University of West London

Coursework Assignment

Responsible AI- Fairness and Biases

Darsheta Dinesh Babu

MSc. Artificial Intelligence

Task 2- Fairness and Bias

1. Abstract

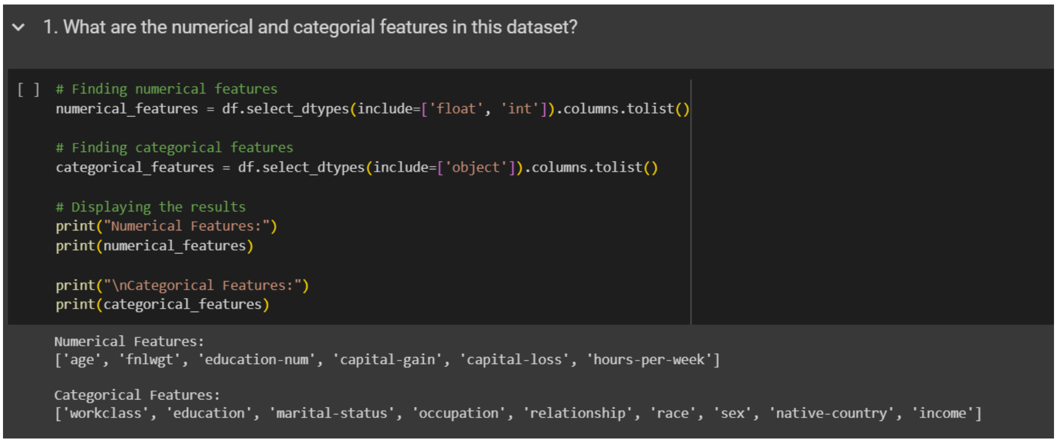
This analysis focuses on exploring the Adult census dataset with a focus on detecting and reducing any biases in Machine Learning model while keeping fairness concerns in mind. The dataset comprises demographic data from individuals who participated in the 1994 Census. This dataset has 32561 rows and 15 columns where the features are 'age', 'workclass', 'fnlwgt', 'education', 'education-num','marital-status', 'occupation', 'relationship', 'race', 'sex', 'capital-gain', 'capital-loss', 'hours-per-week', 'native-country', 'income'. The main aim of this analysis is to find the numerical and categorical features, finding missing values which might impact other variables, decreasing bias in the model, provide histograms to show data skew in various features, to visually compare education level and income relationship and finally gender distribution across marital status. Addressing the missing values, data skewness and understanding the relationship between different features and also the relationship of the features with the target variable is crucial in mitigating the bias and also ensuring standard model predictions. The purpose of this analysis is to show how bias and fairness affects and enhances the model performance and accuracy.

2. Analysis

2.1 Dataset Exploration

The necessary packages are imported first to start exploring the dataset in order to get meaningful insights. The dataset is imported directly from the UCI Machine Learning Repository and the basic information about the dataset is displayed like the rows and columns and the datatype of the features present in the dataset.

To address the first question **“What are the numerical and categorical features in the dataset?”**



1. What are the numerical and categorial features in this dataset?

```
# Finding numerical features
numerical_features = df.select_dtypes(include=['float', 'int']).columns.tolist()

# Finding categorical features
categorical_features = df.select_dtypes(include=['object']).columns.tolist()

# Displaying the results
print("Numerical Features:")
print(numerical_features)

print("\nCategorical Features:")
print(categorical_features)
```

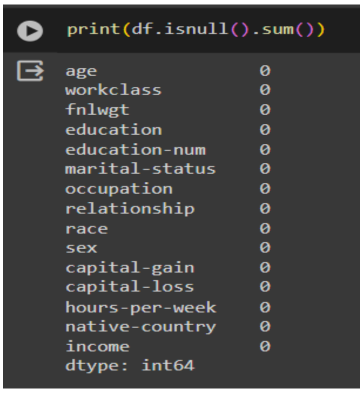
```
Numerical Features:
['age', 'fnlwgt', 'education-num', 'capital-gain', 'capital-loss', 'hours-per-week']

Categorical Features:
['workclass', 'education', 'marital-status', 'occupation', 'relationship', 'race', 'sex', 'native-country', 'income']
```

Fig 2.1.1 Numerical and Categorical Features

The above code created two separate variables to store the numerical and categorical values. The numerical features variable has the features which has int and float datatype and the categorical features variable contains the features whose datatype is object and the results are displayed.

2.2 Missing values analysis:

Identifying missing values and handling them is the most important step in machine learning as mostly missing values lead to bias and data skew and also decreases the accuracy of the model. At first when checked for missing values the model didn't provide any missing values but when analysed further, I could see the missing values in the dataset where specified as "?" which says that there are missing values in the dataset

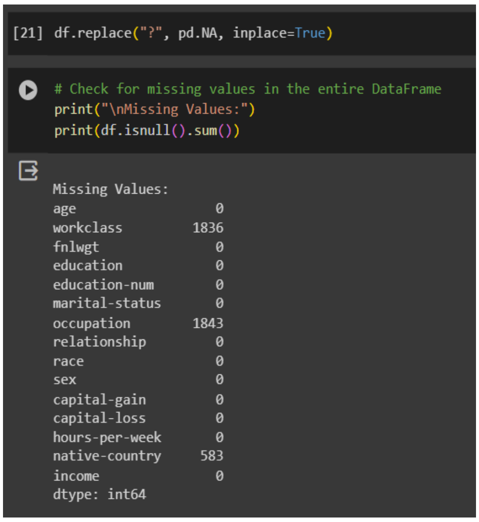


```
print(df.isnull().sum())
```

```
age               0
workclass         0
fnlwgt            0
education         0
education-num     0
marital-status    0
occupation        0
relationship      0
race              0
sex               0
capital-gain      0
capital-loss      0
hours-per-week    0
native-country    0
income            0
dtype: int64
```

Fig 2.2.1 Missing values before replacing "?"

To resolve this issue, I am replacing the "?" with Nan values so that the model knows that there are missing values in the features.



```
[21] df.replace("?", pd.NA, inplace=True)
```

```
# Check for missing values in the entire DataFrame
print("\nMissing Values:")
print(df.isnull().sum())
```

```

Missing Values:
age                  0
workclass         1836
fnlwgt               0
education            0
education-num        0
marital-status       0
occupation        1843
relationship         0
race                 0
sex                  0
capital-gain         0
capital-loss         0
hours-per-week       0
native-country     583
income               0
dtype: int64
```

Fig 2.2.2 Missing values after replacing "?"

Now we can see the missing values are represented in the dataset. Now, we need to handle these values. There are three features with missing values.

Now to address the second question **"Are there missing feature values for a large number of observations? If yes what are those features? Are there features that are missing that might affect other features?"**

We can see from above that some features have missing values. To explain what we did in the above code lets break down the code.

When analysing the data, I noticed that some values was represented as "?" which indicates that we have missing values. If we try to find the missing values, the model can't take it as a missing value though it is, as it is considered as a value too.

So, what we do is just replace the "?" with a Nan value so that the model takes it as a missing value.

The features which have the missing values are ***Workclass, Occupation and Native Country*****.** To handle missing values there are many strategies. Some are

1. Imputing values
2. Dropping-Deletion of rows or columns
3. Predicting missing values based on other features
4. Flagging-Introducing a new variable to indicate a value is missing

In the above case, the missing value percentage is very less so we can either impute the values which is filling the missing values with some other values or dropping the values.

In this case we are imputing it as there is not much missing values in the dataset. The missing value will somehow affect the other features and in order to eliminate that we do address the missing values either by imputing or deleting.

What we are going to do now is simply impute the features to fit in values so that it doesn't have missing values. We can do that by using mode which will find the most frequently occurring value in the feature and use that value to fill in the missing values. In the case of “Workclass” the missing values are replaced with “Private” and likewise the missing values in the “Occupation” is replaced with “Prof-speciality” and in “Native country” it is “United states”. Now the missing value is checked

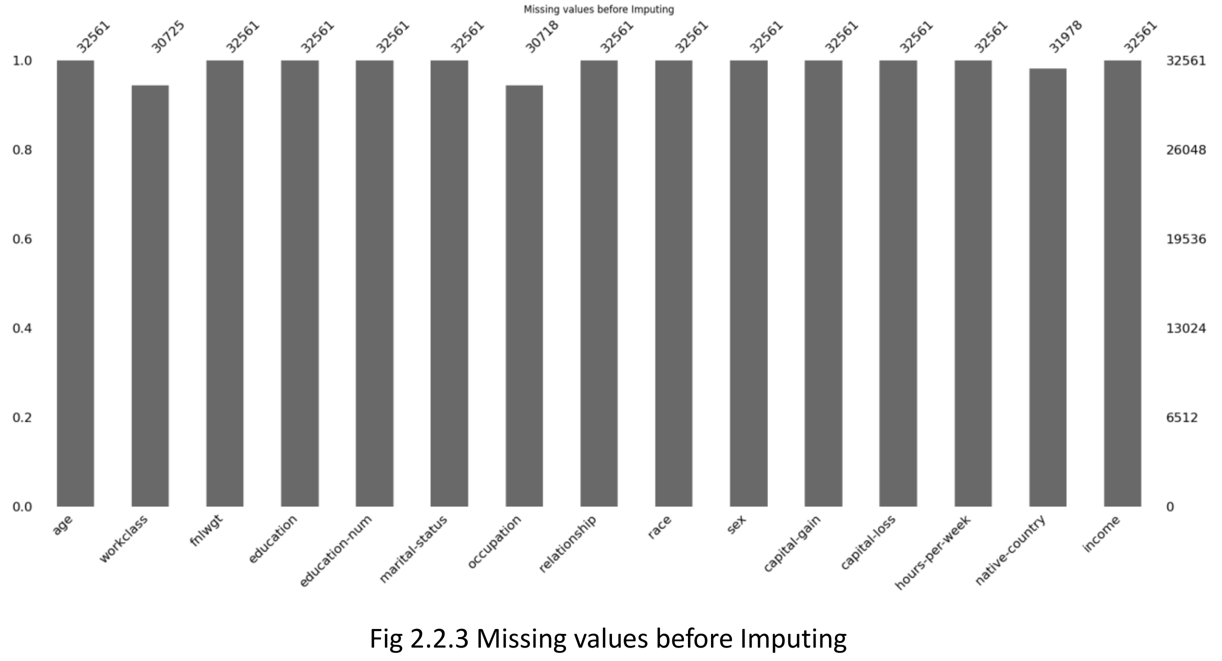


Fig 2.2.3 Missing values before Imputing

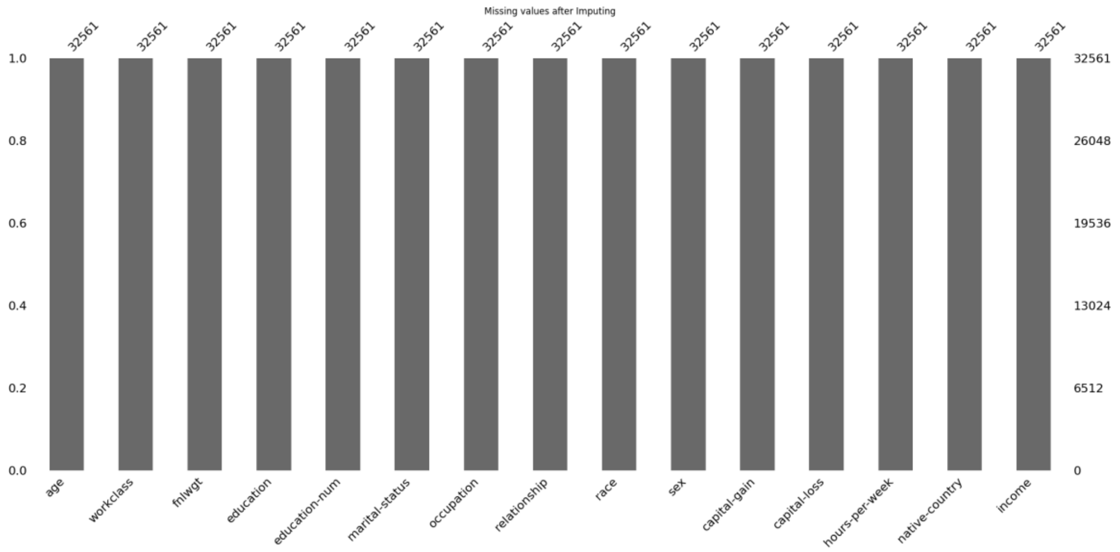


Fig 2.2.4 Missing values after Imputing

2.3 Data Skewness and Visualisation

The data skewness and visualisation are after Data Pre-processing where we analyse the features even further and get a deeper understanding of the data. Next, we do the visualising part where we graphically represent the relationship between all the features and the target variable. The target variable in our analyses is “Income”. To graphically represent the features when compared to the income variable, I used violin plots and bar plots.

To address the next question **"How would you describe the relationship between education level and income bracket in this dataset?"**

I use bar plot to visually show the relationship between the education level and income bracket in the dataset

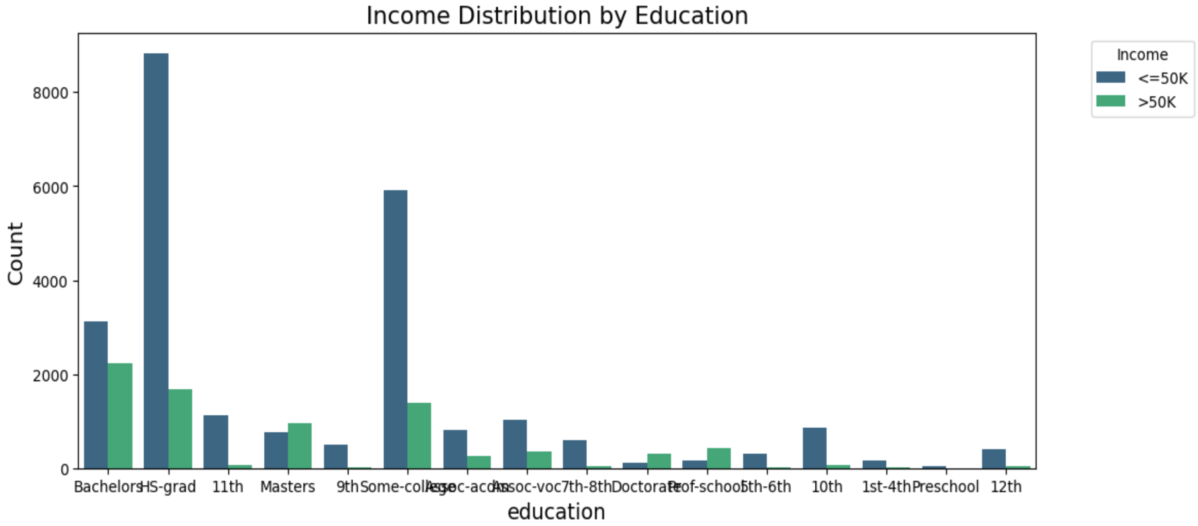


Fig 2.3.1 Income Distribution by Education

It is very well clear from the graph above how education level and income are connected with each other. As the graph state, the x axis has the education feature and the y axis contains the count of people. The green bar represents the people who have their income more than 50K and the blue bar represents the people who have their income less than 50K.

There tends to be a **positive correlation** between education level and income. This implies that people with higher education levels often earn more money overall than people with lower education levels.

It is very clear from the graph that there are more people who fall into the category of income less than 50k but when analysed the data of people who gets income more than 50k, we can see they either have one degree or more than one degree or at least a high school graduate. So, it is very clear that the education one possesses contributes equally to the amount of salary one gets.

In other features like 10th,9th and others except the college degrees, we could see that there are a lot of people who fall into the less than 50k category.

So, in short, the education level and the income bracket have positive correlation where if one increases the other one also increases

To address the question **"What noteworthy observations can you make about the gender distributions for each marital status category?"**

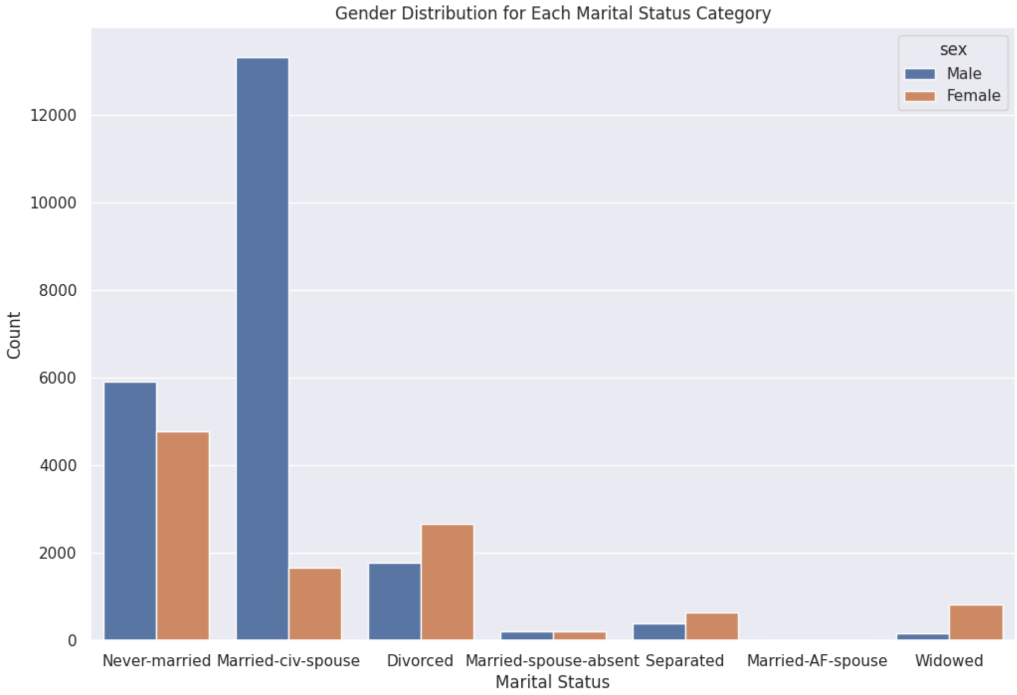


Fig 2.3.2 Gender Distribution for each Marital Status Category

The x-axis represents different marital status categories (e.g., Married, Never Married, Divorced, etc.). Each bar for a specific marital status category will be divided into two colours, representing the count of males and females.

It is very clear from the graph that there is a huge gender imbalance in the dataset. There is almost the same number of men and women in the never married category whereas the divorce rate is very high in women when compared to men.

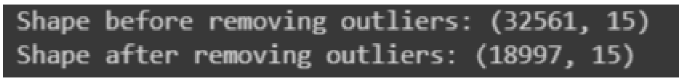
The married civ spouse particularly refers to those who are married to a civilian partner.A greater number of married civilian spouses among males may be due to a number of variables, including gender roles, society norms, or particular dataset features.

The category "married-spouse-absent" designates a person who is legally married but does not currently reside with their spouse. This group is meant to represent people who are married but find themselves in a scenario where their spouse is not present, maybe because of work-related travel, separation, or other situations and it is equal in both the categories so there isn't that much imbalance. There is a high rate of women in divorced and widowed category.

Therefore, when taken individual feature, there is a gender imbalance but when taken in common the imbalances are equal.

To address the question **“What signs of data skew do you see? Provide some histogram graphs for this question and interpret them”**

Before doing the data skew part we convert all the categorical values to numerical features as it will be easy for the machine learning model to deal with numerical values more than the categorical values. To convert the categorical features to numerical we use label encoder which assigns numerical values to all the categorical features. Now a new data frame df_copy is created with all the features as numerical. After that the outliers are detected for the numerical features alone and not the encoded features as it won’t make sense.



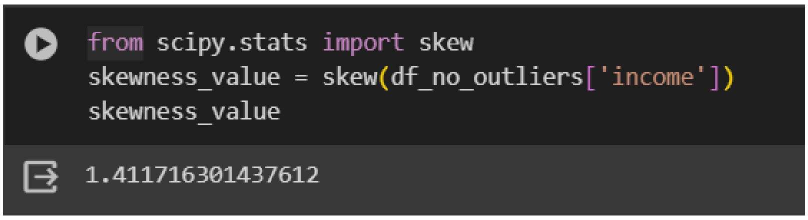
```
Shape before removing outliers: (32561, 15)
Shape after removing outliers: (18997, 15)
```

Fig 2.3.3 Outliers

As discussed earlier, the dataset contains a lot of categorical values

Skewness is a statistical measure designed for numerical data, and it's not directly applicable to categorical values. Skewness measures the asymmetry of a distribution, and categorical variables don't have a distribution in the same way continuous numerical variables do so the skewness of only the numerical value is found

We are finding the skew for the target variable i.e., income here



```
from scipy.stats import skew
skewness_value = skew(df_no_outliers['income'])
skewness_value
```

```
1.411716301437612
```

Fig 2.3.4 Skewness of target variable “Income”

It is clear from the value that the data skew is positive which means all the data points are clustered to the left.

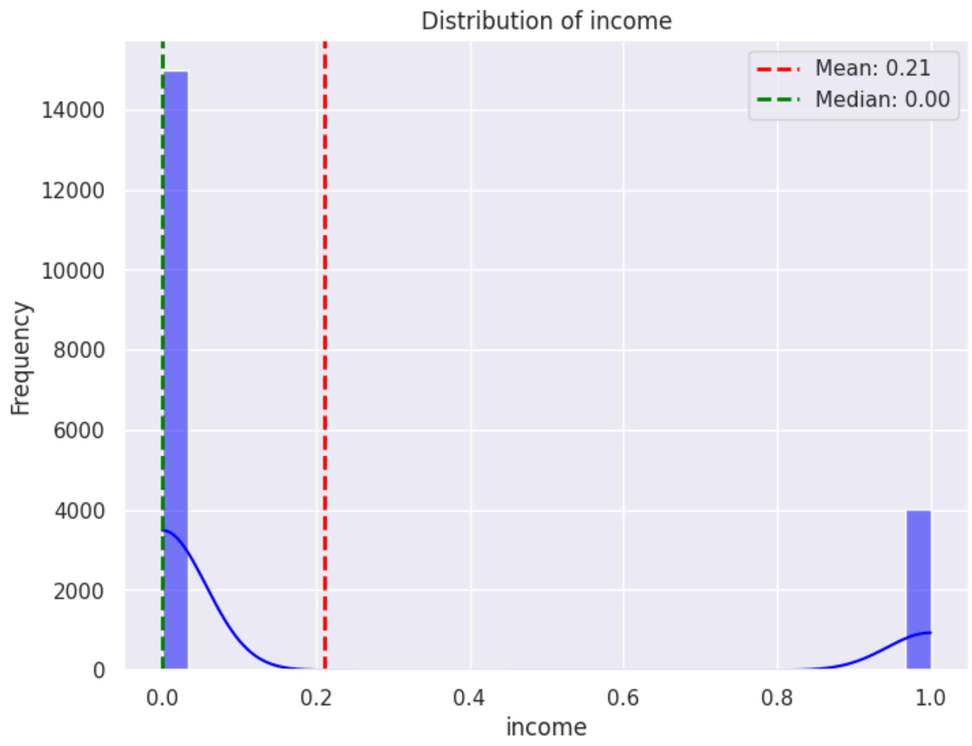


Fig 2.3.4 Distribution of Income

"1.411716301437612" is the number that indicates how skewed a distribution or dataset is. The asymmetry of a probability distribution is quantified by its skewness. A distribution that is positively skewed (skewed to the right) is indicated by a positive skewness value.

With a few bigger values extending the right tail, the majority of the data points are clustered on the left side of the distribution. Because the bigger values push the mean to the right, the mean (average) is usually greater than the median.

Practically speaking, a positive skewness frequently indicates that the dataset may contain outliers or higher values that are pushing the distribution to the right. When examining a dataset's form, it's a crucial indication, particularly in domains like finance and statistics.

Conclusion

In order to sum up, this research explores the Adult census dataset and highlights the need to prioritise fairness issues above bias detection and reduction in Machine Learning models. The study attempts to reduce biases and improve model performance by closely studying the demographic data from the 1994 Census, resolving missing values, analysing data skewness, and investigating correlations between different attributes. A thorough knowledge of how biases and fairness affect model performance is aided by the insights gathered, which include histograms showing data skew, comparisons of education level and income links, and an investigation of gender distribution across marital status. This investigation emphasises how important it is to take fairness and bias into account in order to guarantee accurate and fair machine learning predictions.

Task-3 Interpretability of ML Model

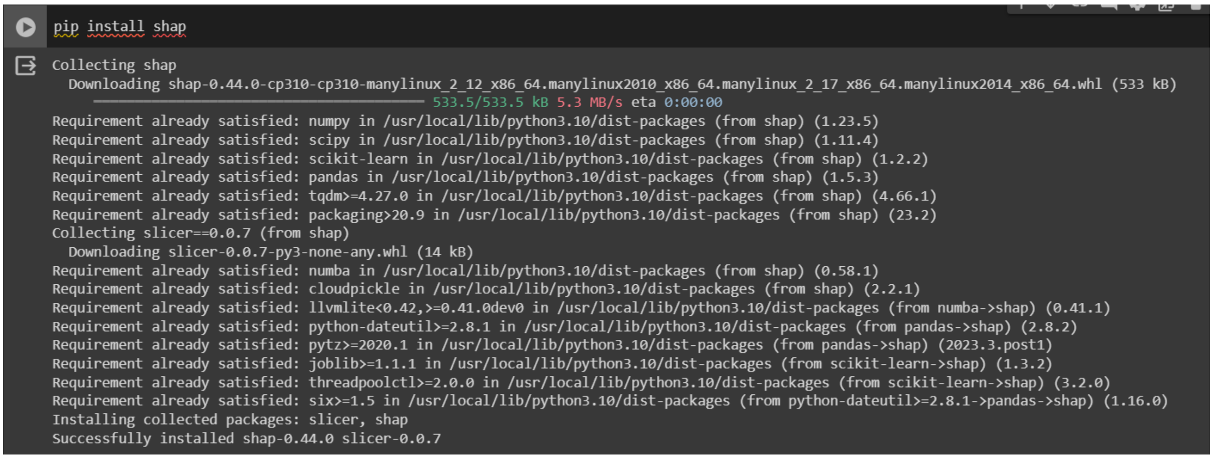
3.1 Abstract

This study splits the UCI Heart Disease dataset for testing and utilises a basic logistic regression model to predict heart disease. SHAP values are used to determine which factors are important. The impact of each element on predictions is displayed through visual summaries and feature significance scores. It's simple to see what's essential using these graphs. We can gain further insight into the model's operation by examining these outcomes. This aids in the understanding of key variables by medical professionals and researchers in the prediction of heart disease. These results simplify and provide value for all those involved in cardiovascular health by offering insightful information.

3.2 Analysis

3.2.1 Shap installation:

In machine learning, the SHAP (SHapley Additive exPlanations) approach is used to explain a model's output by determining how each feature contributes to a particular prediction. Shapley ideals from cooperative game theory serve as its foundation. In order to assist comprehend the influence of different characteristics on model predictions, SHAP values offer a means of equitably distributing the "value" of each feature among all potential combinations. Gaining an understanding of a model's decision-making process, recognising key characteristics, and enhancing transparency are all made possible by this interpretability tool. Transparency is essential for fostering confidence in machine learning models, particularly in delicate fields like healthcare.



```
pip install shap
```

```
Collecting shap
  Downloading shap-0.44.0-cp310-cp310-manylinux_2_12_x86_64.manylinux2010_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl (533 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 533.5/533.5 kB 5.3 MB/s eta 0:00:00
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from shap) (1.23.5)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from shap) (1.11.4)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (from shap) (1.2.2)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from shap) (1.5.3)
Requirement already satisfied: tqdm>=4.27.0 in /usr/local/lib/python3.10/dist-packages (from shap) (4.66.1)
Requirement already satisfied: packaging>20.9 in /usr/local/lib/python3.10/dist-packages (from shap) (23.2)
Collecting slicer==0.0.7 (from shap)
  Downloading slicer-0.0.7-py3-none-any.whl (14 kB)
Requirement already satisfied: numba in /usr/local/lib/python3.10/dist-packages (from shap) (0.58.1)
Requirement already satisfied: cloudpickle in /usr/local/lib/python3.10/dist-packages (from shap) (2.2.1)
Requirement already satisfied: llvmlite<0.42,>=0.41.0dev0 in /usr/local/lib/python3.10/dist-packages (from numba->shap) (0.41.1)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas->shap) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas->shap) (2023.3.post1)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn->shap) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn->shap) (3.2.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.1->pandas->shap) (1.16.0)
Installing collected packages: slicer, shap
Successfully installed shap-0.44.0 slicer-0.0.7
```

Fig 3.2.1 SHAP installation

The next step is to import all the necessary packages and importing the dataset. The dataset url and all the features are given and stored in a dataframe called data. And the basic information about the data is displayed.

3.2.2 Preprocessing the data:

In the dataset all the “?” are replaced by nan values so that the model can know the missing values which will be easy for us to remove as missing values might affect the performance of the model. And then the model is made to drop all the nana values as they are not actually essential for our analysis. Next the data is split into train and test dataframes to perform further analysis. The testing and training variables are X_train,X_test, y_train and y_test

The X_train and X_test will have all the features except the target variable which is “target” in this dataset and the y_train and y_test will have the target feature.

3.3.3 Standardize the features:

Scaling numerical features to have a mean of 0 and a standard deviation of 1 is known as "standardising features," and it is a typical machine learning preprocessing step. Another name for this procedure is Z-score normalisation. A few causes behind this:

1. Equal Scale: This guarantees that each feature makes an equal contribution to the analysis.

2. Faster Convergence: It facilitates the quicker and more effective search for the optimal solution by optimisation algorithms.

3. Interpretability: It facilitates comprehension of the significance of every characteristic in a model.

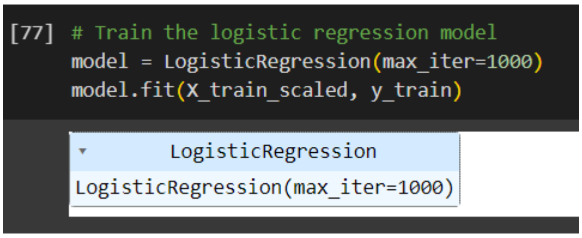
4. Consistent Regularisation: It makes the application of penalties by regularisation approaches more uniform across all characteristics.

5. Outlier Impact Reduction: This technique makes the model less susceptible to outliers by reducing the influence of extreme values.

6. Algorithm Assumptions: Support vector machines and principal component analysis are two examples of algorithms that perform better when features are standardised.

Train the logistic regression model:

Using a dataset, one may train a logistic regression model by determining the ideal weights (coefficients) for the features and the bias term. A statistical technique called logistic regression is applied to binary classification situations in which the response variable or outcome is categorical and contains two classes. Logistic regression is not used for regression; rather, it is used for classification.



```
[77] # Train the logistic regression model
     model = LogisticRegression(max_iter=1000)
     model.fit(X_train_scaled, y_train)

      ▾ LogisticRegression
     LogisticRegression(max_iter=1000)
```

Fig 3.3.2 Logistic Regression

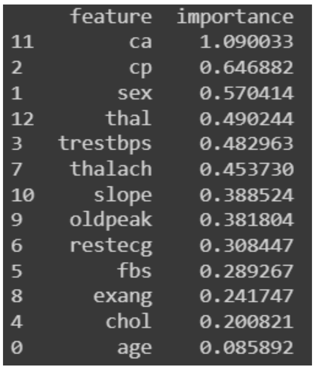
3.3.4 SHAP:

3.3.4.1 Calculate Shap values

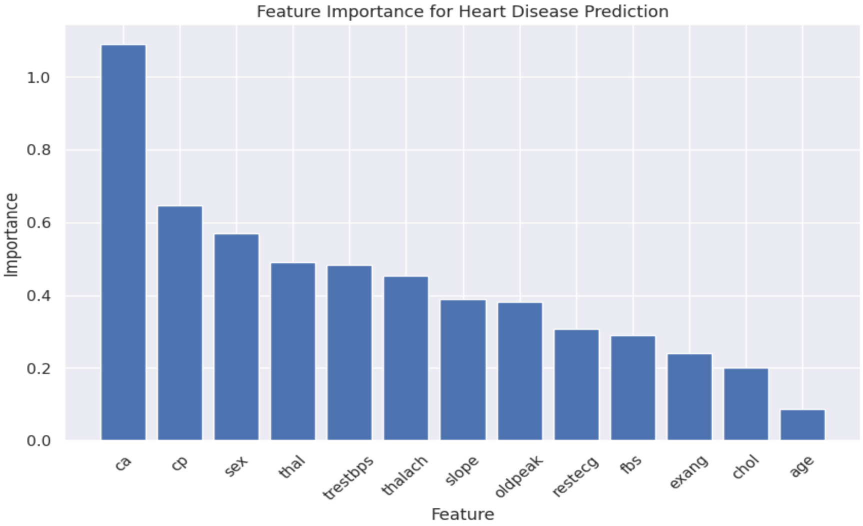
To determine how each feature affects a machine learning model's prediction, SHAP values are computed. These data shed light on the process by which the model determines a certain forecast for a particular occurrence. SHAP values enhances the interpretability, increases the feature importance, gives a better understanding of the model, address the black box nature of models , debugs the model and validates it.

3.3.4.2 Calculate feature importance:

Determining the significance of a feature is essential for comprehending the influence and input of various characteristics inside a machine learning model. It helps in understanding the model behaviour, debugs the model and guides in selecting the feature which is relevant for the analysis.



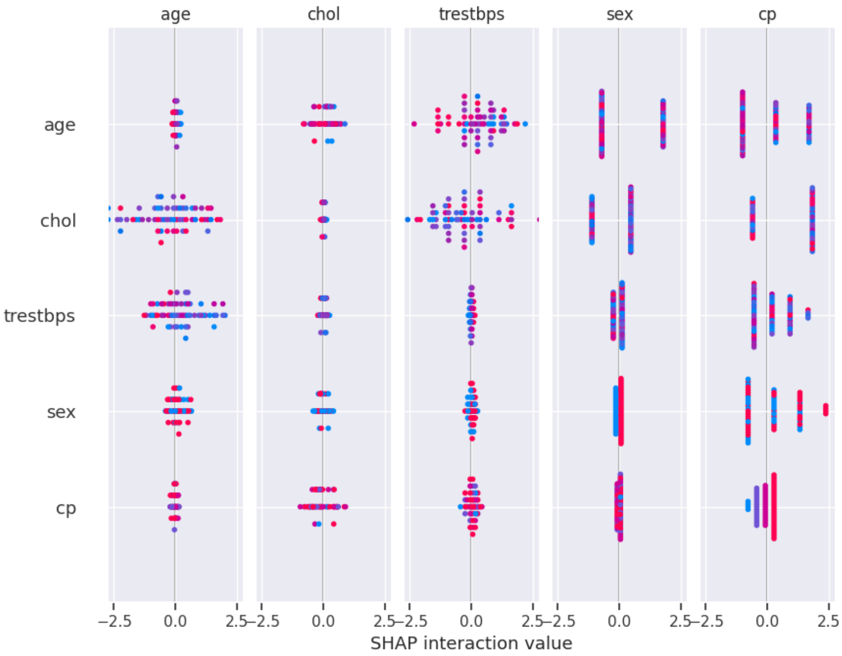
```
     feature  importance
11        ca    1.090033
2         cp    0.646882
1        sex    0.570414
12      thal    0.490244
3   trestbps    0.482963
7    thalach    0.453730
10     slope    0.388524
9    oldpeak    0.381804
6    restecg    0.308447
5        fbs    0.289267
8      exang    0.241747
4       chol    0.200821
0        age    0.085892
```



3.3.4.2.1 Feature importance

The relative value of several characteristics in a machine learning model for heart disease prediction is shown by the feature importance analysis. With a feature relevance score of 1.090033, "ca"—the number of main vessels coloured by fluoroscopy—emerges as the most significant characteristic among those examined. With a feature value of 0.646882, the "cp" feature, which denotes the kind of chest pain, comes in second place and makes a major contribution to the model's predictions. Other significant characteristics include "trestbps" (resting blood pressure), "thal" (thalassemia), and "sex" (gender), each of which has a unique impact on the model's results. These feature significance insights offer helpful direction for comprehending the main forces influencing the model's predictions, supporting interpretability, decision-making, and pointing out possible directions for more research or development. It is clear that several clinical indicators—like the number of main arteries and the kind of chest pain—carry a significant amount of weight when determining the risk of heart disease, highlighting their significance in the prognosis of cardiovascular health.

3.3.4.3 SHAP Summary plot:



It creates a summary plot for a machine learning model using the SHAP library. The summary graphic provides a thorough overview of feature influences on model outputs over several instances by utilising a collection of scaled test data (X_test_scaled) and SHAP values, which quantify each feature's contribution to model predictions. Labelling the features in the plot is made easier using the optional feature_names argument. The visual representation, which is usually in the form of a horizontal bar chart, is created using the shap.summary_plot function. Each bar in this figure represents a feature, and its length shows the average magnitude of SHAP values for that feature over all test cases. Each bar's colour indicates whether the attribute has a favourable or negative influence on the model's predictions. For activities like feature significance analysis, debugging the model, and communicating the results to non-technical stakeholders, this summary graphic is a useful tool for analysing and elucidating the behaviour of the machine learning model.