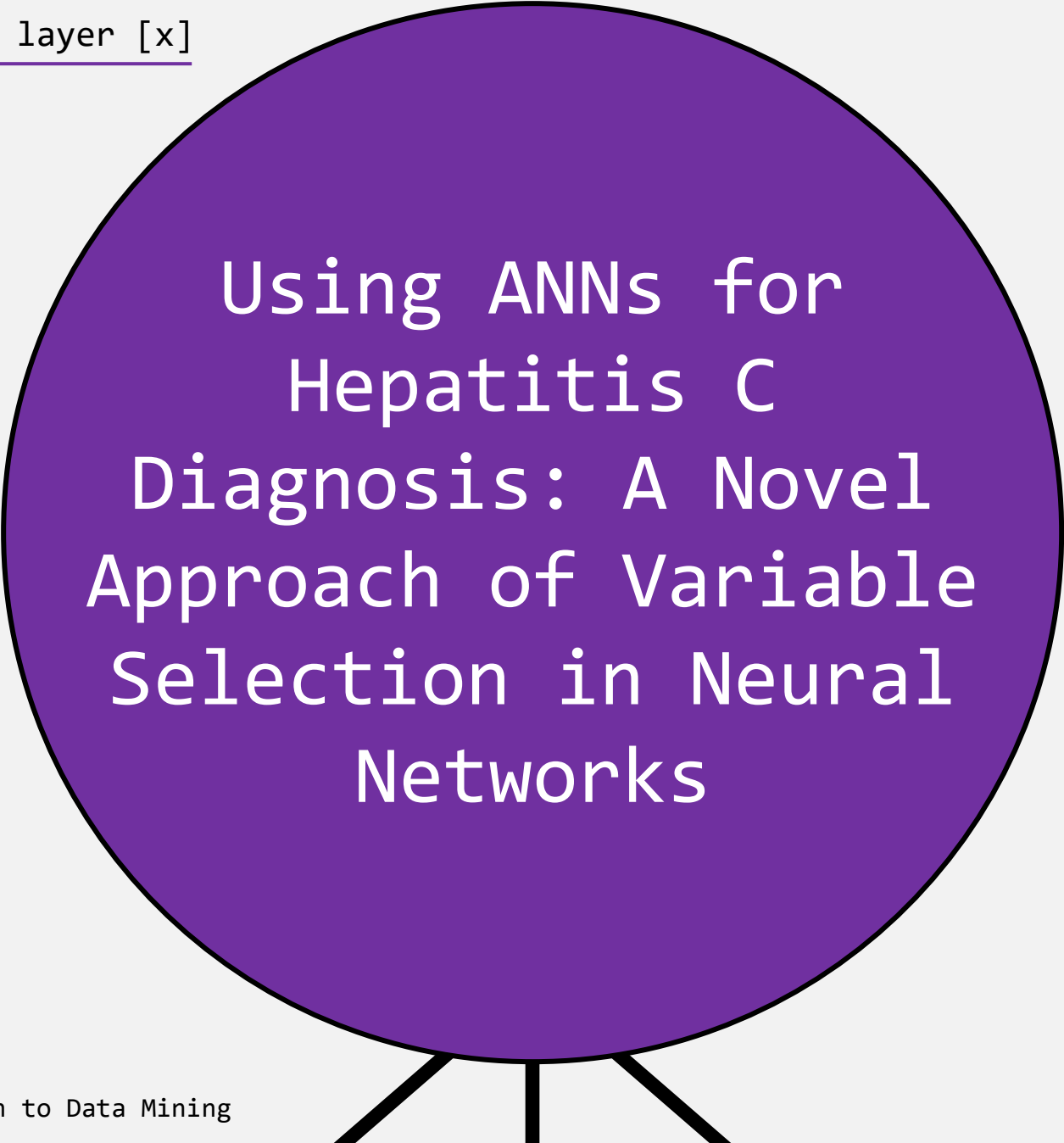


input layer [x]



Using ANNs for Hepatitis C Diagnosis: A Novel Approach of Variable Selection in Neural Networks



Outline

- Introductino
- Obras Relacionadas
- Exploración de datos
 - Transformación
 - Limpieza
- Metodología
- Resultados
- Hallazgos principales
- Conclusiones



Introduction

def(Hepatitis):

Virus that infects the liver and causes liver tissue damage through inflammations

Functions of liver:

- Production of certain proteins for blood plasma
- Production of cholesterol and special proteins to help carry fats through the body,
- Removing waste and breaking down fats in the small intestine.
- 3 main types of Hepatitis:
 - Hepatitis A
 - Hepatitis B
 - Hepatitis C
- 354 Million homeboys suffer from chronic hepatitis B and C globally
- Treatment is available (vaccine only for Hepatitis A and B) but is extremely costly

Misdiagnosis is bad



Related Works

- ANNs architecture performs the best and yields the most accurate results
- Previous literature suggests that hybrid neural networks outperform unsupervised learning methods/approaches
 - Ex for bad networks/models:
 - **Decision Trees** - minor changes in the input caused huge changes in decision tree
 - **Self Organizing Maps** - inefficient because it needed a larger training dataset to contain target set
 - Ex of good networks/arch:
 - Hybrid ANN model (**Best for real-life application**)
 - Conic Section Function Neural Network (**CSFNN** – combines *RBF* & *MLP*)
- Most previous literature works suggest that a good ANN structure can help aid doctors in Hepatitis C diagnosis but *cannot fully replace them.*



Dataset

```
df = pd.read_csv("/content/drive/MyDrive/hcvdat0 (1).csv")
```

- Contains data from laboratory results of patients that are either:
 - Blood Donors (i.e. no sign of liver tissue damage)
 - Hepatitis C patients
 - Fibrosis patients (induced Hepatitis C)
 - Cirrhosis patients (again, induced Hepatitis C)
- A total of 614 patients
- 12 variables, shown below:
 - Age
 - Sex
 - ALB (Albumin Blood Test)
 - ALP (Alkaline phosphatase)
 - ALT (Alkaline Transaminase)
 - BIL (Bilirubin)
 - CHE (Acetylcholinesterase)
 - CHOL (Cholesterol)
 - CREA (Creatinine)
 - GGT (Gamma-Glutamyl Transferase)
 - PROT (Proteins)

```
df.head()
```

	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



More on the Dataset

```
df.describe()
```

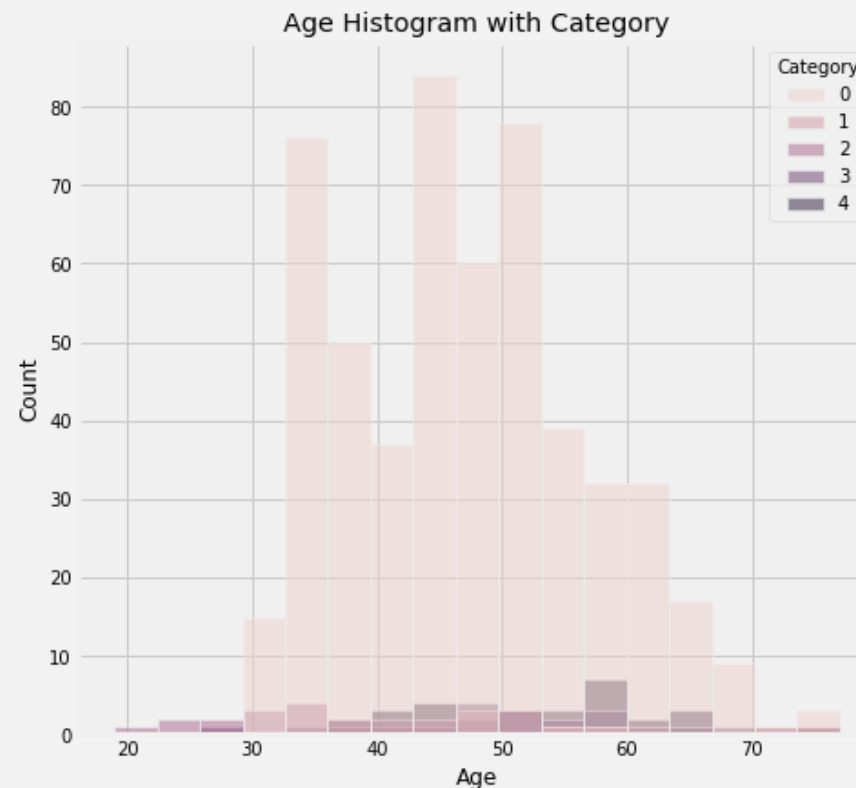
	Age	ALB	ALP	ALT	AST	BIL	CHE	CHOL	CREA	GGT	PROT
count	615.000000	615.000000	615.000000	615.000000	615.000000	615.000000	615.000000	615.000000	615.000000	615.000000	615.000000
mean	47.408130	41.620195	68.283920	28.450814	34.786341	11.396748	8.196634	5.368099	81.287805	39.533171	72.044137
std	10.055105	5.775920	25.643955	25.448940	33.090690	19.673150	2.205657	1.123466	49.756166	54.661071	5.398234
min	19.000000	14.900000	11.300000	0.900000	10.600000	0.800000	1.420000	1.430000	8.000000	4.500000	44.800000
25%	39.000000	38.800000	52.950000	16.400000	21.600000	5.300000	6.935000	4.620000	67.000000	15.700000	69.300000
50%	47.000000	41.900000	66.700000	23.000000	25.900000	7.300000	8.260000	5.310000	77.000000	23.300000	72.200000
75%	54.000000	45.200000	79.300000	33.050000	32.900000	11.200000	9.590000	6.055000	88.000000	40.200000	75.400000
max	77.000000	82.200000	416.600000	325.300000	324.000000	254.000000	16.410000	9.670000	1079.100000	650.900000	90.000000

- Issues with dataset:
 - N/A Variables need to be removed
 - Data is unbalanced (more representation for '0 = Blood Donor' than Hepatitis C, Cirrhosis, etc...)
 - Too many variables (?)



Exploring the data (Observationally)

```
sns.histplot(df[['Age', 'Category']], x='Age', hue='Category', color= ['darkorchid', 'darkviolet'])  
plt.show()
```

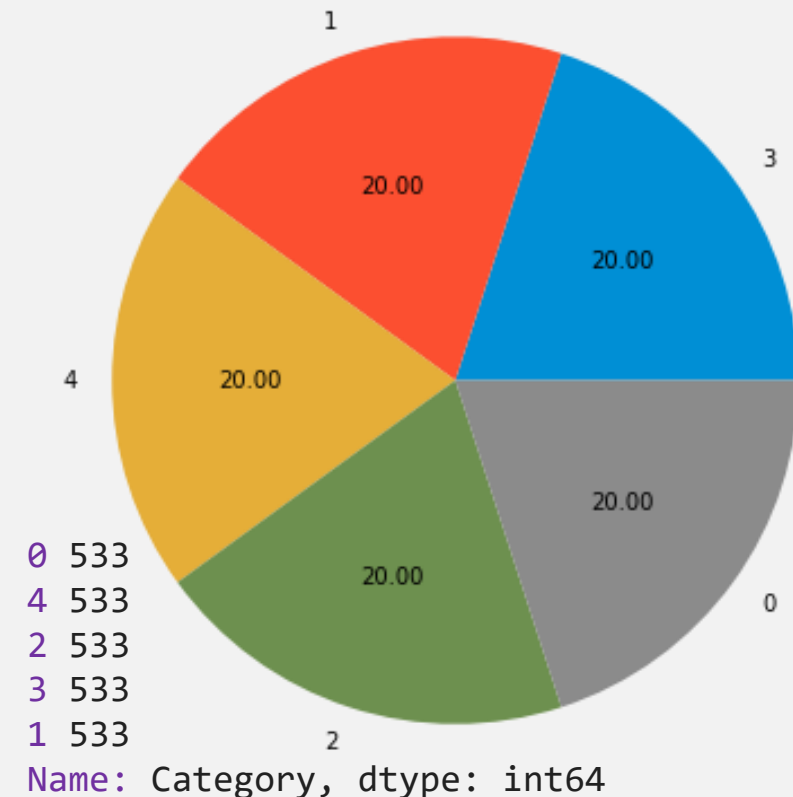
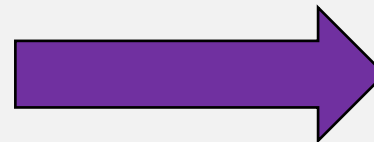
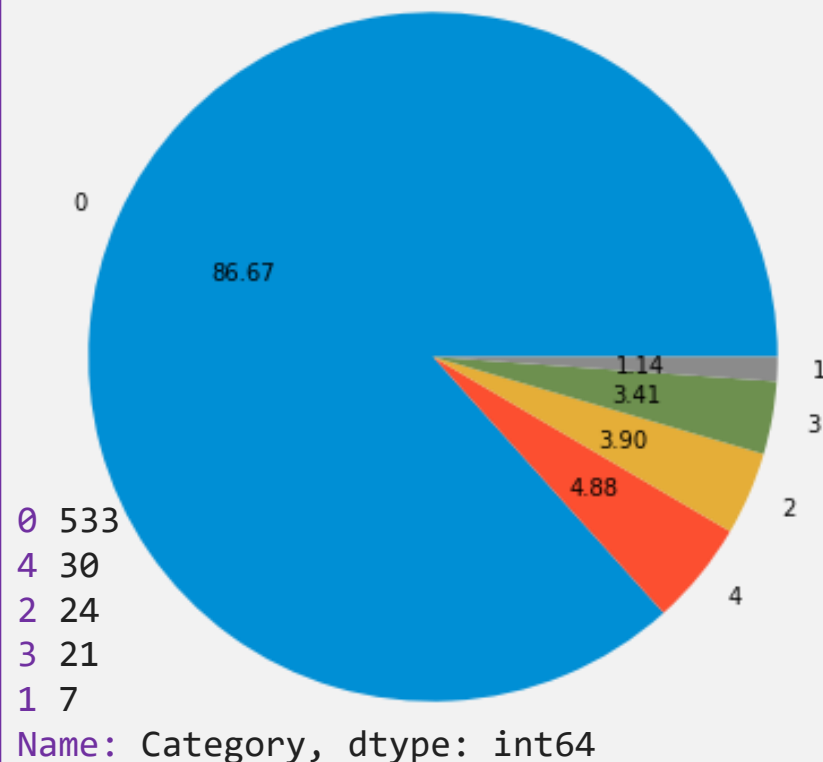


Is there evidence of a correlation between the age of a patient and them suffering from any of the diseases?



Imbalanced Dataset

- Pie chart shows that the dataset is severely imbalanced – Category “0 = Blood Donor” takes up almost 87% (!)
- Solution: Using rebalancing methods (In our case, SMOTE)





Synthetic Minority Oversampling Technique (SMOTE)

- Oversampling: Taking small set of data and re-distributing artificially so as to increase the number of datapoints in a category
- Very effective when considering neural networks (imbalanced data is bad, rebalancing = 😊)
- SMOTE: Works through the K-nearest Neighbors algorithm.
 - Uses K-nearest neighbors to a point, synthetically creates a point between those points (within same set)
 - Ensures distance between the points within a set (in addition to the synthetic datapoints) and other sets are maintained;
 - Stops when we have the same number of points as the largest set



Methodology

- Refresher: Neural Networks method used for diagnosis of Hepatitis C (and other liver tissue diseases) in patients.
- Neural Networks (ANNs) using sequential (feed-forward) architecture, experimentally decided on 2 dense, hidden layers, along with input and output (input layer changes based on model, output layer has 5 neurons for each of the 5 possible classes).

Model Summary →

Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
dense (Dense)	(None, 12)	156
dense_1 (Dense)	(None, 12)	156
dense_2 (Dense)	(None, 5)	65

=====

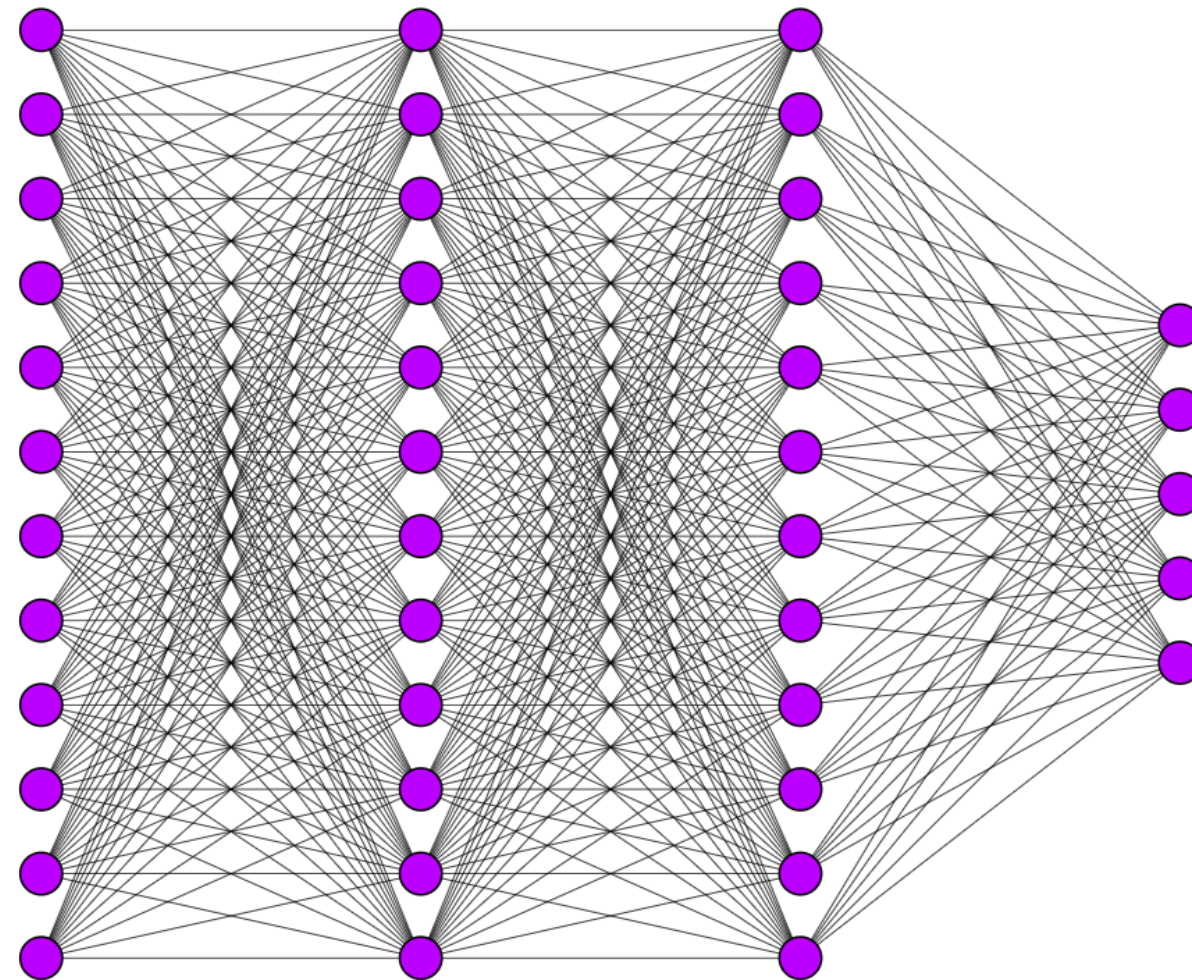
Total params: 377
Trainable params: 377
Non-trainable params: 0

=====



Cool visual Representation

- Full model contains all 12 input variables
- The other models will first require us to go through variable selection, and then reduce from this model.



Input Layer $\in \mathbb{R}^{12}$

Hidden Layer $\in \mathbb{R}^{12}$

Hidden Layer $\in \mathbb{R}^{12}$

Output Layer $\in \mathbb{R}^5$



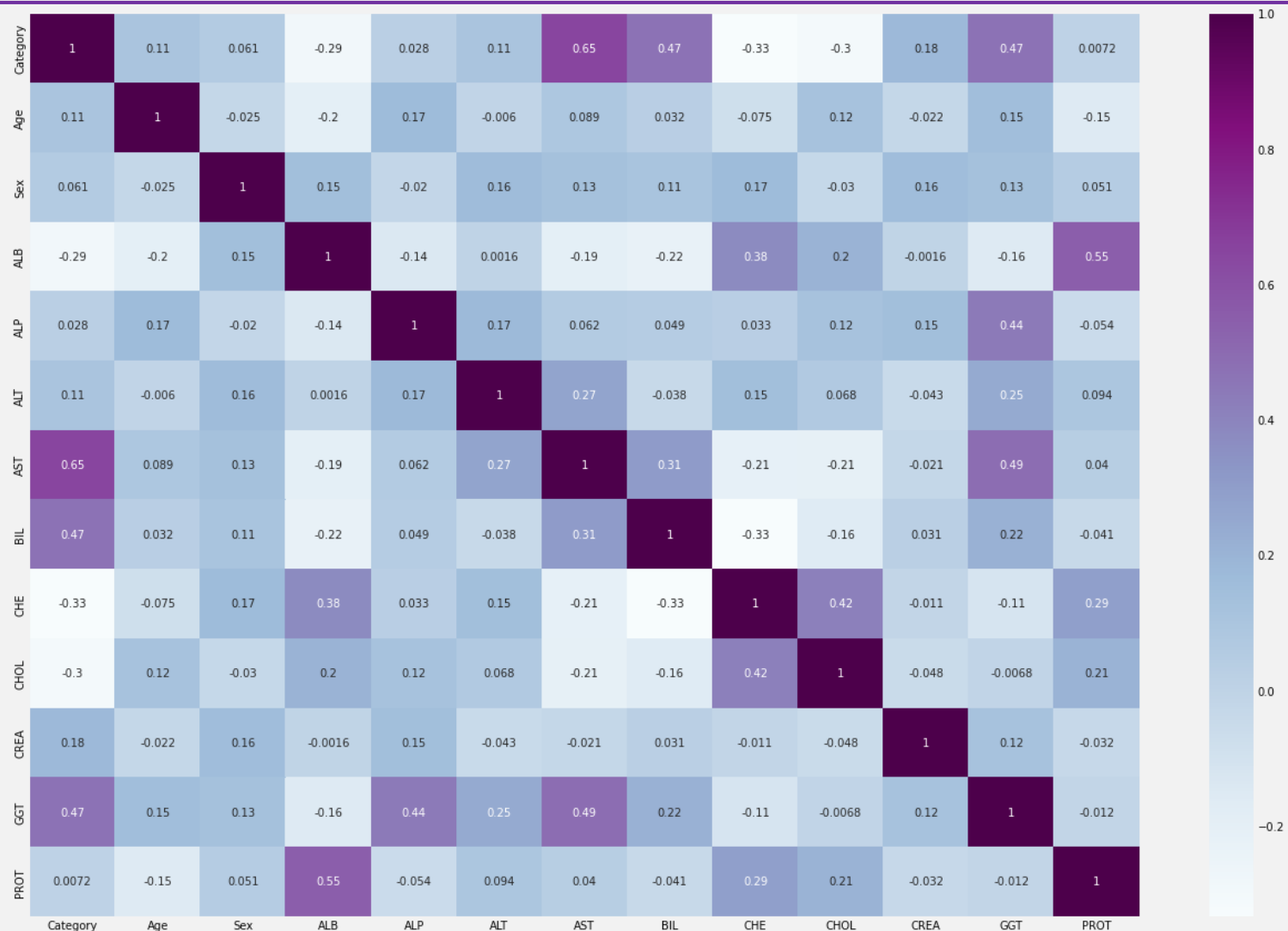
Variable Selection Tools

Correlation Matrix:

Consider the relationships between each of the variables, and base the variable selection on that.

Based on this correlation matrix, we can see that the variables most highly correlated with the **Category** are: **AST**, **BIL** and **GGT**

Thus we have the:
simplest model.





Novel Variable Selection Method

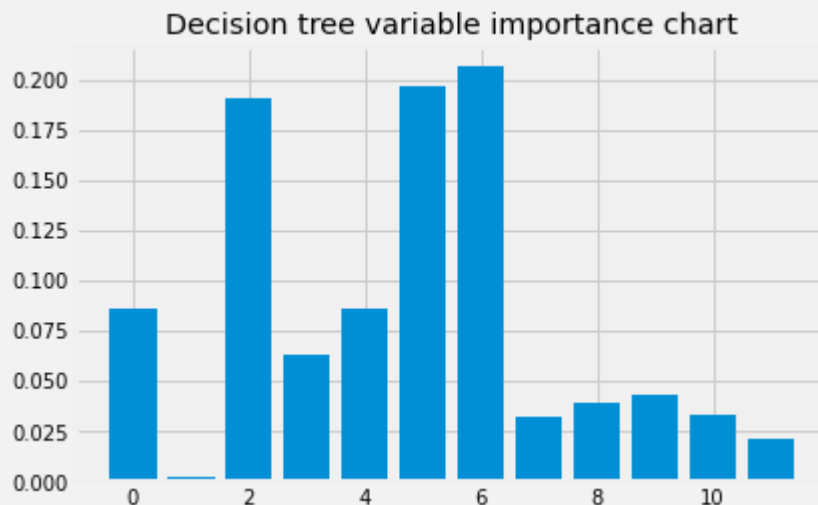
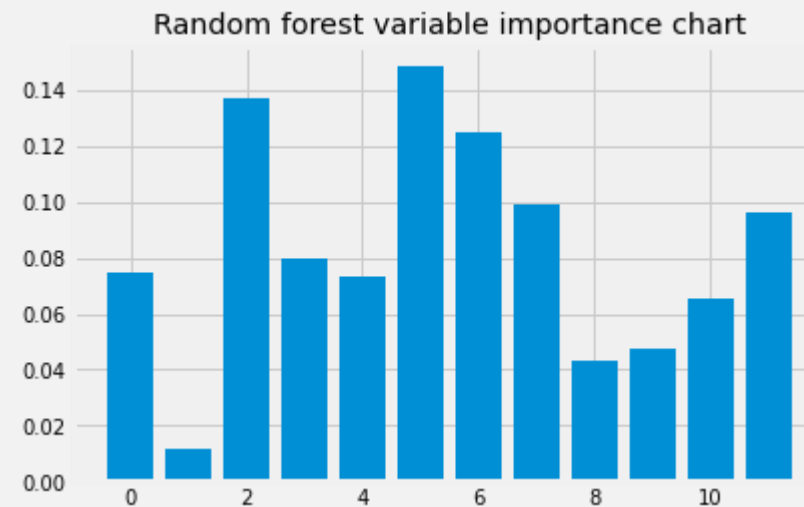
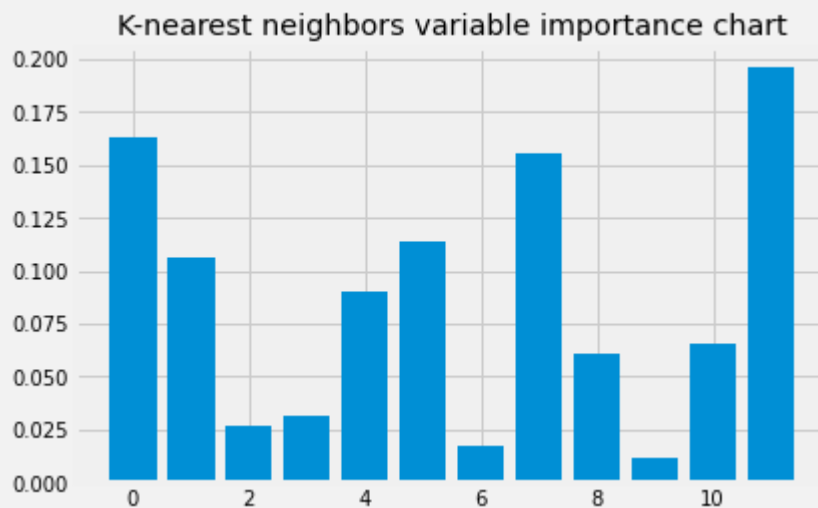
- Method incorporates previously well-established ML techniques and builds the neural networks on those
- Specifically, the models used for the prediction can use different variables based on the performance of the ML methods.
- The problem becomes less about the neural network itself and more about the pre-processing of the data → Less complicated, but still a hybrid system.

ML techniques used: K-Nearest Neighbors, Random Forest, Decision Tree (derived from the literature)

- Metric: Variable Importance (How much a variable contributes to the prediction outcome)



Variable Importance + Score



Variable	Importance Score
AGE (VAR 0)	0.10770947635466771
SEX (VAR 1)	0.03989632374876976
ALB (VAR 2)	0.11816277579941716
ALP (VAR 3)	0.05790419266882033
ALT (VAR 4)	0.08338147526141615
AST (VAR 5)	0.15291713605933596
BIL (VAR 6)	0.11608543460118827
CHE (VAR 7)	0.09546399895371337
CHOL (VAR 8)	0.04772722579894203
CREA (VAR 9)	0.033844343496945296
GGT (VAR 10)	0.054781737475094495
PROT (VAR 11)	0.1045298118639057



Final Models Chosen

- **Model 1:** Full model (all 12 variables are included)
- **Model 2:** “CREA,” “CHOL” and “GGT” are dropped from the model
- **Model 3:** Simplest model (only the variables “AST,” “BIL,” and “GGT” are included)
- **Model 4:** Age is dropped as an input variable (all predictors except “AGE”)
- **Model 5:** “ALP,” “ALT,” “GGT” and “CHOL” are dropped from the model



Results and Metrics Considered

- Model accuracy (and model loss) per unit training run
- Training data prediction (using classification matrix)
- Testing (unseen) data prediction (using classification matrix)
- Classification report (precision, recall & F1 score)

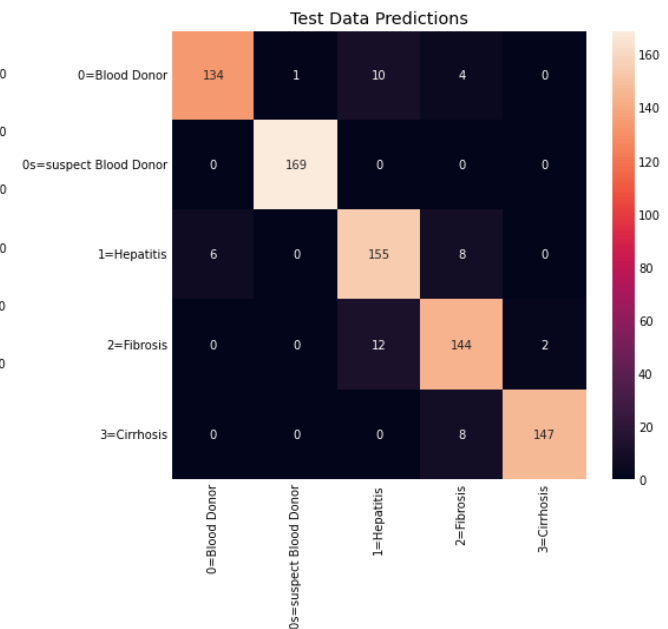
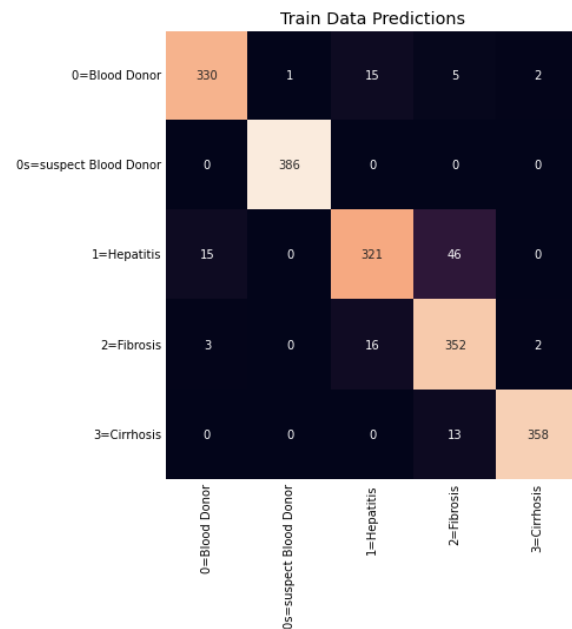
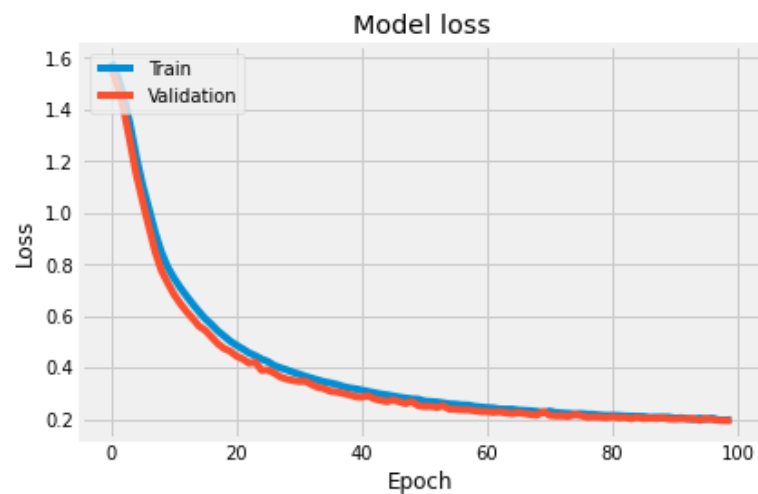
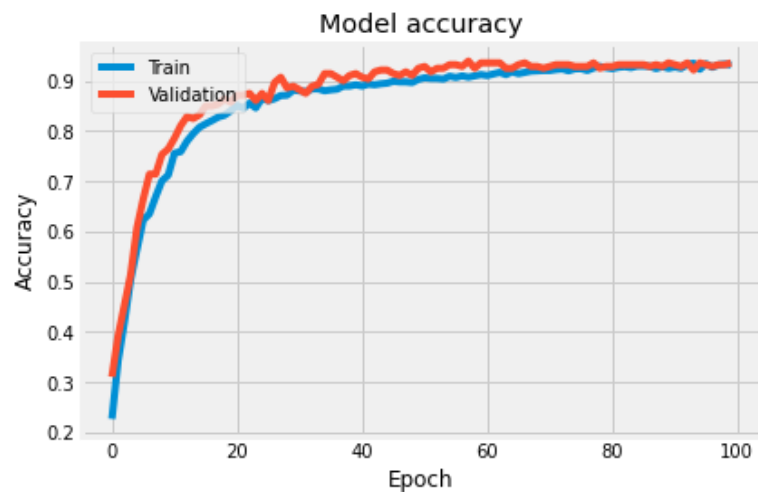
$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F1} = \frac{2(\text{Precision})(\text{Recall})}{\text{Precision} + \text{Recall}}$$



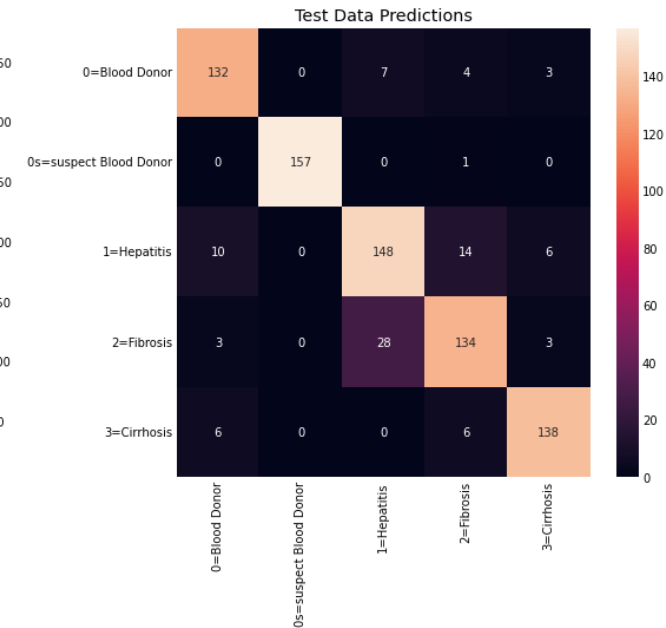
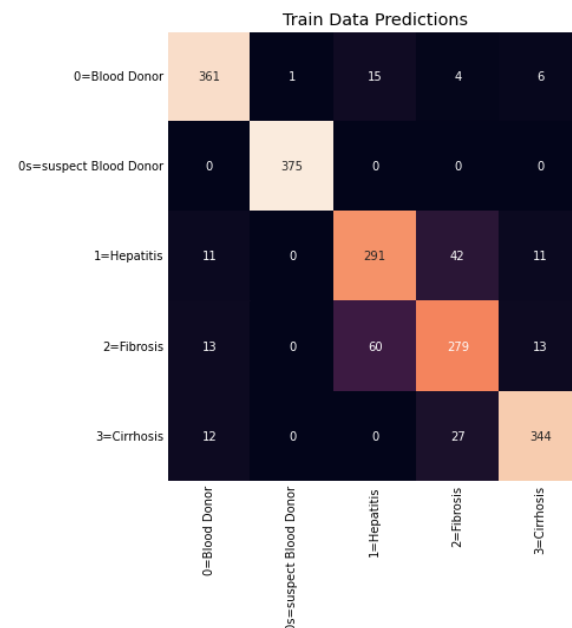
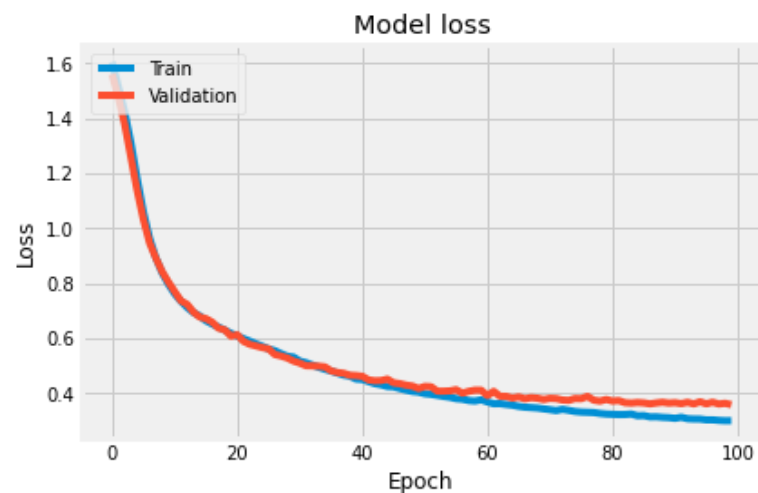
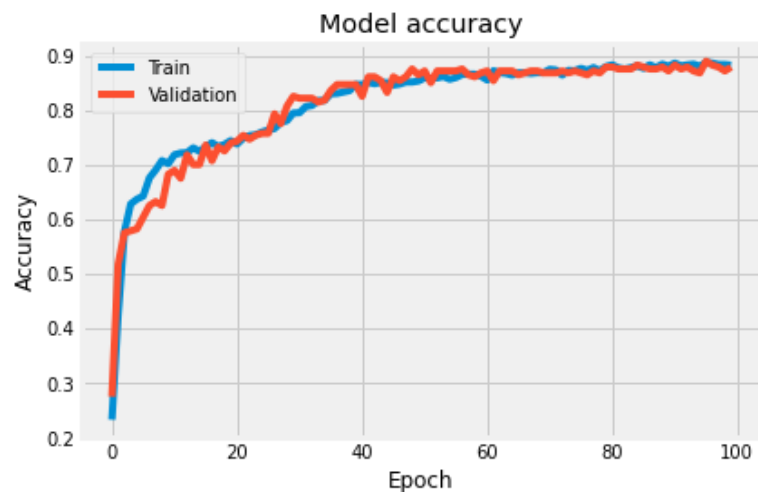
Model 1



Classification Report				
	precision	recall	f1-score	support
0	0.97	0.94	0.95	180
1	1.00	1.00	1.00	147
2	0.88	0.85	0.86	151
3	0.83	0.93	0.88	160
4	1.00	0.96	0.98	162
accuracy			0.93	800
macro avg	0.94	0.93	0.93	800
weighted avg	0.94	0.93	0.93	800



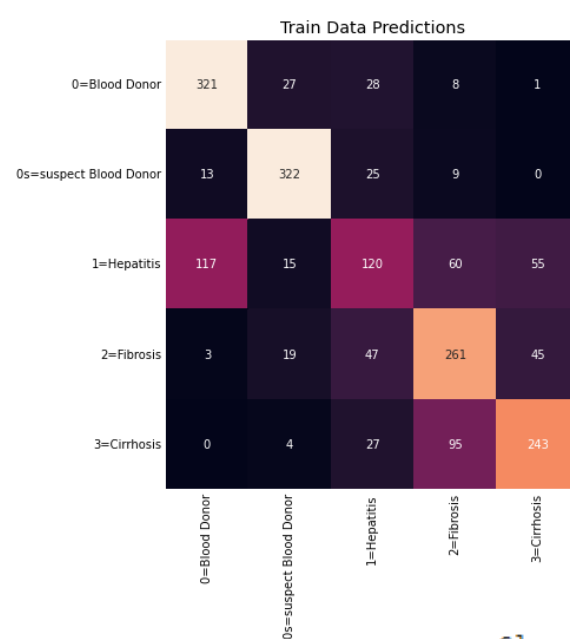
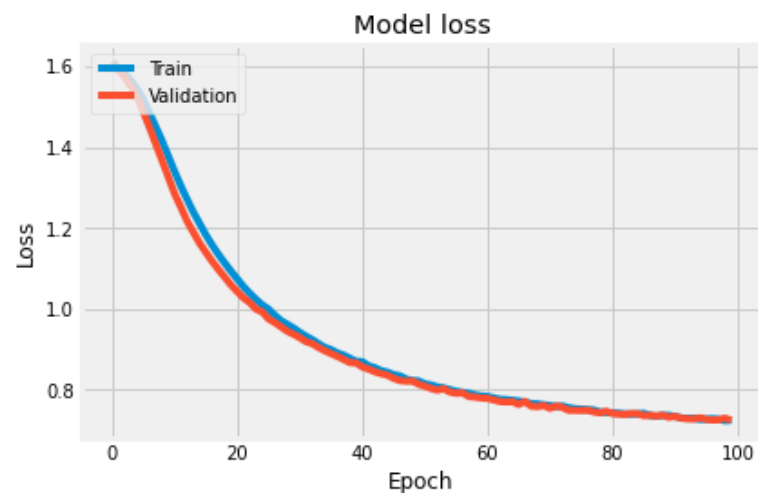
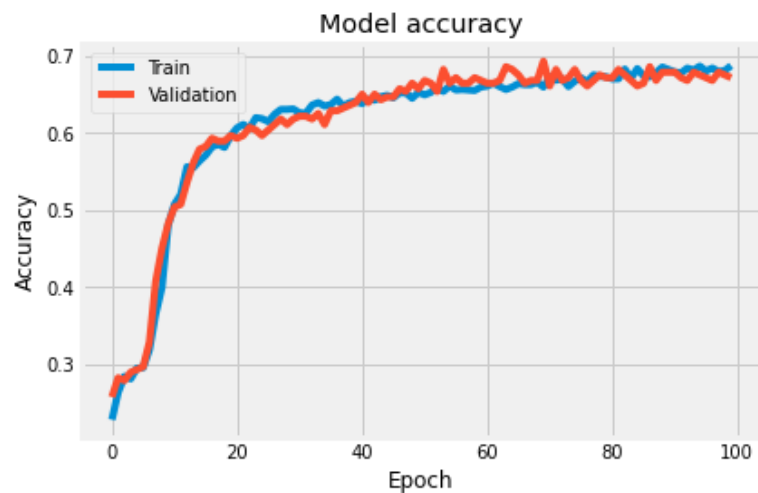
Model 2



Classification Report				
	precision	recall	f1-score	support
0	0.87	0.90	0.89	146
1	1.00	0.99	1.00	158
2	0.81	0.83	0.82	178
3	0.84	0.80	0.82	168
4	0.92	0.92	0.92	150
accuracy			0.89	800
macro avg	0.89	0.89	0.89	800
weighted avg	0.89	0.89	0.89	800



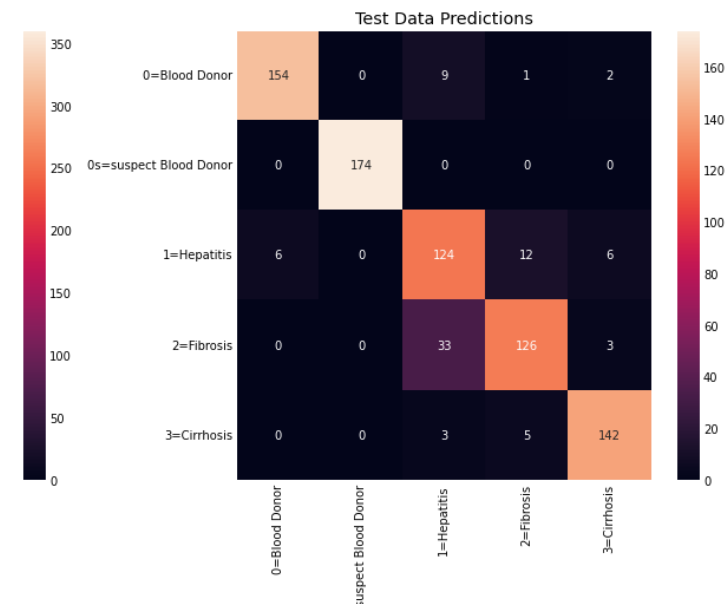
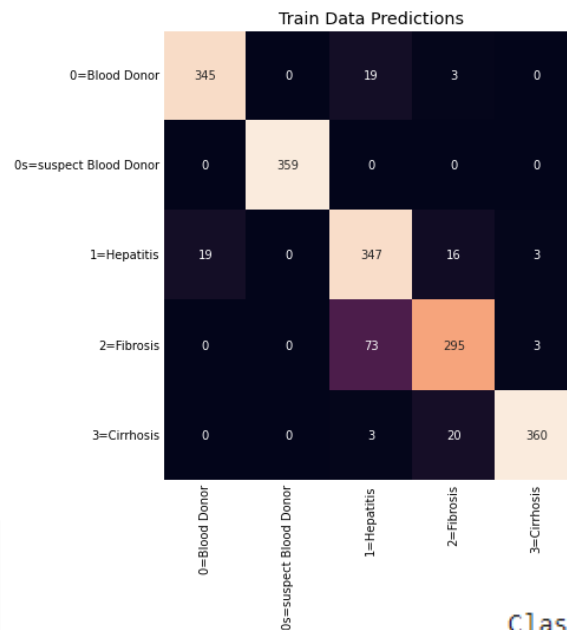
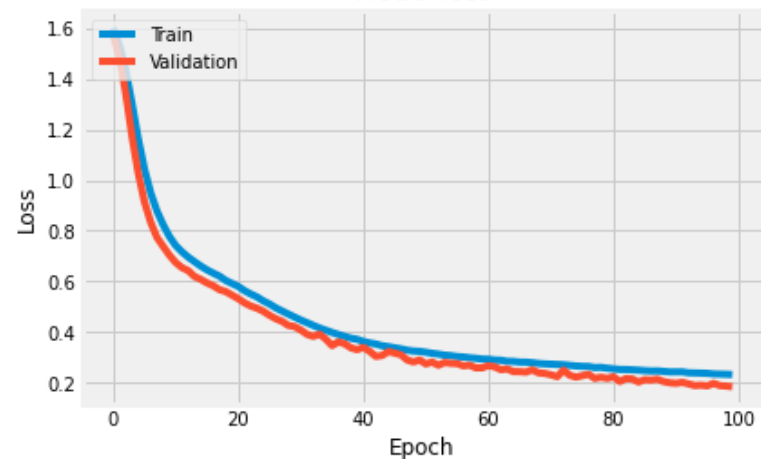
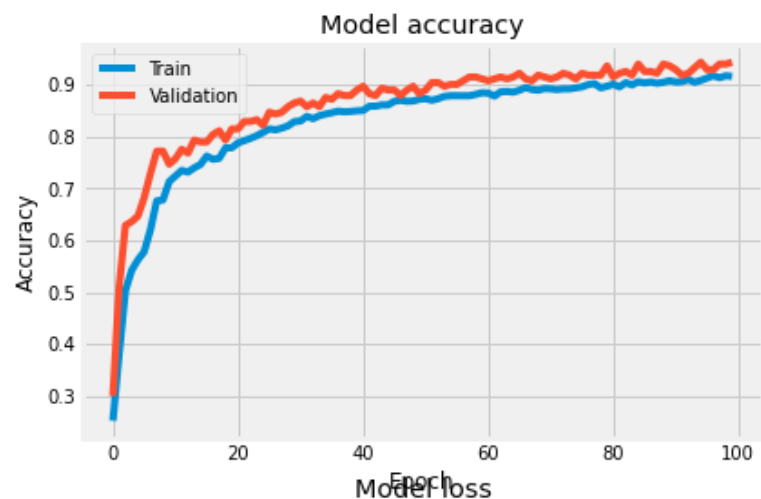
Model 3



Classification Report				
	precision	recall	f1-score	support
0	0.72	0.83	0.77	148
1	0.86	0.91	0.88	164
2	0.50	0.36	0.41	166
3	0.56	0.67	0.61	158
4	0.70	0.63	0.66	164
accuracy			0.68	800
macro avg	0.67	0.68	0.67	800
weighted avg	0.67	0.68	0.67	800



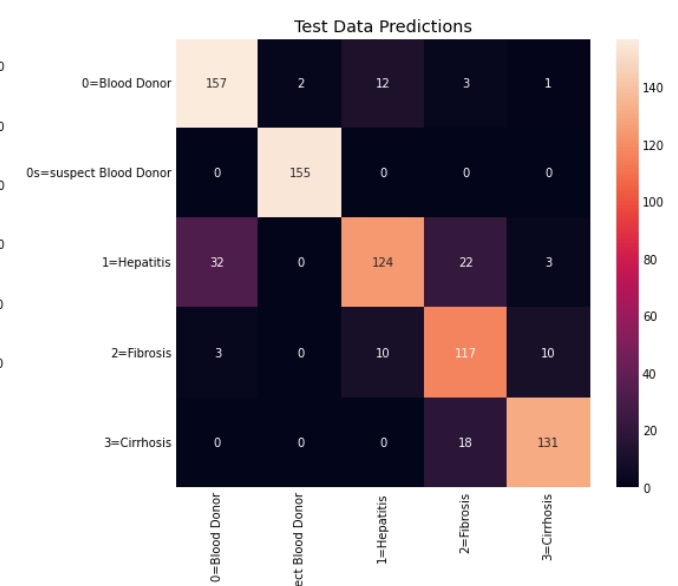
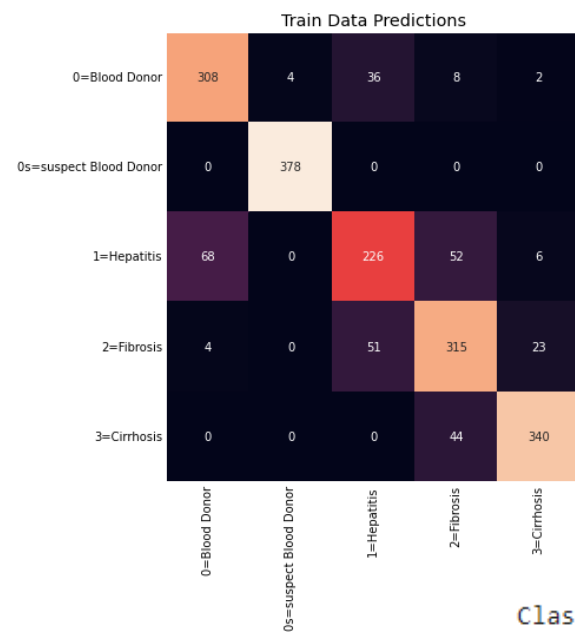
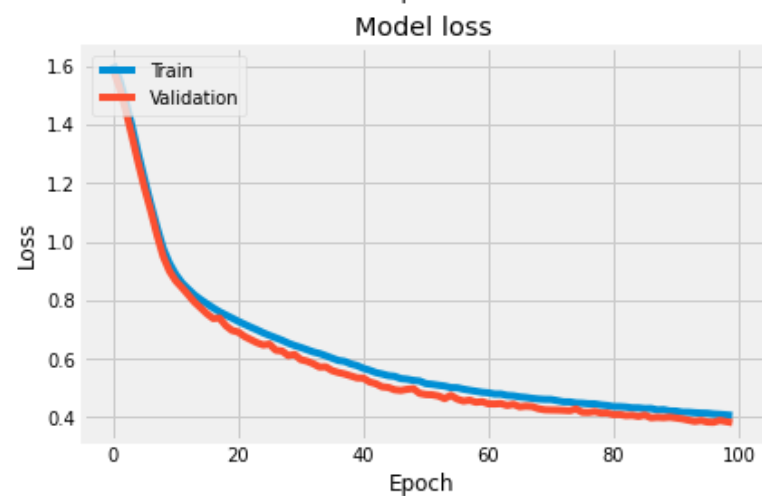
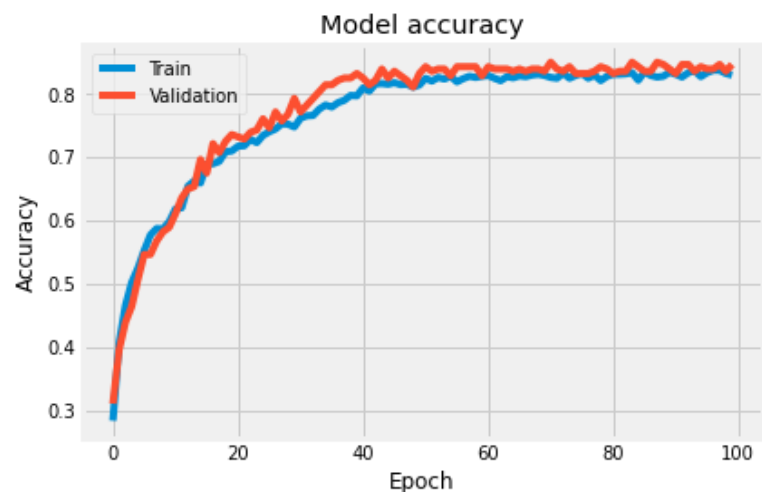
Model 4



Classification Report				
	precision	recall	f1-score	support
0	0.96	0.93	0.94	166
1	1.00	1.00	1.00	174
2	0.73	0.84	0.78	148
3	0.88	0.78	0.82	162
4	0.93	0.95	0.94	150
accuracy			0.90	800
macro avg	0.90	0.90	0.90	800
weighted avg	0.90	0.90	0.90	800



Model 5

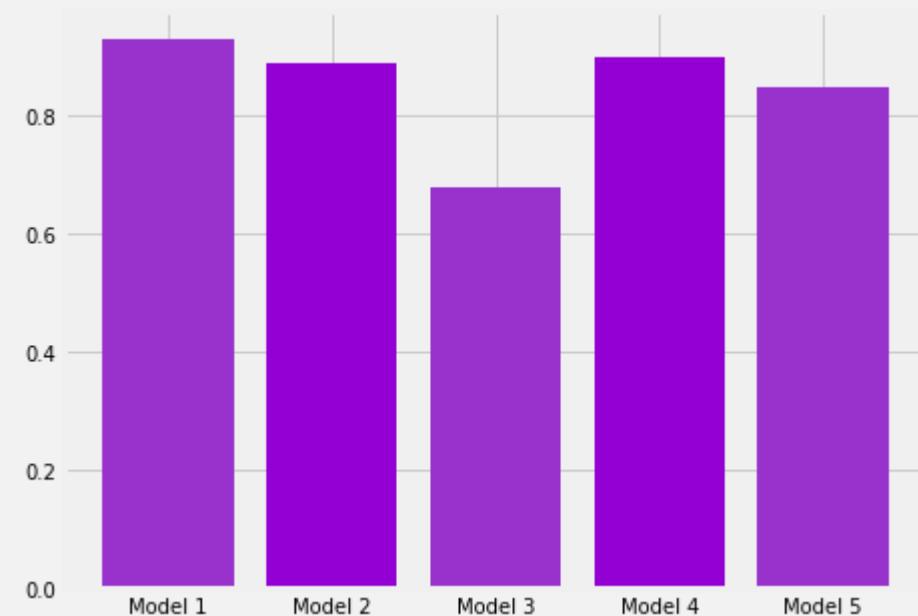
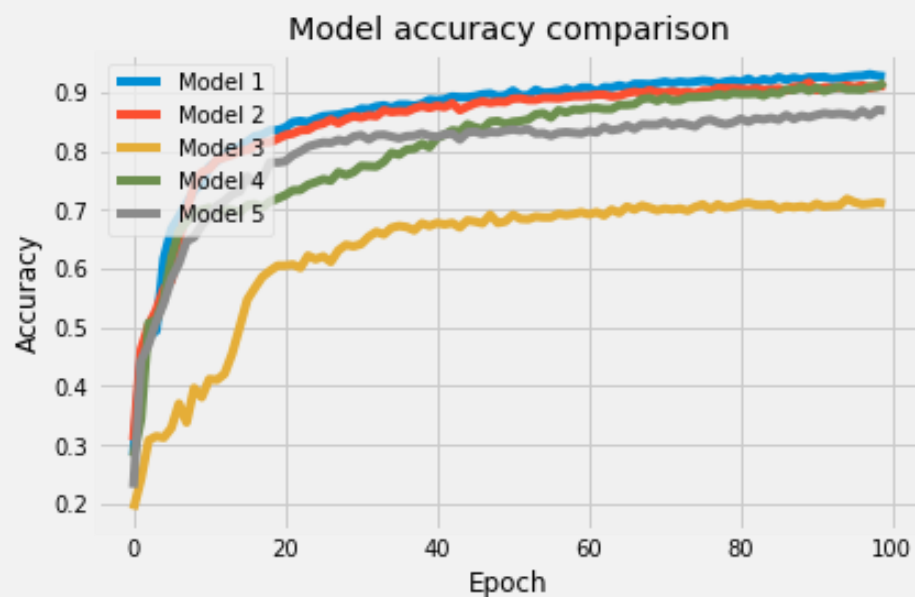


Classification Report				
	precision	recall	f1-score	support
0	0.82	0.90	0.86	175
1	0.99	1.00	0.99	155
2	0.85	0.69	0.76	181
3	0.73	0.84	0.78	140
4	0.90	0.88	0.89	149
accuracy			0.85	800
macro avg	0.86	0.86	0.86	800
weighted avg	0.86	0.85	0.85	800



Model Comparisons

- Bar chart on the right compares the different models' F1-score accuracies, on the left is the curve representation per epoch.





Main Findings

- Full Model: Best model with F1-score accuracy of 93% for unseen data
- Age variable is insignificant (drop to 90% accuracy from the full model's 93%)
- Model with only three variables, AST, GGT and BIL (correlation matrix highest correlations) produces a score of 68%, suggesting that the correlation matrix approach to variable selection could be useful for higher number of variables
- Combination of ML techniques for variable selection is both useful in reducing the complexity of our models, but also maintaining the high F1-score accuracy (still near 90%. Hypothesis turned out to be true.



Conclusions

- Potential improvements on the variable selection: Associating a weight to each of the ML techniques' variable importance → weighted averages, etc.
- Issues with the neural networks themselves: increasing the number of epochs, errors left unaccounted for, metrics for evaluation of models
- Improvement on pre-existing methods observed in literature, boasting high accuracies of $\geq 90\%$ (!)

Always more to be studied and improved upon (time limitations and so on...)

output layer [y]

Thank you
for listening

STA 401 – Conclusion to Data Mining 😊

Expected output:
Fatigue
Actual output:
Fatigue