Machine Learning

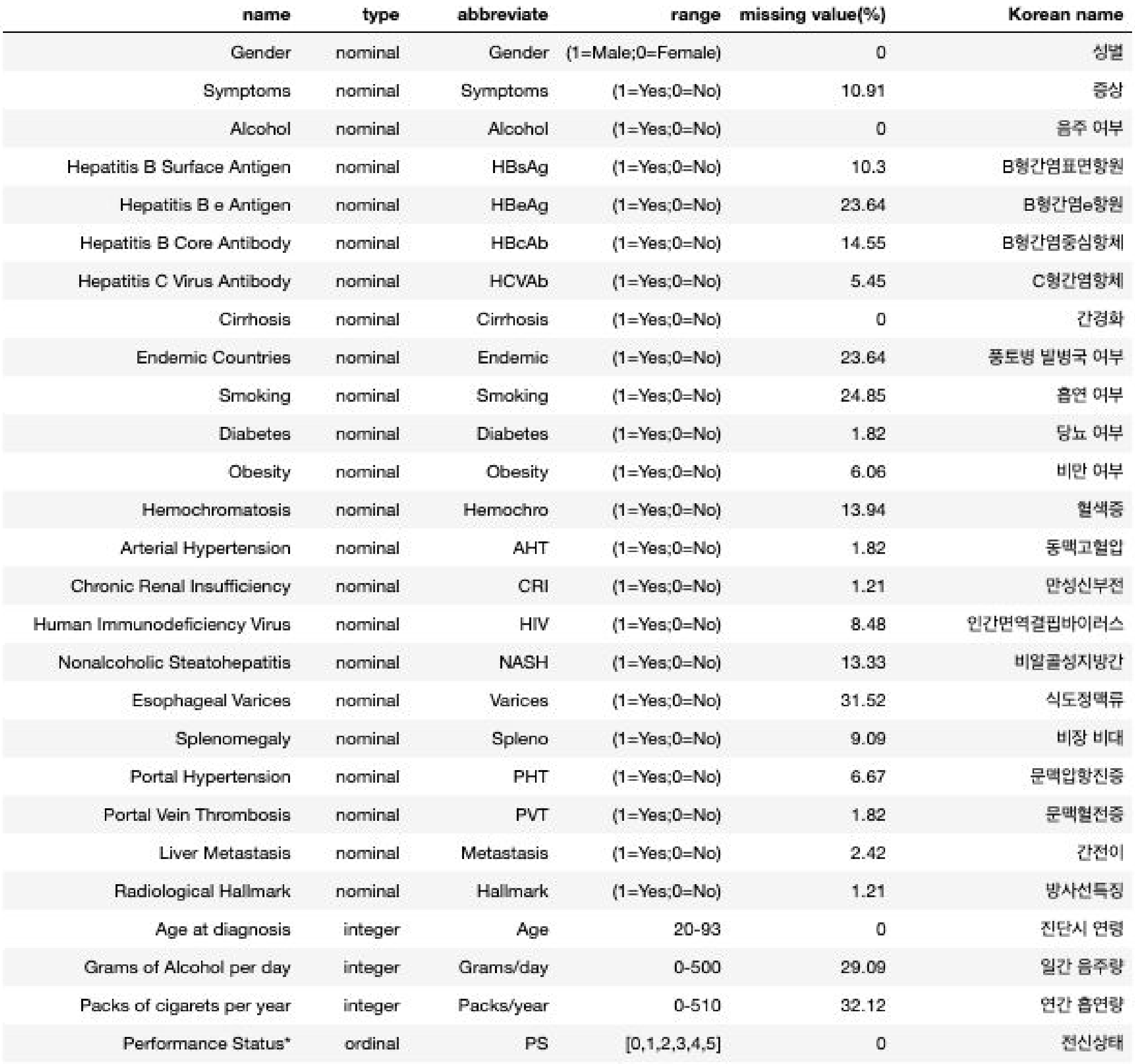
HW2

2016147530
Yerin Kwon

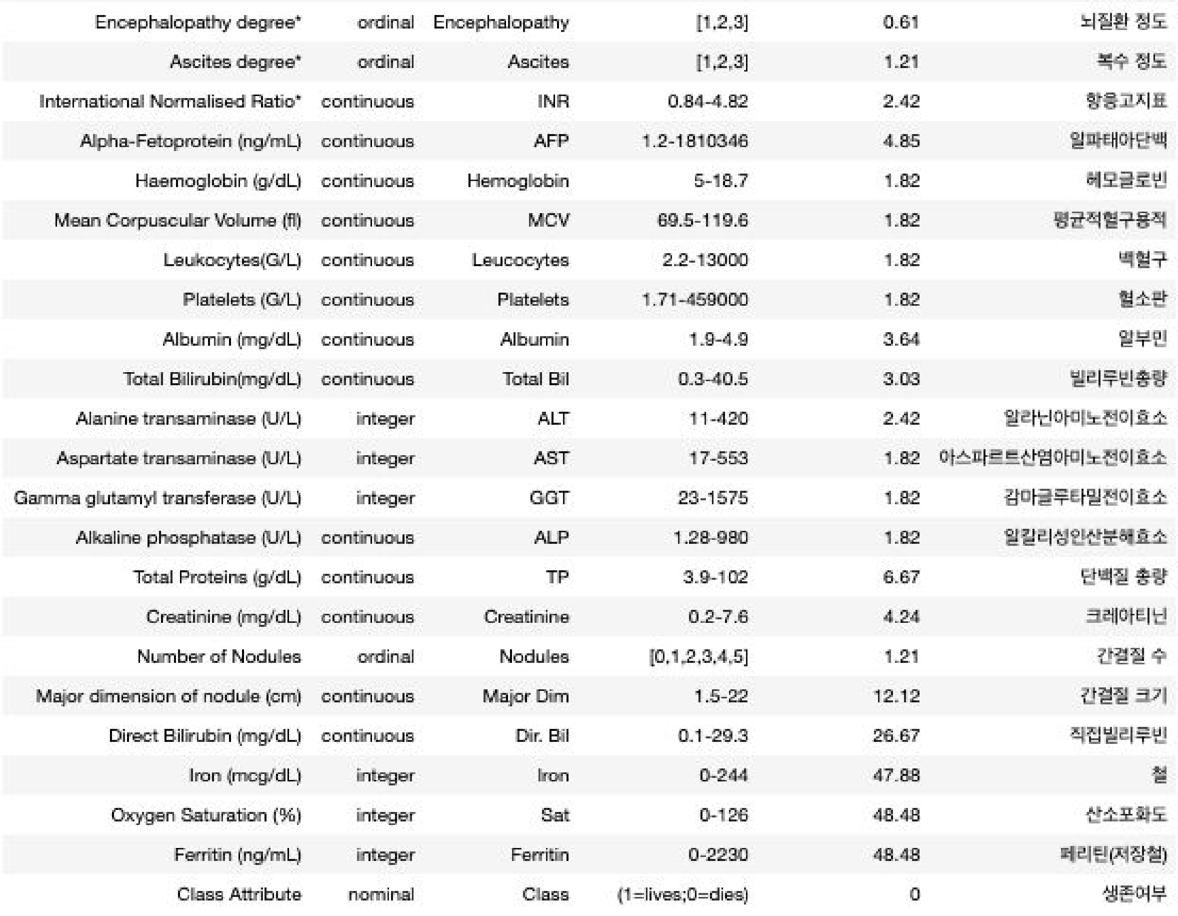
Dataset

Overview

This HCC dataset was obtained at a University Hospital in Portugal and contains several demographic, risk factors, laboratory and overall survival features of 165 real patients diagnosed with HCC. The dataset contains 49 features selected according to the EASL-EORTC (European Association for the Study of the Liver - European Organisation for Research and Treatment of Cancer) Clinical Practice Guidelines, which are the current state-of-the-art on the management of HCC.



name	type	abbreviate	range	missing value(%)	Korean name
Gender	nominal	Gender	(1=Male;0=Female)	0	성별
Symptoms	nominal	Symptoms	(1=Yes;0=No)	10.91	증상
Alcohol	nominal	Alcohol	(1=Yes;0=No)	0	음주 여부
Hepatitis B Surface Antigen	nominal	HBsAg	(1=Yes;0=No)	10.3	B형간염표면항원
Hepatitis B e Antigen	nominal	HBeAg	(1=Yes;0=No)	23.64	B형간염e항원
Hepatitis B Core Antibody	nominal	HBcAb	(1=Yes;0=No)	14.55	B형간염중심항체
Hepatitis C Virus Antibody	nominal	HCVAb	(1=Yes;0=No)	5.45	C형간염항체
Cirrhosis	nominal	Cirrhosis	(1=Yes;0=No)	0	간경화
Endemic Countries	nominal	Endemic	(1=Yes;0=No)	23.64	풍토병 발병국 여부
Smoking	nominal	Smoking	(1=Yes;0=No)	24.85	흡연 여부
Diabetes	nominal	Diabetes	(1=Yes;0=No)	1.82	당뇨 여부
Obesity	nominal	Obesity	(1=Yes;0=No)	6.06	비만 여부
Hemochromatosis	nominal	Hemochro	(1=Yes;0=No)	13.94	혈색증
Arterial Hypertension	nominal	AHT	(1=Yes;0=No)	1.82	동맥고혈압
Chronic Renal Insufficiency	nominal	CRI	(1=Yes;0=No)	1.21	만성신부전
Human Immunodeficiency Virus	nominal	HIV	(1=Yes;0=No)	8.48	인간면역결핍바이러스
Nonalcoholic Steatohepatitis	nominal	NASH	(1=Yes;0=No)	13.33	비알골성지방간
Esophageal Varices	nominal	Varices	(1=Yes;0=No)	31.52	식도정맥류
Splenomegaly	nominal	Spleno	(1=Yes;0=No)	9.09	비장 비대
Portal Hypertension	nominal	PHT	(1=Yes;0=No)	6.67	문맥압항진증
Portal Vein Thrombosis	nominal	PVT	(1=Yes;0=No)	1.82	문맥혈전증
Liver Metastasis	nominal	Metastasis	(1=Yes;0=No)	2.42	간전이
Radiological Hallmark	nominal	Hallmark	(1=Yes;0=No)	1.21	방사선특징
Age at diagnosis	integer	Age	20-93	0	진단시 연령
Grams of Alcohol per day	integer	Grams/day	0-500	29.09	일간 음주량
Packs of cigarets per year	integer	Packs/year	0-510	32.12	연간 흡연량
Performance Status*	ordinal	PS	[0,1,2,3,4,5]	0	전신상태

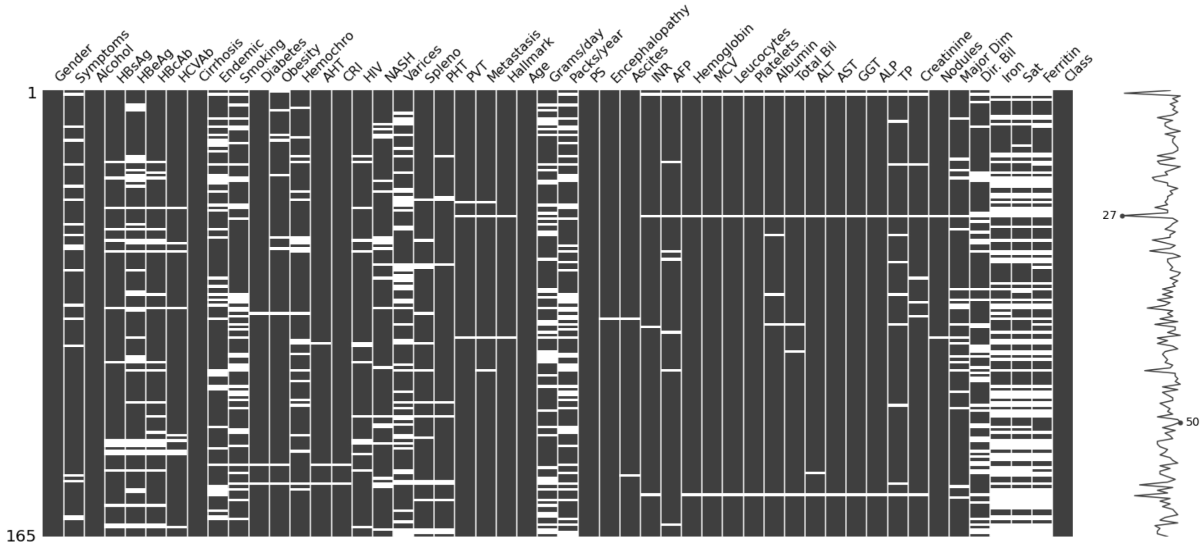


Encephalopathy degree*	ordinal	Encephalopathy	[1,2,3]	0.61	뇌질환 정도
Ascites degree*	ordinal	Ascites	[1,2,3]	1.21	복수 정도
International Normalised Ratio*	continuous	INR	0.84-4.82	2.42	항응고지표
Alpha-Fetoprotein (ng/mL)	continuous	AFP	1.2-1810346	4.85	알파태아단백
Haemoglobin (g/dL)	continuous	Hemoglobin	5-18.7	1.82	헤모글로빈
Mean Corpuscular Volume (fl)	continuous	MCV	69.5-119.6	1.82	평균적혈구용적
Leukocytes(G/L)	continuous	Leucocytes	2.2-13000	1.82	백혈구
Platelets (G/L)	continuous	Platelets	1.71-459000	1.82	혈소판
Albumin (mg/dL)	continuous	Albumin	1.9-4.9	3.64	알부민
Total Bilirubin(mg/dL)	continuous	Total Bil	0.3-40.5	3.03	빌리루빈총량
Alanine transaminase (U/L)	integer	ALT	11-420	2.42	알라닌아미노전이효소
Aspartate transaminase (U/L)	integer	AST	17-553	1.82	아스파르트산염아미노전이효소
Gamma glutamyl transferase (U/L)	integer	GGT	23-1575	1.82	감마글루타밀전이효소
Alkaline phosphatase (U/L)	continuous	ALP	1.28-980	1.82	알칼리성인산분해효소
Total Proteins (g/dL)	continuous	TP	3.9-102	6.67	단백질 총량
Creatinine (mg/dL)	continuous	Creatinine	0.2-7.6	4.24	크레아티닌
Number of Nodules	ordinal	Nodules	[0,1,2,3,4,5]	1.21	간결절 수
Major dimension of nodule (cm)	continuous	Major Dim	1.5-22	12.12	간결절 크기
Direct Bilirubin (mg/dL)	continuous	Dir. Bil	0.1-29.3	26.67	직접빌리루빈
Iron (mcg/dL)	integer	Iron	0-244	47.88	철
Oxygen Saturation (%)	integer	Sat	0-126	48.48	산소포화도
Ferritin (ng/mL)	integer	Ferritin	0-2230	48.48	페리틴(저장철)
Class Attribute	nominal	Class	(1=lives;0=dies)	0	생존여부

[Data Description]

This is an heterogeneous dataset, with 23 quantitative variables, and 26 qualitative variables. Overall, missing data represents 10.22% of the whole dataset and only eight patients have complete information in all fields (4.85%). The target(dependent) variable is the survival at 1 year, and was encoded as a binary variable: 0 (die) and 1 (lives). A certain degree of class-imbalance is also present (63 cases labeled as dies and 102 as lives)

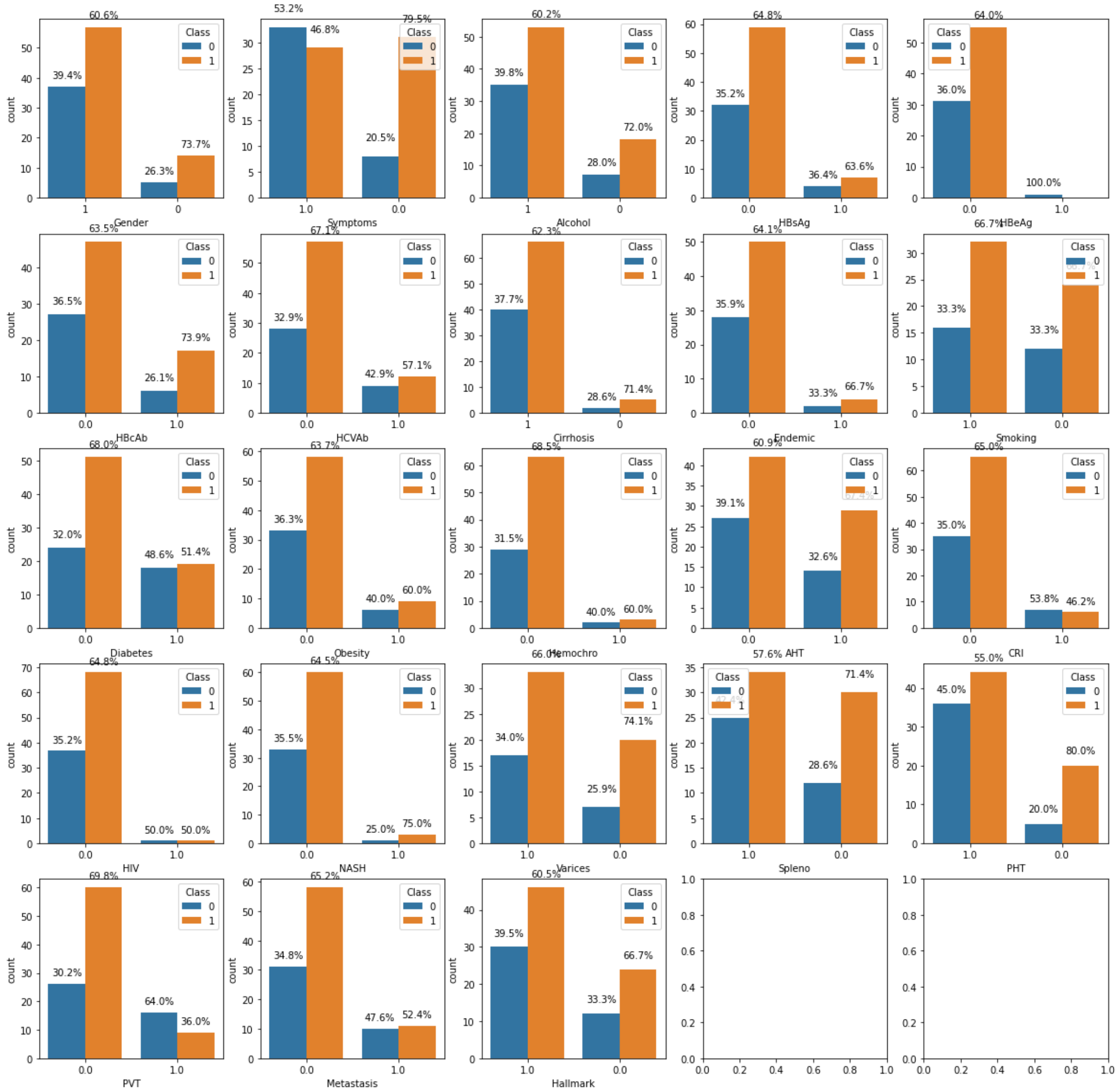
Missing Data



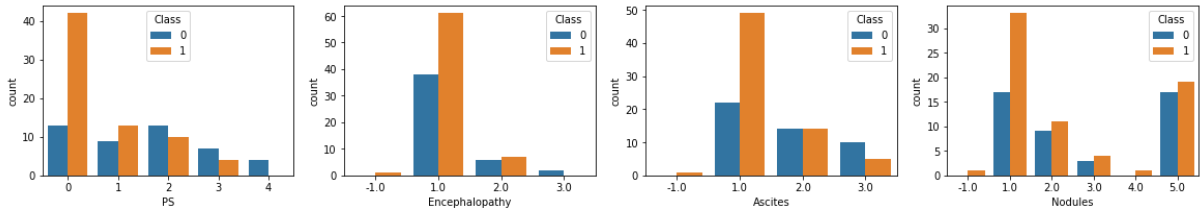
[Missing Data Analysis]

Iron, Sat, and Ferritin have similar missing patterns and also have high missing rate which is over 40%. Thus, we may consider dropping those columns to improve our performance. Also, we can choose removing rows that has a lot of common missing columns. By observation, we can find the rows without Hemoglobin value also have many other missing columns, which means that no blood test was conducted for those patients. For other missing values, since we have little amount of data, it seems better to replace into some other values.

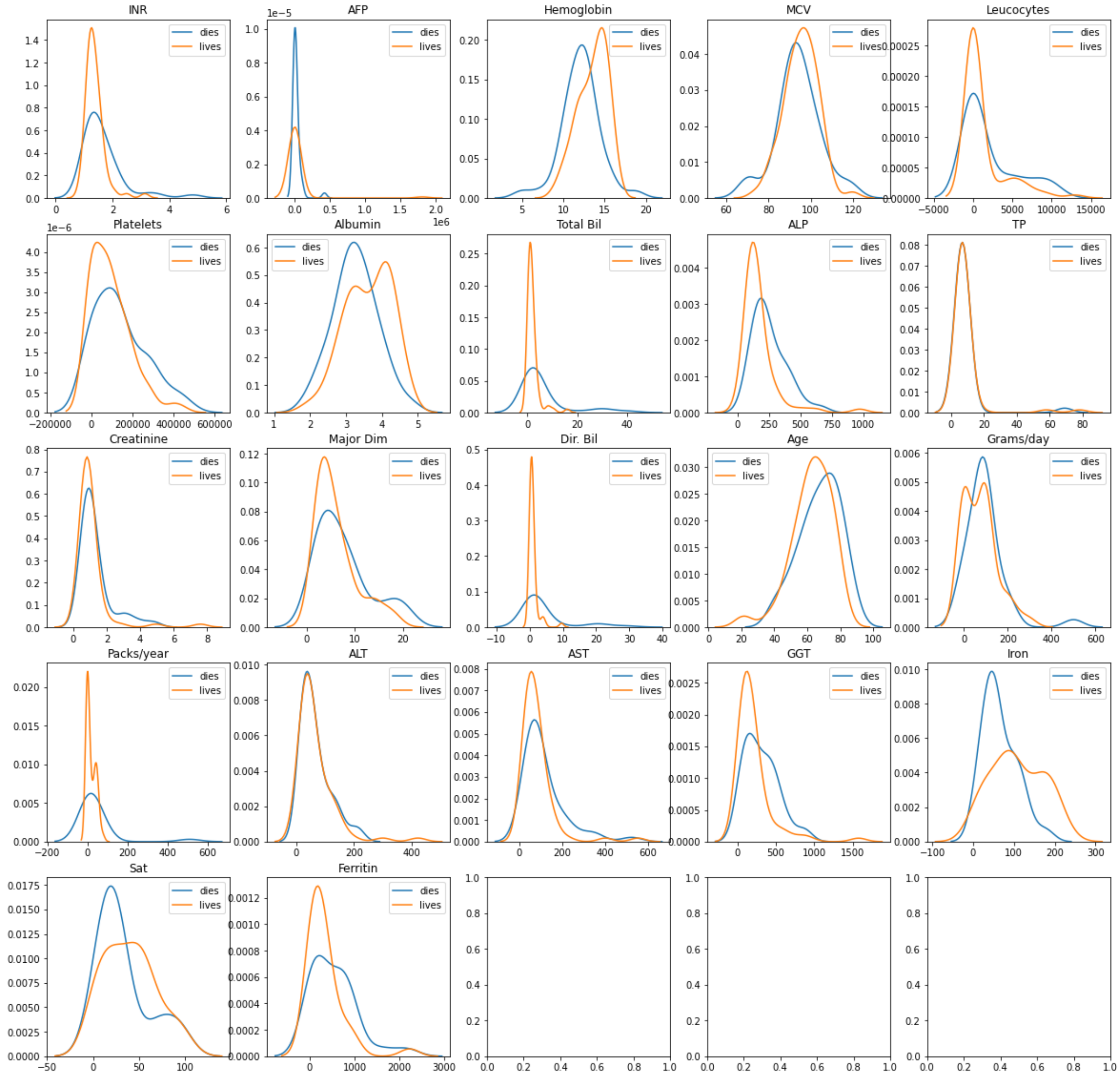
Data Analysis



[Barplot on nominal columns]



[Barplot on ordinal columns]



[KDE plot on continuous columns]

A set of graphs above shows how this data is distributed.

For those nominal values, we can find out that some of features are highly imbalanced so that it is hard to directly extract some meaningful relation between these features and the target. For example, if you look at a barplot of HBeAg (Hepatitis B e Antigen), 100% of the people who have this antigen is dead after a year. However, the number of people who have this antigen is super low so that we cannot say the existence of HBeAg means death.

For ordinal columns, every features seems to have its own semantic. For example, people who have active performance status and no encephalopathy are likely to be still alive after 1 year.

For continuous columns, one interesting point is that Iron, Oxygen Saturation and Ferritin have significantly different distribution between dies and lives. That is, even though they have about 40% of missings, these features are quite important so we should not drop them.

To get more insights whether to select the feature or not, I analyzed correlations between 2 features with heatmap and also tried feature selection in 2 different ways: univariate selection using ANOVA and feature importance from random forest.

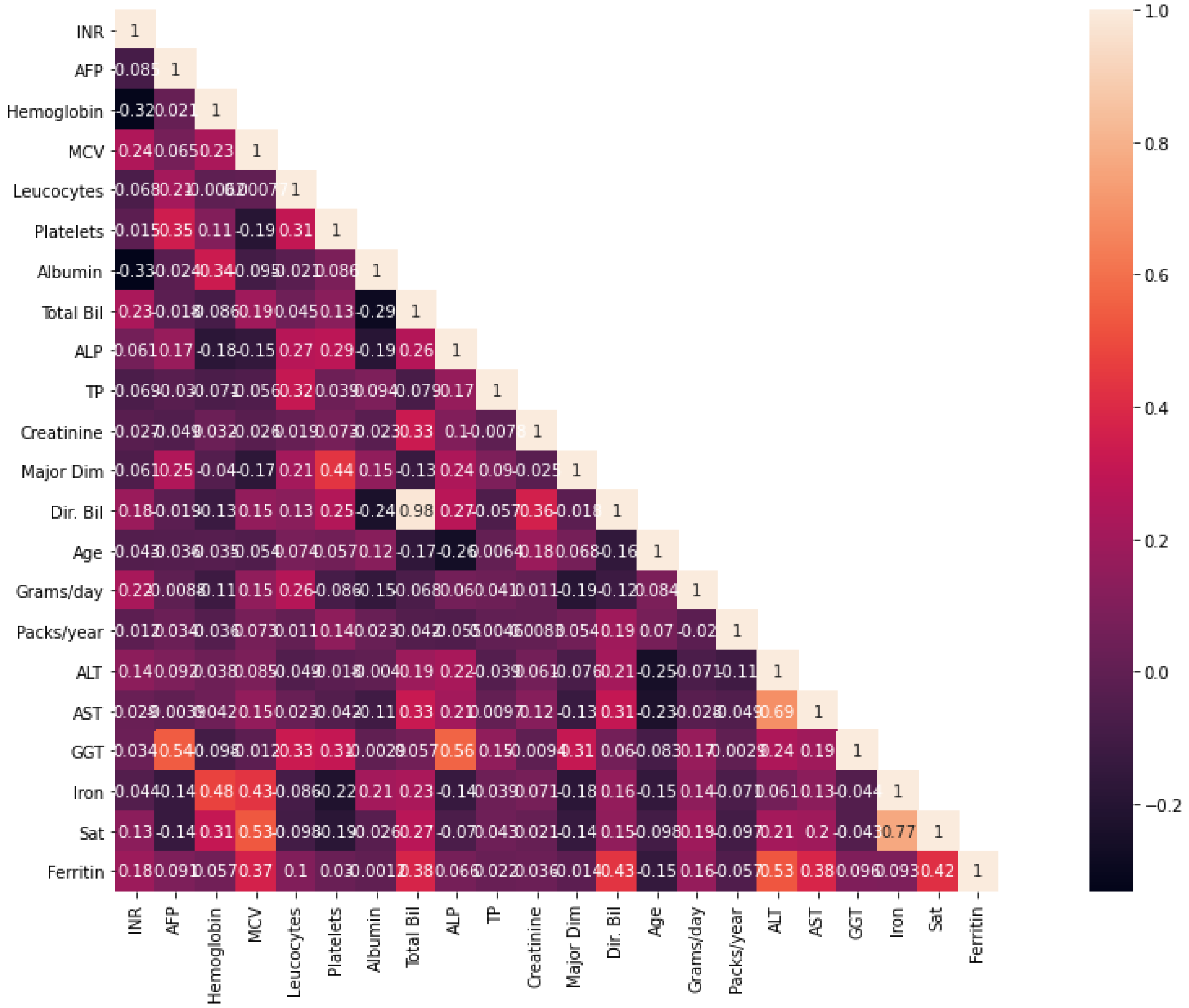
Feature Engineering

Filling Missing Data

I tried several methods like filling with single value(-1), median, mean, most frequent value and KNN impute. Filling with single value gave the best performance so I chose it.

single value	median	mean	mode	KNN
0.6938775510204082	0.5918367346938775	0.673469387755102	0.5510204081632653	0.6326530612244898

Drop Highly correlated features



[Correlation between continuous columns]

This is a heatmap that shows correlation between all pair of variables. The reason I analyzed correlations is because highly correlated features can mask interactions between variables and get numerically unstable solution. Here, Direct Bilirubin(Dir. Bil) and Total Bilirubin(Total Bil) has high correlation. Since Direct Bilirubin has higher missing rows, I decided to remove Dir. Bil rather than Total Bil. After removing this column, I got 4% of increase in performance when using logistic regression with default hyperparameters.

Before After

```
0.6938775510204082 0.7346938775510204
```

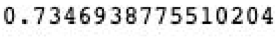
Feature Selection

This data has 49 features and using all of these features may lead the model to be overfitted. Thus, we can observe the features, select only a few and get a better understanding.

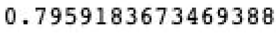
Univariate Selection using ANOVA

One way to select meaningful features is to observe the relationship between each feature and target. This is called univariate feature selection and this method examines each feature individually to determine how much the feature is strongly related to the target. There are a few methods to conduct this feature selection and what I chose is ANOVA. I iterated 1 to 100 (the percentage of remaining features) and discovered that choosing top 47% of existing features gives best performance. Here, I got 6% of performance improvement after dropping highly correlated feature.

Before After



```
0.7346938775510204
```

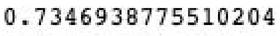


```
0.7959183673469388
```

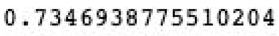
Calculating Feature importance

Another way to select meaningful features is to calculate feature importance with tree-based model. These feature importance is determined by summation of Gini impurity on each node. I extracted the features that has feature importance under 0.005 and 10 columns were selected. After dropping them, I did not get any performance improvement comparing to the result after dropping highly correlated feature.

Before After



```
0.7346938775510204
```



```
0.7346938775510204
```

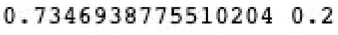
Univariate selection using ANOVA showed better performance than using feature importance in this state. However, this result can be flipped after tuning hyperparameters in model so I tried parameter tuning after conducting both feature selection one by one.

Parameter Tuning

Until now, I used logistic regression model with default hyperparameters when comparing performance. To get higher performance, I tried hyperparameter tuning on this LR model. I iterated 1 through 100 and calculated the score of current model. The parameter I tuned is C, which means inverse of regularization strength and I made it to have a value of i/10 so that it ranges from 0.1 to 10. For other hyperparameters, I used 1e-5 of tolerance, 'lbfgs' solver and 500 maximum iteration.

For those after univariate selection with ANOVA, the performance decreases about 6% from the former result when C=0.2

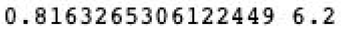
Score C



```
0.7346938775510204 0.2
```

For those after feature selection based on feature importance, the performance improved about 8% from the former result when C=6.2

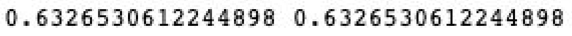
Score C



```
0.8163265306122449 6.2
```

Also, I wanted to check and compare the performance of tree based model. So, I implemented random forest model. I first tried simplest one with default hyperparameters. The result is as follows: the performance before feature engineering and after feature engineering has no difference and is much lower than the one with logistic regression. So, I decided to use logistic regression as my final model.

Before After



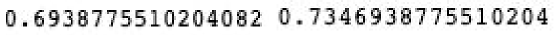
```
0.6326530612244898 0.6326530612244898
```

Result

Following scores and graphs are comparison on basic logistic regression model with no feature engineering and our tuned logistic regression model with final selected & modified features. Since this dataset is about binary classification, I implemented ROC(Receiver Operating Characteristic) curve to examine improvement on performance. We can easily check this by AUC(Area Under Curve). When looking at the result of ANOVA, the test score is 4% higher than the original one and AUC is 1% higher than that of original LR model. For the result of Feature Importance, it shows 12% higher test score and 4% higher AUC. Thus, the latter is the better model and this is my final selected model.

<ANOVA>

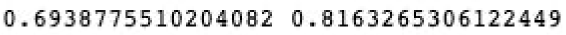
Before After



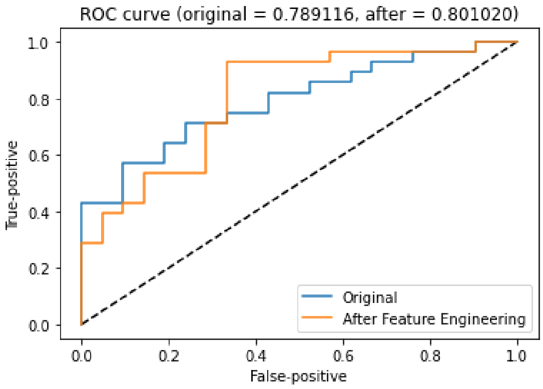
```
0.6938775510204082 0.7346938775510204
```

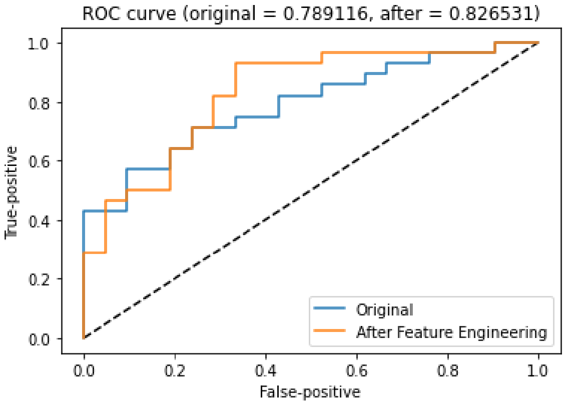
<Feature Importance>

Before After



```
0.6938775510204082 0.8163265306122449
```





Experimental Environment

All my experiments were conducted on following environment:

- Python 3.7.4
- Scikit-learn 0.23.1
- pandas 1.0.5
- numpy 1.19.0
- matplotlib 3.2.2
- seaborn 0.10.1
- missingno 0.4.2