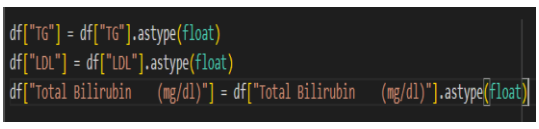
Data Collection and Preprocessing Phase

Date	20th June 2024
Team ID	SWTID1720080161
Project Title	Revolutionizing Liver Care : Predicting Liver Cirrhosis Using Advanced Machine Learning Techniques
Maximum Marks	2 Marks

Data Quality Report:

The Data Quality Report will summarize data quality issues from the selected source, including severity levels and resolution plans. It will aid in systematically identifying and rectifying data discrepancies.

Data Source	Data Quality Issue	Severity	Resolution Plan
Kaggle	• Too many NULL values	Moderate	Filling the Numerical Columns with Mean and removing abnormalities from categorical column.
Kaggle	• Wrong Data Type	Moderate	Changing the data type by type casting. Ex- 

```
df["TG"] = df["TG"].astype(float)
df["LDL"] = df["LDL"].astype(float)
df["Total Bilirubin    (mg/dl)"] = df["Total Bilirubin    (mg/dl)"].astype(float)
```

<table>
<tr>
<td>Kaggle</td>
<td>

- Ambiguous string entries in multiple column

</td>
<td>Low</td>
<td>Dropping the below rows

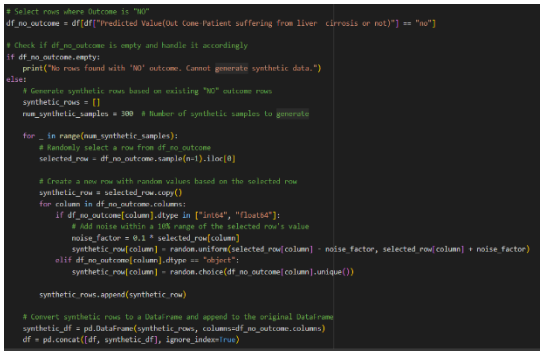
```
df = df[df['TG'] != '130LDL']
df = df[df['LDL'] != 'HDL']
df = df[df['Total Bilirubin    (mg/dl)'] != 'o.4']
```

</td>
</tr>
<tr>
<td>Kaggle</td>
<td>

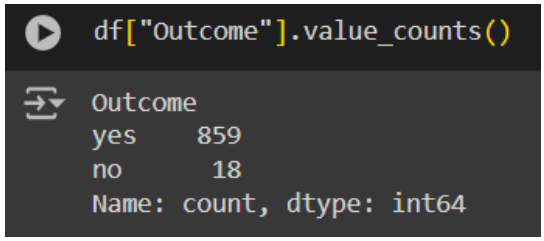
- Highly imbalanced outcome

</td>
<td>Moderate</td>
<td>

- Synthetically generating 300 rows with minority class



- Checking the value counts.



```
df["Outcome"].value_counts()
```

```
Outcome
yes    859
no      18
Name: count, dtype: int64
```

- Filling the null values with MODE.

```
df["Outcome"].fillna("yes", inplace=True)
```

</td>
</tr>
<tr>
<td>Kaggle</td>
<td>

- Many unimportant Features

</td>
<td>Moderate</td>
<td>

- Removing the features by analyzing the importance scores.

</td>
</tr>
</table>

```
from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier(n_estimators=100)
model.fit(X, y)
importances = model.feature_importances_

# Print feature importances
for feature, importance in zip(X.columns, importances):
    print(f"{feature}: {importance:.4f}")
```

```
Age: 0.0006
Gender: 0.0000
Duration of alcohol consumption(years): 0.1940
Quantity of alcohol consumption (quarters/day): 0.0206
Type of alcohol consumed: 0.0000
Hepatitis B infection: 0.0000
Hepatitis C infection: 0.0000
Diabetes Result: 0.0044
Blood pressure (mmhg): 0.0001
Obesity: 0.0000
Family history of cirrhosis/ hereditary: 0.0001
TCH: 0.0001
TG: 0.0001
LDL: 0.0002
HDL: 0.0003
Hemoglobin  (g/dl): 0.0011
PCV  (%): 0.0007
RBC  (million cells/microliter): 0.0282
MCV  (femtoliters/cell): 0.0007
MCH  (picograms/cell): 0.0194
MCHC  (grams/deciliter): 0.0534
Total Count: 0.0010
Polymorphs  (%) : 0.0104
Lymphocytes  (%): 0.0058
Monocytes  (%): 0.0025
Eosinophils  (%): 0.0000
Basophils  (%): 0.0074
Platelet Count  (lakhs/mm): 0.0203
Total Bilirubin  (mg/dl): 0.1604
Direct  (mg/dl): 0.1125
Indirect  (mg/dl): 0.0092
Total Protein  (g/dl): 0.0024
Albumin  (g/dl): 0.0800
Globulin  (g/dl): 0.0003
A/G Ratio: 0.0518
AL.Phosphatase  (U/L): 0.0204
SGOT/AST  (U/L): 0.0199
SGPT/ALT (U/L): 0.0114
USG Abdomen (diffuse liver or  not): 0.1605
```

- Dropping all the unnecessary columns.

```
for col in drop_col:
  if col in X.columns:
    X.drop(columns=[col],inplace=True)
```