project is successful in categorizing the Blood donor's data into given five types: 0 = Blood Donor, 0s = Suspect Blood Donors, 1 = Hepatitis, 2 = Fibrosis, and 3 = Cirrhosis.

As a result, the logistic regression model has demonstrated its effectiveness in classifying the issue of predicting liver fibrosis in people with chronic hepatitis C (CHC). The algorithm was able to predict both severe fibrosis and cirrhosis in CHC patients using regular laboratory data. Comparing the creation of a single, basic model to the present complex models, which require different equations to categorize cirrhosis and severe fibrosis, is a huge advance. The medical industry will be able to identify CHC patients based on the lab results of blood donors and treat them pro-actively as a result thanks to this logistic regression model. A major improvement in the diagnosis and treatment of CHC patients has been made overall thanks to the effective application of this logistic regression model.

Additionally, the logistic regression model will be more accurate at classifying blood donors into the proper categories of severe fibrosis and cirrhosis when more data from blood donors is added to the dataset. By using test data to detect and diagnose CHC patients, this will give medical professionals and clinicians a trustworthy tool. By enabling early detection and treatment of liver fibrosis in CHC patients, the application of this model in the medical field can considerably enhance patient outcomes. In conclusion, the continued application and enhancement of a logistic regression model will assist physicians and other healthcare professionals in the efficient management of CHC patients.