# Assignment 1 - Question #4

October 12, 2020

## 1 Importing Libraries

[16]:
```
import pandas as pd
pd.plotting.register_matplotlib_converters()
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from sklearn import tree
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import cross_val_score
from sklearn.neighbors import KNeighborsClassifier
print("Setup Complete")
```

```
Setup Complete
```

## 2 Read data

### 2.1 Specify path of dataset

[17]:
```
my_path = './hcvdat0.csv'
```

[18]:
```
hcv_data = pd.read_csv(my_path)
hcv_data.head()
```

[18]:

| | 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 |

### 2.2 Make sure class column is the last column in data-frame

[19]:
```
cols = hcv_data.columns.tolist()
cols.remove('Category')
cols.append('Category')
hcv_data = hcv_data[cols]
hcv_data
```

[19]:

| | Unnamed: 0 | Age | Sex | ALB | ALP | ALT | AST | BIL | CHE | CHOL | CREA | GGT | PROT | Category |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 32 | m | 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 | m | 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 | m | 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 | m | 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 | m | 39.2 | 74.1 | 32.6 | 24.8 | 9.6 | 9.15 | 4.32 | 76.0 | 29.9 | 68.7 | 0=Blood Donor |
| .. | … | … | .. | … | … | … | … | … | … | … | … | … | … | … |
| 610 | 611 | 62 | f | 32.0 | 416.6 | 5.9 | 110.3 | 50.0 | 5.57 | 6.30 | 55.7 | 650.9 | 68.5 | 3=Cirrhosis |
| 611 | 612 | 64 | f | 24.0 | 102.8 | 2.9 | 44.4 | 20.0 | 1.54 | 3.02 | 63.0 | 35.9 | 71.3 | 3=Cirrhosis |
| 612 | 613 | 64 | f | 29.0 | 87.3 | 3.5 | 99.0 | 48.0 | 1.66 | 3.63 | 66.7 | 64.2 | 82.0 | 3=Cirrhosis |
| 613 | 614 | 46 | f | 33.0 | NaN | 39.0 | 62.0 | 20.0 | 3.56 | 4.20 | 52.0 | 50.0 | 71.0 | 3=Cirrhosis |
| 614 | 615 | 59 | f | 36.0 | NaN | 100.0 | 80.0 | 12.0 | 9.07 | 5.30 | 67.0 | 34.0 | 68.0 | 3=Cirrhosis |

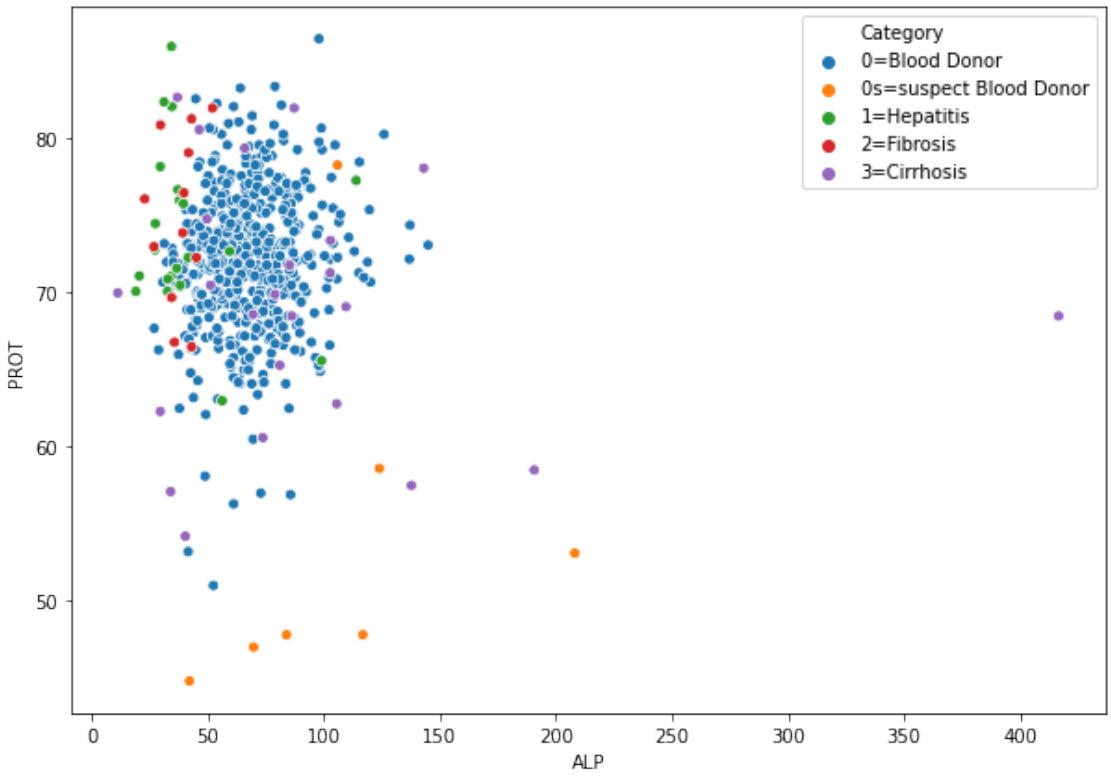
[615 rows x 14 columns]

# 3 Scatter Plots

## 3.1 ALP-PROT

We can see most of our data members have '0=Blood Donor' class. Also members of '2=Fibrosis' and '1=Hepatitis' class, mostly have 'ALP' less than 50

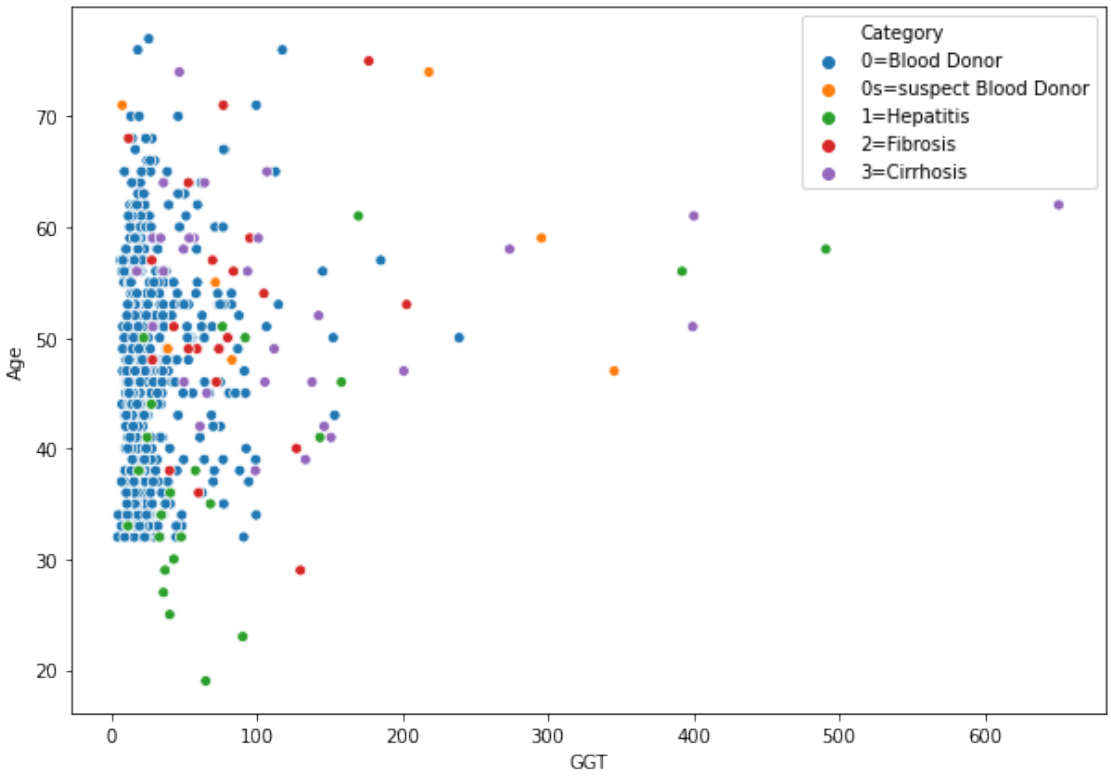
[20]:
```
plt.figure(figsize=(10,7))
sns.scatterplot(x=hcv_data['ALP'], y=hcv_data['PROT'], hue=hcv_data['Category'])
plt.show()
```




### 3.2 ALP-CREA

We can see, classes are distributed evenly, but ages between 30-60, hold hold most of blood donor counts.

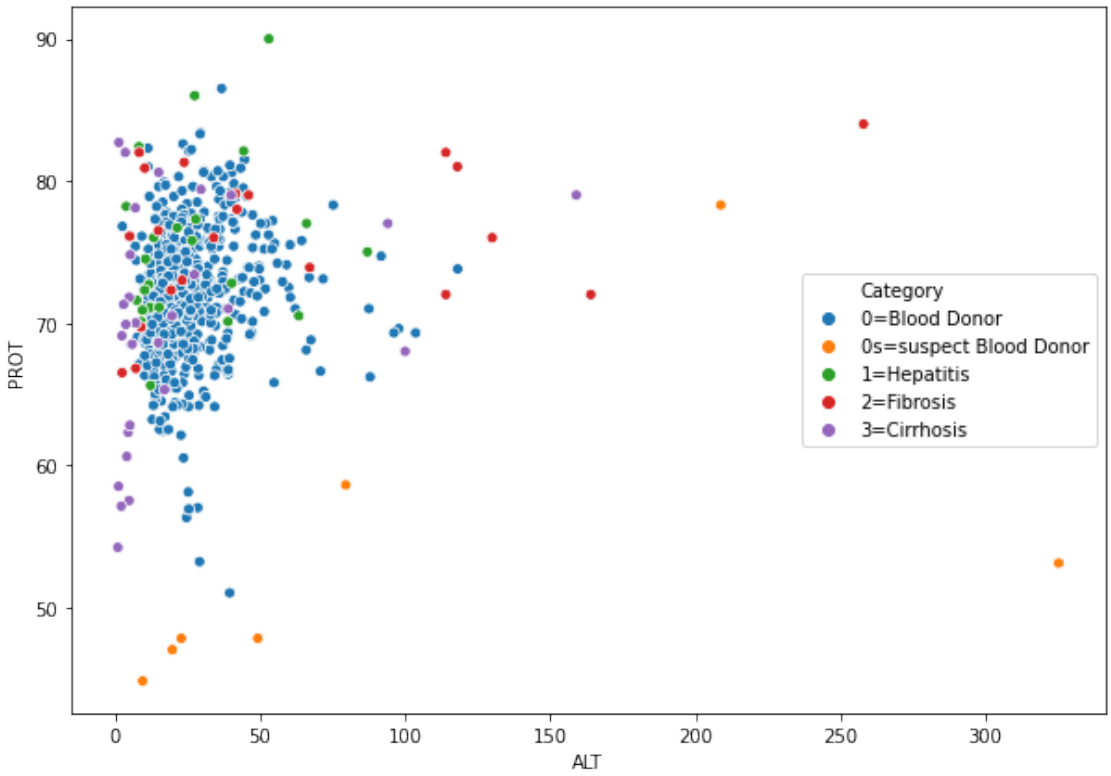
[21]:
```
plt.figure(figsize=(10,7))
sns.scatterplot(x=hcv_data['GGT'], y=hcv_data['Age'], hue=hcv_data['Category'])
plt.show()
```




### 3.3 ALT-PROT

Data members of class '3=Cirrhosis' have less 'ALT' values than others.

[22]:
```
plt.figure(figsize=(10,7))
sns.scatterplot(x=hcv_data['ALT'], y=hcv_data['PROT'], hue=hcv_data['Category'])
plt.show()
```




# 4 Decsion Tree

## 4.1 10-fold validation

Since last fold, has less score than others, could suspect that are model has been over-fitted by our data. Another cause of first and last fold scores, being less than others, might be outliers.

[23]:
```
# Drop rows containing NAN
hcv_data = hcv_data.dropna(axis = 0)
x = hcv_data[hcv_data.columns[:-1]]
y = hcv_data['Category']
# Remove 'Sex', because it's categorical
x = x.drop('Sex', axis=1)
clf = tree.DecisionTreeClassifier()
# Perform 10-fold cross validation
scores = cross_val_score(estimator=clf, X=x, y=y, cv=10, n_jobs=4)
print(scores)
```

```
[0.96610169 1.         1.         1.         1.         1.
 0.98305085 1.         1.         0.53448276]
```

```
/home/raycatcher/.local/lib/python3.8/site-
packages/sklearn/model_selection/_split.py:665: UserWarning: The least populated
```

```
class in y has only 7 members, which is less than n_splits=10.
  warnings.warn(("The least populated class in y has only %d"
```

# 5 Preprocessing

## 5.1 Find categorical variables

[24]:
```
# Get list of categorical variables
s = (hcv_data.dtypes == 'object')
object_cols = list(s[s].index)

print("Categorical variables:")
print(object_cols)
```

```
Categorical variables:
['Sex', 'Category']
```

## 5.2 Label encode categorical variables

[25]:
```
from sklearn.preprocessing import LabelEncoder

# Make copy to avoid changing original data
hcv_numerical = hcv_data.copy()

# Apply label encoder to each column with categorical data
label_encoder = LabelEncoder()
for col in object_cols:
    hcv_numerical[col] = label_encoder.fit_transform(hcv_data[col])
hcv_numerical
```

[25]:

| | Unnamed: 0 | Age | Sex | ALB | ALP | ALT | AST | BIL | CHE | CHOL | CREA | GGT | PROT | Category |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |
| 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 |
| 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 |
| 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 |
| 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 |
| .. | … | … | … | … | … | … | … | … | … | … | … | … | … | … |
| 608 | 609 | 58 | 0 | 34.0 | 46.4 | 15.0 | 150.0 | 8.0 | 6.26 | 3.98 | 56.0 | 12.0 | 71.3 | 3 |
| 609 | 610 | 59 | 0 | 39.0 | 51.3 | 19.6 | 285.8 | 40.0 | 5.77 | 4.51 | 136.1 | 19.7 | 70.0 | 3 |
| 610 | 611 | 62 | 0 | 32.0 | 416.6 | 5.9 | 110.3 | 50.0 | 5.57 | 6.30 | 55.7 | 650.9 | 68.5 | 3 |
| 611 | 612 | 64 | 0 | 24.0 | 102.8 | 2.9 | 44.4 | 20.0 | 1.54 | 3.02 | 63.0 | 35.9 | 71.3 | 3 |
| 612 | 613 | 64 | 0 | 29.0 | 87.3 | 3.5 | 99.0 | 48.0 | 1.66 | 3.63 | 66.7 | 64.2 | 82.0 | 3 |

```
3     33.8  75.7         0
4     29.9  68.7         0
..     …   …       …
608   49.7  80.6         4
609  101.1  70.5         4
610  650.9  68.5         4
611   35.9  71.3         4
612   64.2  82.0         4

[589 rows x 14 columns]
```

### 5.3 Normalize data columns

[26]:
```
from sklearn import preprocessing

x_data = hcv_numerical.values #returns a numpy array
min_max_scaler = preprocessing.MinMaxScaler()
x_scaled = min_max_scaler.fit_transform(x_data)
df = pd.DataFrame(x_scaled)
df.columns = hcv_numerical.columns
df.head()
```

[26]:

| | Unnamed: 0 | Age | Sex | ALB | ALP | ALT | AST | \ |
|---|---|---|---|---|---|---|---|---|
| 0 | 0.000000 | 0.166667 | 1.0 | 0.350669 | 0.101653 | 0.020962 | 0.036694 | |
| 1 | 0.001634 | 0.166667 | 1.0 | 0.350669 | 0.145571 | 0.052713 | 0.044990 | |
| 2 | 0.003268 | 0.166667 | 1.0 | 0.475483 | 0.156427 | 0.108816 | 0.134014 | |
| 3 | 0.004902 | 0.166667 | 1.0 | 0.420505 | 0.100419 | 0.091554 | 0.038290 | |
| 4 | 0.006536 | 0.166667 | 1.0 | 0.361070 | 0.154947 | 0.097719 | 0.045310 | |

| | BIL | CHE | CHOL | CREA | GGT | PROT | Category |
|---|---|---|---|---|---|---|---|
| 0 | 0.032181 | 0.367578 | 0.218447 | 0.091495 | 0.011757 | 0.580336 | 0.0 |
| 1 | 0.014890 | 0.650434 | 0.408981 | 0.061619 | 0.017172 | 0.760192 | 0.0 |
| 2 | 0.025456 | 0.494997 | 0.457524 | 0.072822 | 0.044400 | 0.827338 | 0.0 |
| 3 | 0.086936 | 0.394263 | 0.401699 | 0.067221 | 0.045328 | 0.741007 | 0.0 |
| 4 | 0.042267 | 0.515677 | 0.350728 | 0.063486 | 0.039295 | 0.573141 | 0.0 |

## 6 KNN classifier

We can see that scores of CV has improved for KNN classifier. scores are almost the same with an acceptable mean value for them.

[27]:
```
from sklearn.model_selection import cross_val_score
import numpy as np
#create a new KNN model
knn_cv = KNeighborsClassifier(n_neighbors=3)
#train model with cv of 5
```

```
cv_scores = cross_val_score(knn_cv, hcv_numerical[hcv_numerical.columns[:-1]],␣
 ↪hcv_numerical['Category'], cv=10)
#print each cv score (accuracy) and average them
print(cv_scores)
print('cv_scores mean:{}'.format(np.mean(cv_scores)))
```

```
[0.96610169 0.93220339 0.96610169 0.98305085 0.94915254 1.
 0.91525424 0.91525424 0.96610169 0.9137931 ]
cv_scores mean:0.9507013442431326
```

```
/home/raycatcher/.local/lib/python3.8/site-
packages/sklearn/model_selection/_split.py:665: UserWarning: The least populated
class in y has only 7 members, which is less than n_splits=10.
  warnings.warn(("The least populated class in y has only %d"
```