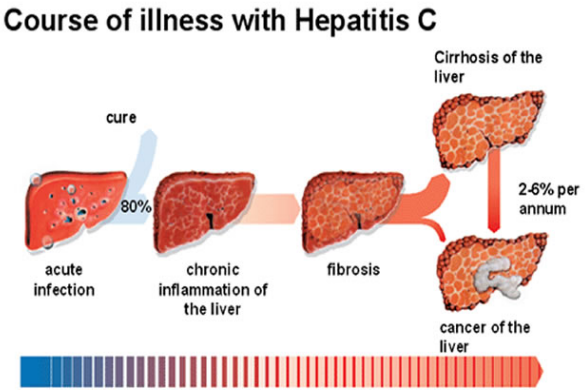
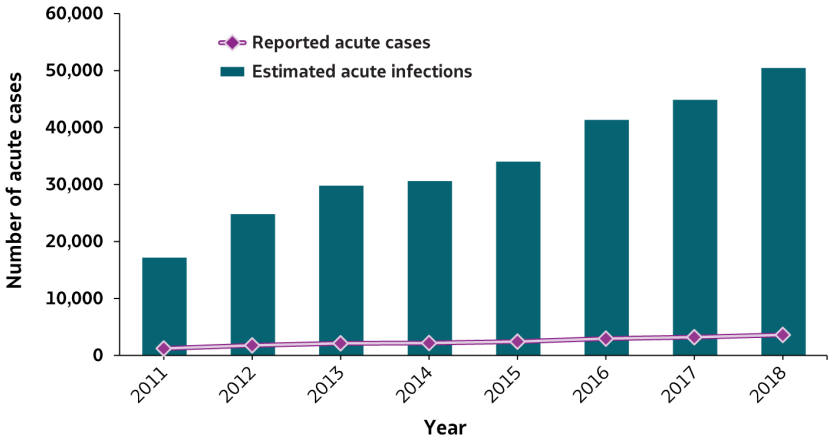
Classifying the Risk of Hepatitis C based on common blood test results

Background information

Hepatitis C is a liver infection spread by contact with blood. It is estimated that yearly around 15000 Americans die annually form Hepatitis C and worldwide, that figure is around 290000. If caught early, treatment with antiviral tablets will be administered and the prognosis is typically very good. If left untreated, typical progression of the disease begins with inflammation of the liver followed by milk scarring. At this stage, symptoms might be completely unnoticeable or mild. This can progress into more serious liver scarring and fibrosis (accumulation of scar tissue) which ultimately leads to cirrhosis of the liver (when scar tissue is extremely prevalent). This hinders liver function and will increase the risk of liver cancer. This illustration demonstrates this phenomenon:



In addition to this, the cases of Hepatitis C have been rising year on year at an alarming rate as outlined by this data from the CDC:



Most deaths occur at the stage of liver cirrhosis and liver cancer, and as a result, the importance of useful early screening results is paramount and being able to detect Hepatitic C early will dramatically improve patient outcomes. Consequently, our study aims to classify

on multiple types of Hepatitis C categorisations to attempt to delineate the severity of disease.

Dataset information

This dataset is taken from Lichtinghagen,Ralf, Klawonn,Frank, and Hoffmann,Georg. (2020). HCV data. UCI Machine Learning Repository. https://doi.org/10.24432/C5D612 *Using machine learning techniques to generate laboratory diagnostic pathways—a case study*. This sources underlying data from:

Lichtinghagen, R., Pietsch, D., Bantel, H., Manns, M. P., Brand, K., & Bahr, M. J. (2013). The enhanced liver fibrosis (ELF) score: Normal values, influence factors and proposed cut-off values. Journal of Hepatology, 59(2), 236–242. https://doi.org/10.1016/j.jhep.2013.03.016

The data contains a sample of 540 healthy patients and 75 patients who were diagnosed with hepatitis C.

The healthy volunteers were recruited from Hannover Medical School and were exmained prior to being included in the study for factors such as liver disease, alcoholism, renal function and Hepatitis C itself (amongst other things). Participation was entirely voluntary and all patients gave written consent. Patients underwent a blood test which measured a variety of demographic data as well as liver function analysis, including liver enzyme activity, etc.

The following information about each of the patients was taken:

- **Age**: Exact age in years
- **Sex**: [male,female]
- **ALB**: Albumin (g/L)
- **ALP**: Alkaline Phosphatase (IU/L)
- **ALT**:Alanine Transaminase (IU/L)
- **AST**: Aspartate Aminotransferase(IU/L)
- **BIL**: Bilirubin (µmol/L)
- **CHE**: Acetylcholinesterase(IU/L)
- **CHOL**: Cholesterol(mmol/L)
- **CREA**: Creatinine(µmol/L)
- **CGT**: Gamma-glutamyl Transferase(U/L)
- **PROT**: Protein(g/L)

And the response was classified into 5 categories:

- Blood donor('0')
- Suspected blood donor ('0s')
- Hepatitis ('1')
- Fibrosis ('2')
- Cirrhosis ('3')

where the various levels denote the progression of Hepatitis C.

Data Acquisition

The way I acquired the data was by using the 'ucimlrepo' module which had this data preloaded:

```
from ucimlrepo import fetch_ucirepo

hcv_data = fetch_ucirepo(id=571)
```

FAIRness of the data provider

Findable: The data was very findable, it had a digital object identifier (https://doi.org/10.24432/C5D612). The data also had important metadata such as data type(continuous, categorical, etc) and was split up into Features and a Target. The Target also had the various categories explained in metadata. Presence of missing values for each of the columns of data was also mentioned. Also included was a brief description of the dataset which made it easy to understand what field of study the dataset was related to making it more findable.
Accessible: The UCI machine learning repository has become almost a canonical source for searching datasets to use for classification and regression models. Data from it is always available for download.
Interoperable: The metadata uses language that is understandable to researchers from across the world and the language it uses has become very standard language in the field of classification and regression.
Reusable: This license for the dataset was a Creative Commons Attribution 4.0 International (CC BY 4.0) license which allows for the sharing and adaptation for any reason provided credit is given. So, this data is very reusable and is allowed to be freely modified.

Dataset preprocessing

Cleaning:
Generally, there was not a whole lot of cleaning to be done. NA values were clearly marked, etc. We did rename one of the columns to better reflect the name of what was being measured (gamma-glutamyl transferase) from CGT to GGT, the latter is common in the literature. This may have been a typo.

Preprocessing:
One point that must be mentioned is that during part of my classification analysis, categorical values were not always directly handled by the classifier algorithm that I was going to use and I resorted to one-hot encoding the relevant columns when necessary.

Moreover, for certain methods that did not allow for missing values, in which cases I imputed the values with multiple imputation after determining missingness was at random (MAR) – see subsequent analysis for how this was done.

We also added a column 'CategorySimple' to indicate the binary outcome of whether or not

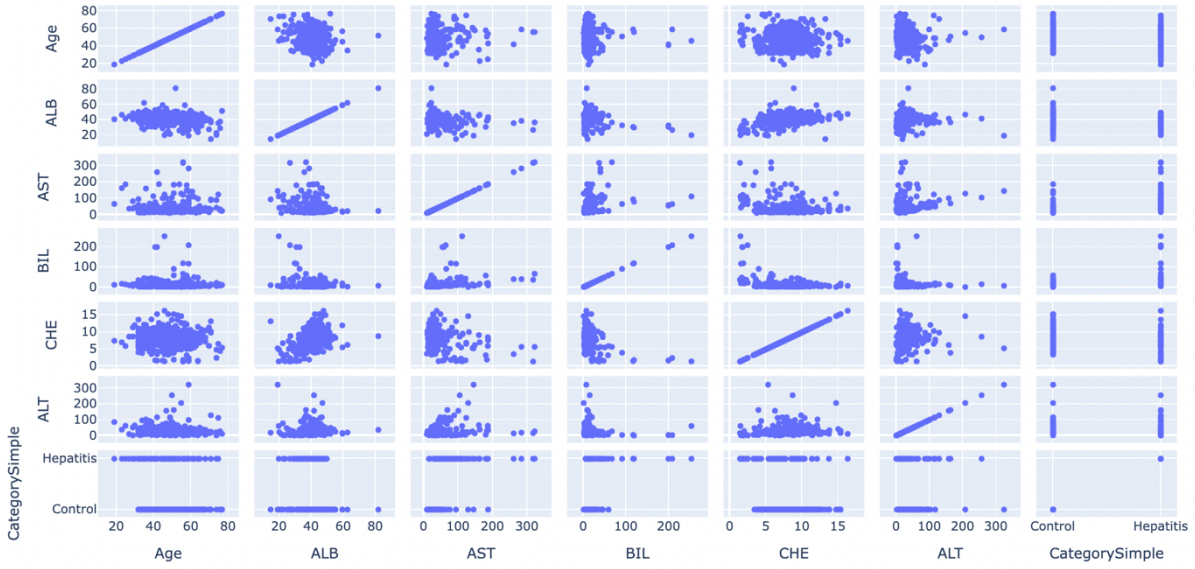
a patient has Hepatitis. The '0s' suspected blood donor group is also absorbed into the main control group as well. I thought this made sense especially since there were very few items in that group. The names were also altered to be more interpretable, the raw data category columns were of the form e.g. '0=Blood Donor'.

Summary Statistics

An overall depiction of the summary statistics can be found here:

	Age	ALB	ALP	AST	BIL	CHE	CHOL	CREA	CGT	PROT	ALT
count	615.000000	614.000000	597.000000	615.000000	615.000000	615.000000	605.000000	615.000000	615.000000	614.000000	614.000000
mean	47.408130	41.620195	68.283920	34.786341	11.396748	8.196634	5.368099	81.287805	39.533171	72.044137	28.450814
std	10.055105	5.780629	26.028315	33.090690	19.673150	2.205657	1.132728	49.756166	54.661071	5.402636	25.469689
min	19.000000	14.900000	11.300000	10.600000	0.800000	1.420000	1.430000	8.000000	4.500000	44.800000	0.900000
25%	39.000000	38.800000	52.500000	21.600000	5.300000	6.935000	4.610000	67.000000	15.700000	69.300000	16.400000
50%	47.000000	41.950000	66.200000	25.900000	7.300000	8.260000	5.300000	77.000000	23.300000	72.200000	23.000000
75%	54.000000	45.200000	80.100000	32.900000	11.200000	9.590000	6.060000	88.000000	40.200000	75.400000	33.075000
max	77.000000	82.200000	416.600000	324.000000	254.000000	16.410000	9.670000	1079.100000	650.900000	90.000000	325.300000

In terms of a scree plot (with several columns removed to improve interpretability) –



We see that the standard deviations of ALP, AST and BIL are relatively high compared to their means. This could possibly be because there is a skewing from the Hepatitis patients and we will investigate this further. We also note that the dataset has some degree of missingness, and we further analyse the missingness later. The column with the highest degree of missingness is 'ALP'.

One way this could be misleading is that they don't differentiate between the target columns. So even if there is a clear difference between the rows corresponding to patients with hepatitis and patients without hepatitis. This might have contributed to the high variance observed for ALP, AST, BIL.

Moreover, since there was **missingness** in the ALP column such observations will not contribute to the mean at all. Since these were actually Missing at random(**MAR**) as we will see later, this would lead to inaccurate summary statistics since certain data was unavailable and could meaningfully change the mean, standard deviation and quantiles.

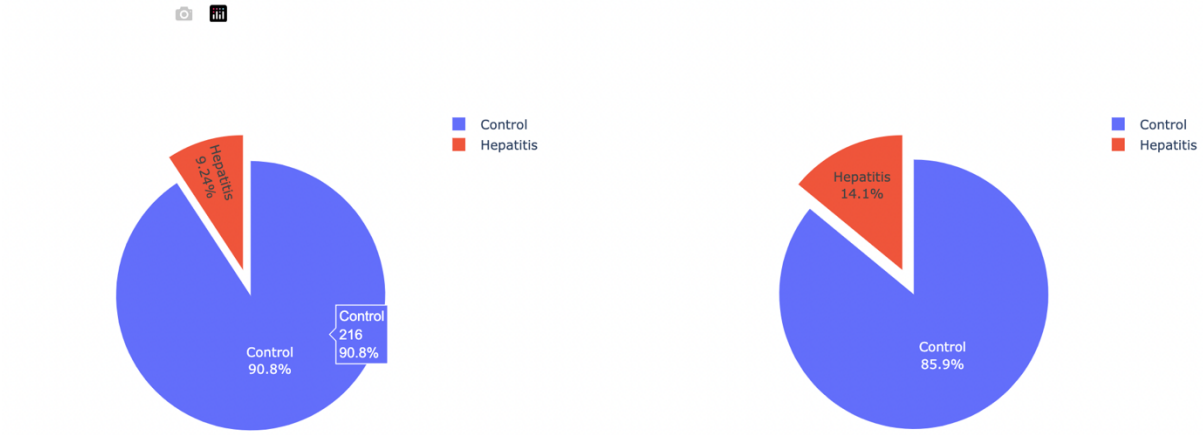
Analyses

1. Is the data imbalanced for particular groups?

Results: We found that there was significant imbalances with regards to both the sex ratio and the ratio of those with the disease and those without. This was determined first by noting the column counts finding that:

There are 75 patients with the disease and 540 without.
There are 337 male patients and 278 female patients.

This effect was demonstrated by a pair of interactive pie charts:



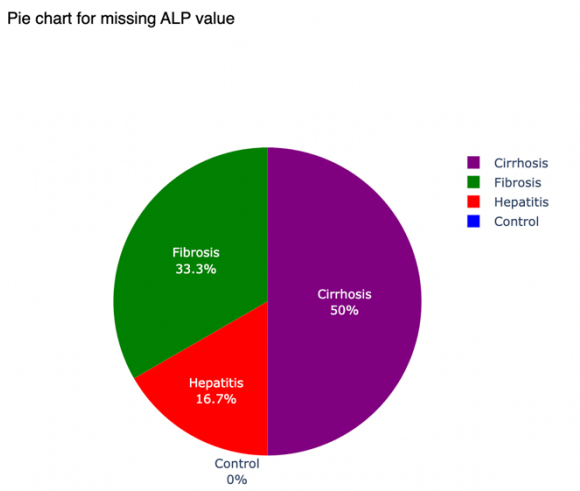
Which shows how men were overrepresented in HCV cases.

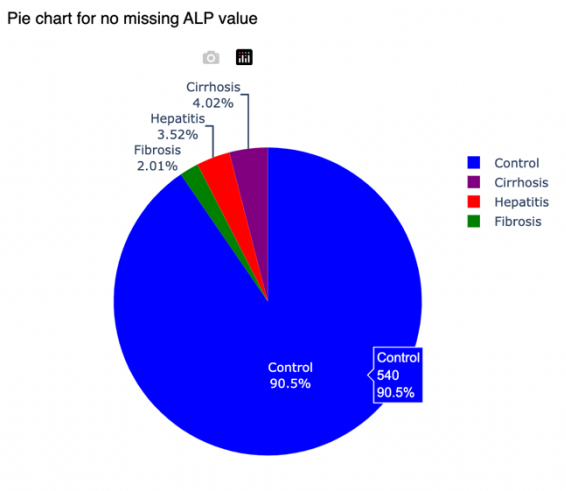
Surprises: The result wasn't surprising as such but informative since I was not aware of the sex disparity prior to looking at these results.

Validation: I validated this with the existing medical literature which verified this imbalance between sexes. The disparity in HCV was expected but we have to note that this sample was not taken from the general population rather individuals with known Hepatitis C history were combined with a random control group.

2. How does the missingness off ALP levels correspond to the prevalence of Hepatitis C?

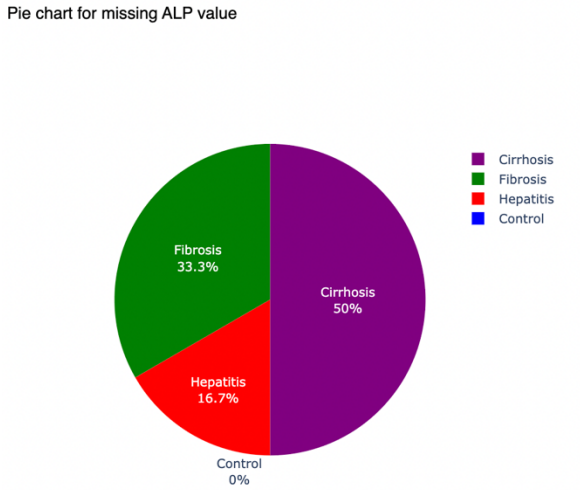
Result: From our interactive table (and summary statistics), we could see that most of our missing data came from the ALP column. We are going to analyse the presence of hepatitis amongst groups with and without a missing value for ALP:

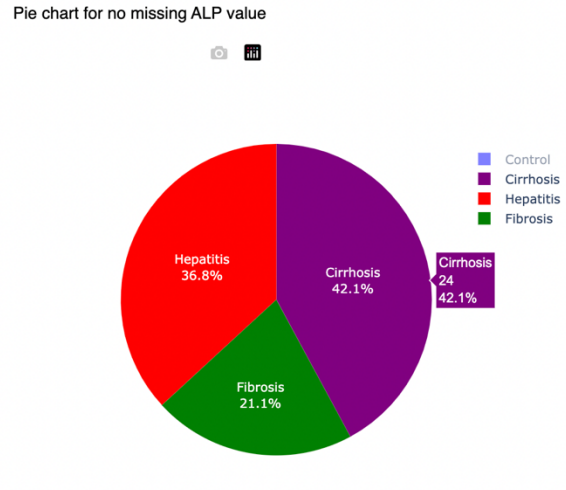




From this, we can see the missingness in this column is MAR(Missing at random) since all the data when the ALP value was missing corresponded to patients with Hepatitis.

Even when we look at those with hepatitis in both groups there is a difference in proportions:



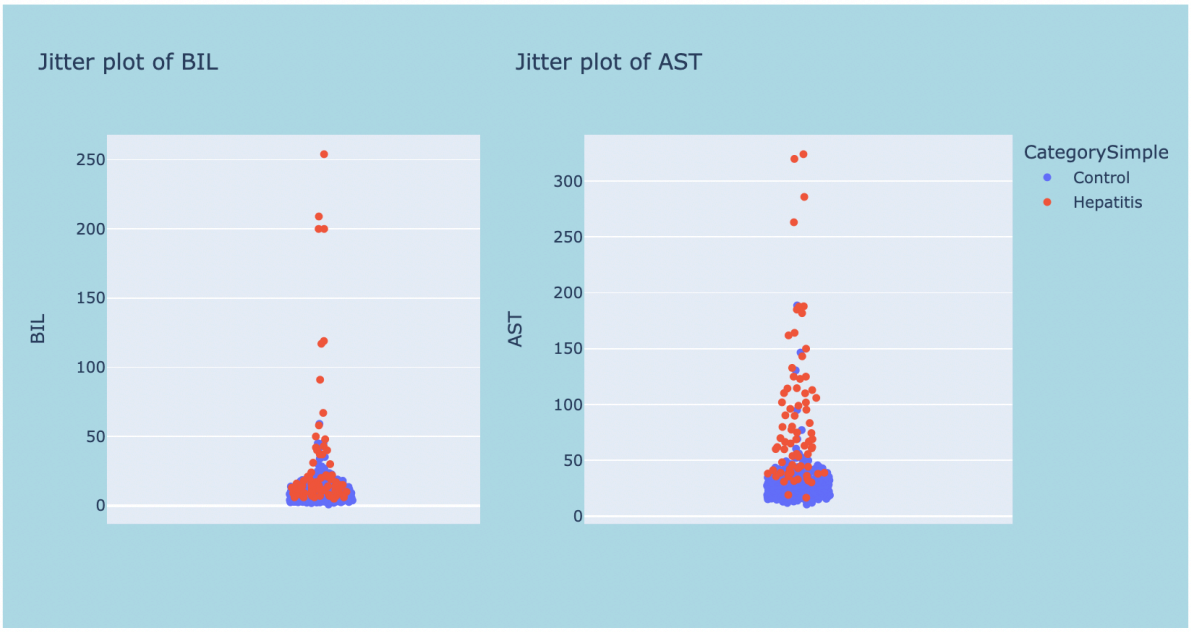


Surprises: When I first saw the missingness, I assumed it was MCAR but this analysis proved useful in demonstrating the type of missingness was actually correlated with the response.

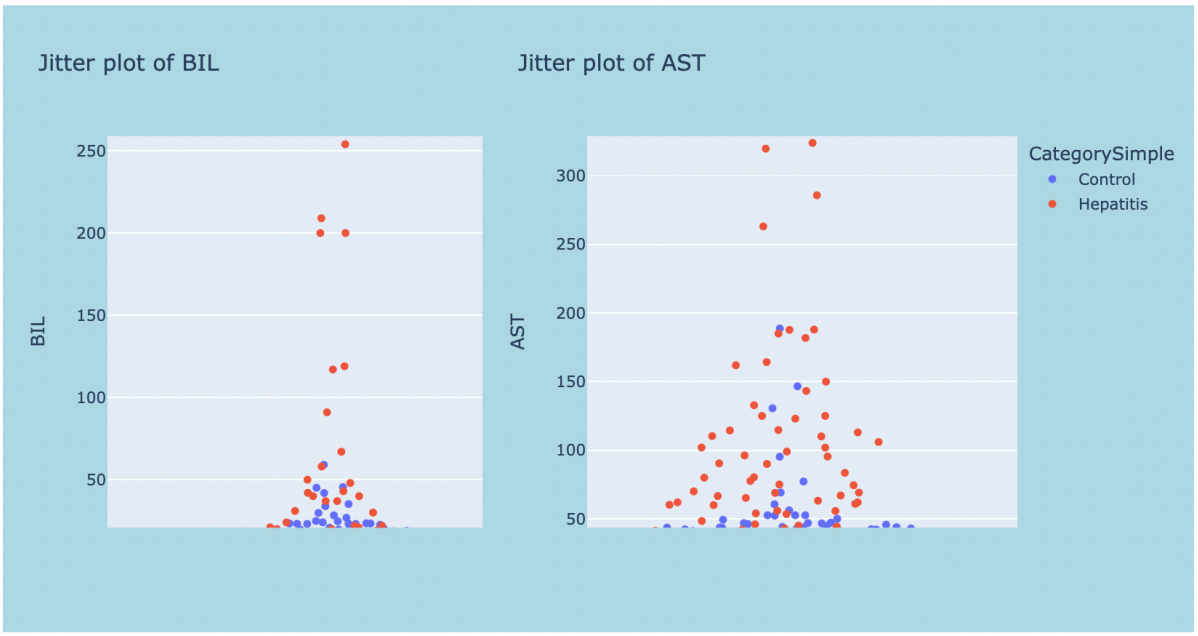
Validation: I went back to the table to visually verify the results we saw.

3. How do the BIL and AST levels relate to the prevalence of Hepatitis C?

We used a pair of jitter plots:



So, we can clearly see a strong link between the AST and BIL levels and the presence of Hepatitis. This is especially the case with AST with most high values being marked as having Hepatitis. If we zoom in only on high values of both, this becomes even more apparent.



Surprises: This wasn't too surprising after looking at the initial table as well as looking up that both were controlled by liver activity.

Validation: I verified from medical literature that these components were heavily related to liver activity and were linked with hepatitis C.

4. Can classification models accurately predict the presence of Hepatitis C?

We used 4 different classification models (XGBoost, Support Vector Classifier, Ridge Classifier, Multinomial Classifier) to be able to predict the target. We employed the model both on the 4 category 'Category' column and the 2 category 'CategorySimple' category.

Handling Missing Data & Categeorical variables
We employed multiple imputation whenever needed with 10 rounds of iterations to impute new values. This should help capture the effects of some of the MAR we observed.

Categeorical variables were one-hot encoded when needed.

Stratified Sampling
We are using stratified sampling in this case since the healthy group represent a very large proportion of the data. When we combine this with cross validation, some of our folds will have very few patients with hepatitis. Stratified sampling here aims to rectify this issue by trying to guarantee all of the folds have proportions of items in each category.

Hyperparameter Tuning Details

The following hyperparameters are being tuned for the above methods:

- **XGBoost**: None
- **Support Vector Classifier**: C, the penalty term for the L-2 norm penalty
- **Ridge Classifier**: λ, the standard penalty term for ridge regression
- **Multinomial Classifier**: 1/C, where C is the coeffecient of the l2-norm

In general, the methods have been tuned using **5-fold stratified sampled cross-validation**.

Our web interface allowed you to manually enter hyperparameters. In our web-interface, we also displayed the confusion matrix for a single run.

We display the results of using each of our 4 classifiers (tuning parameters when necessary). We average the errors over **100** simulations and display the outputs in the table below:

Method	Balanced Accuracy on Category	Balanced Accuracy on CategorySimple
XGBoost	65.6%	94.9%
Support Vector Classifier	44.7%	82.1%
Ridge Classifier	42.6%	71.3%
Multinomial Classifier	59.7%	85.3%

Note: The Balanced Accuracy Score has been used in the literature when we have unbalanced groups as an alternate classification accuracy metric, essentially, we average the classification rates over the different groups.

From this we can see that the performance of XGBoost is superior comapred to the other methods, however, all methods have reasonably high accuracy on the 2 category classification problem.

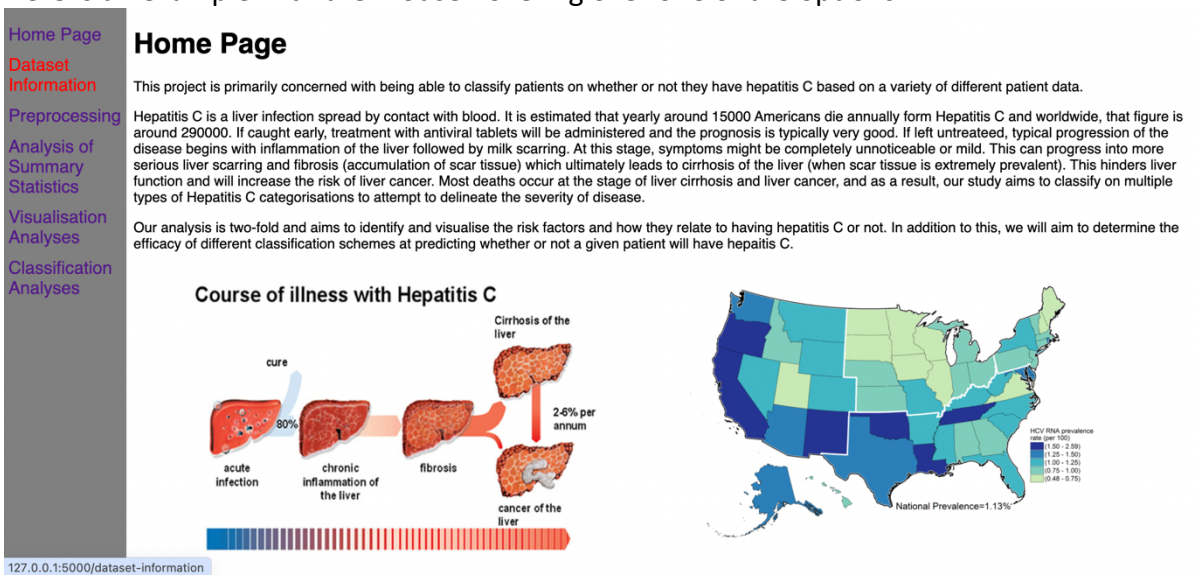
We do note, however that all methods performed relatively poorly on the multi cateogry classification, with in general much lower scores. We saw this on the confusion matrix as well, that although we were able to differentiate between the hepatitis vs. non-hepatitis categories, the exact determination of the stage of disease was much poorer.

Surprises: I was somewhat surprised that the multi-category classification was so poor, I was expecting it to be able to predict all categories with more success.

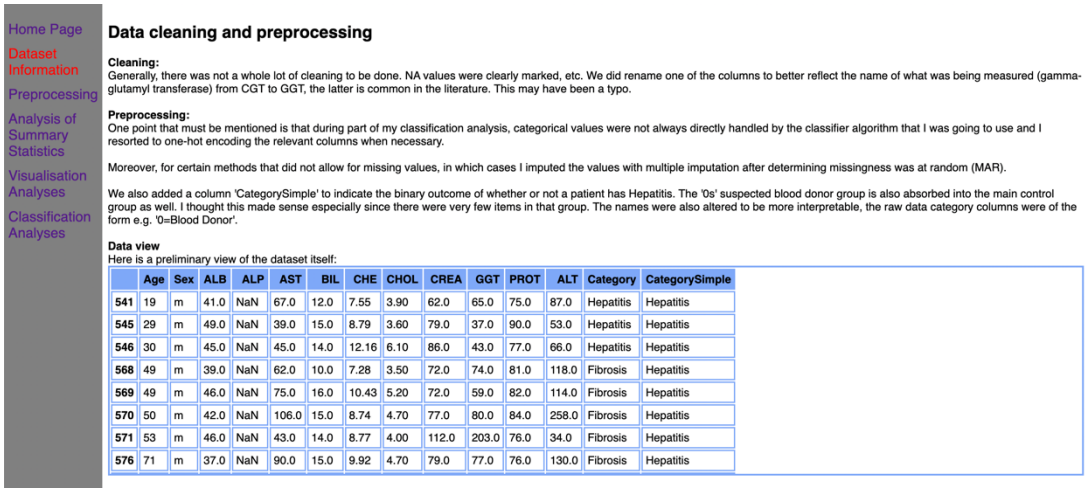
Validation: I had separated the test and training data so the balanced accuracy score itself was a good form of validation and allowed me to compare the different methods.

Web front-end

We have several different sites on the website and a sidebar that navigates between them. Here is an example with the mouse hovering over one of the options:



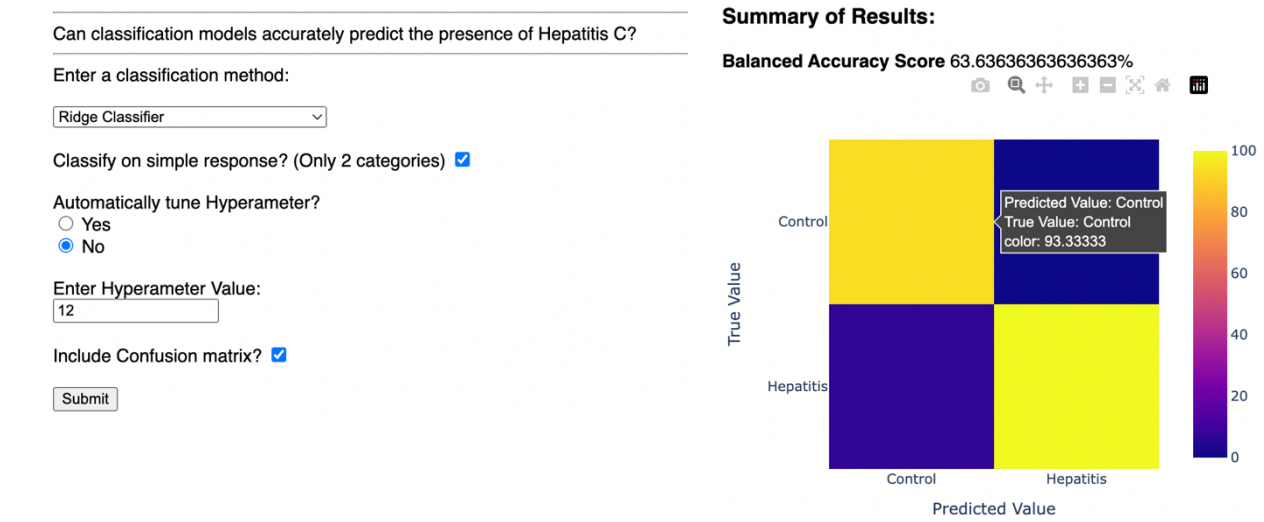
In our data cleaning and preprocessing example, we provide our table in scrollable format with the columns allowing for sorting the data numerically. Here is an example where the column ALP has been sorted with NaN's on the top:

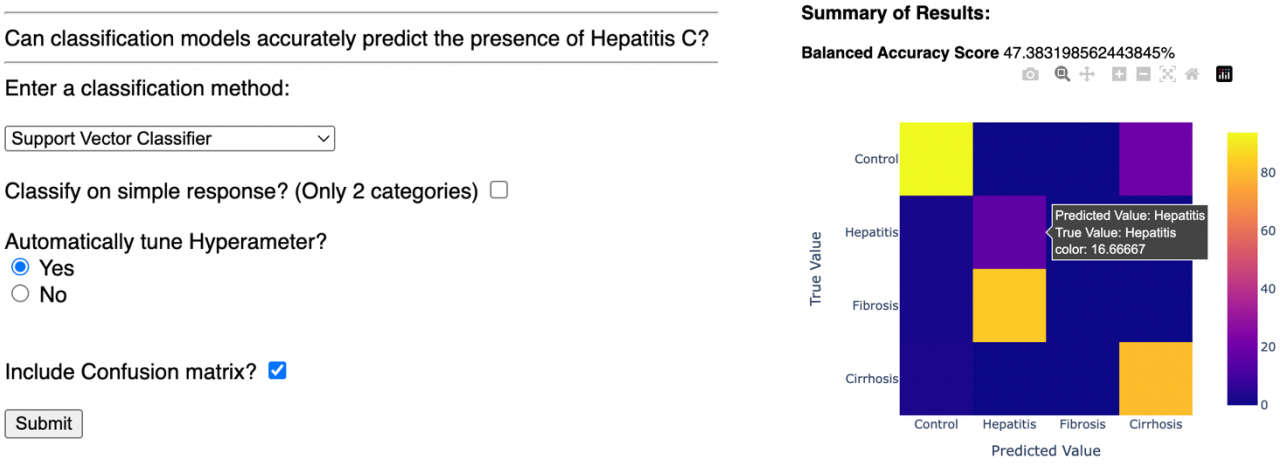


	Age	Sex	ALB	ALP	AST	BIL	CHE	CHOL	CREA	GGT	PROT	ALT	Category	CategorySimple
541	19	m	41.0	NaN	67.0	12.0	7.55	3.90	62.0	65.0	75.0	87.0	Hepatitis	Hepatitis
545	29	m	49.0	NaN	39.0	15.0	8.79	3.60	79.0	37.0	90.0	53.0	Hepatitis	Hepatitis
546	30	m	45.0	NaN	45.0	14.0	12.16	6.10	86.0	43.0	77.0	66.0	Hepatitis	Hepatitis
568	49	m	39.0	NaN	62.0	10.0	7.28	3.50	72.0	74.0	81.0	118.0	Fibrosis	Hepatitis
569	49	m	46.0	NaN	75.0	16.0	10.43	5.20	72.0	59.0	82.0	114.0	Fibrosis	Hepatitis
570	50	m	42.0	NaN	106.0	15.0	8.74	4.70	77.0	80.0	84.0	258.0	Fibrosis	Hepatitis
571	53	m	46.0	NaN	43.0	14.0	8.77	4.00	112.0	203.0	76.0	34.0	Fibrosis	Hepatitis
576	71	m	37.0	NaN	90.0	15.0	9.92	4.70	79.0	77.0	76.0	130.0	Fibrosis	Hepatitis

The summary statistics and visualization analyses provide all the interactive graphs and tables we described earlier.

The classification analysis page allows us to specify our model, whether we want to predict on ‘CategorySimple’ or ‘Category’, whether to automatically tune a hyperparameter and whether to include a confusion matrix. A few examples of its use is included here:





The confusion matrix we use here outputs the column wise percentage of being in each group since I thought this was more interpretable due to group imbalance.

Server API

The website is built on a flask server python web framework that uses jinja templates to render each of the different websites.

We used a Remote Procedure Call (RPC) API here, the URLs we use are easy to understand what it is displaying/doing, e.g. “/dataset-information”, “/preprocessing”, ” /summary-statistics”, “/render-analysis”, etc. Also, the URLs are clean and separate the entity from implementation with no file extensions in the URL.

However, we do pass parameters to functions in the URL itself. This is the case in the classification analysis page which passes information onto the render-analysis function and URLs for this function can look like: "/render-analysis?method=svc&simple=True&tune=0&hypervalue=12&confusion=True"

So overall, I would say that the API that is actually inspired by REST but also uses RPC.

Surprising results/Unexpected difficulties

1) I found attempting to integrate tables into the html rather challenging. Initially I attempted to try and create each table manually/try and incorporate them into the jinja template row by row. In the end I found just using the to_html feature of pandas dataframe, followed by styling to be easier.
2) I also found creating plotly objects rather difficult but thought it was important to have interactive graphs. I found that the python packages such as plotly express to be very important since they were able to automatically convert plotly objects to html divs.
3) I found creating conditional dropdown menus where one item appears conditional on if a given checkbox is created challenging. I had to understand more how JavaScript works to be able to make this work.