Analysis & Prediction on Wards
- In use of Covid 19 Clinical Dataset

Members: Lilin, Swee Ming, Yuh Jiin

Albert Einstein Isarelite Hospital, Brazil

- Large number of cases daily
- Delay in diagnostic test
- Sharing data with community to find solutions

https://www.kaggle.com/einsteindata4u/covid19

Problem Statement

Healthcare systems have been challenged

COVID-19 cases have overwhelm the health systems around the world with a demand for ICU beds far above the existing capacity.

The need of hospitalised patients who are infected with COVID-19 are complex.

Based on the severity of illness, patients have been placed to a recommended ward (regular, semi-intensive unit, Intensive care unit)

Due to the scarcity, increasing the ICU units is one way to address it, however, one possible approach is to increase the effectiveness of recommending ICU to only those patients who meets the criteria after their laboratory test results.

Objective

By leveraging on the clinical data and laboratory tests, we seek to predict which patients will need to be admitted to a regular ward, semi-intensive unit, or intensive care unit.

COVID-19 Clinical Dataset

There are a total of 5644 rows and 111 columns (From 28th Mar to 3st Apr 2020)

The dataset contains anonymized data from patients seen at the hospital, and who had samples collected to perform the SARS-CoV-2 RT-PCR and additional laboratory tests during a visit to the hospital.

	0	1	2	3	4
Patient ID	44477f75e8169d2	126e9dd13932f68	a46b4402a0e5696	f7d619a94f97c45	d9e41465789c2b5
Patient age quantile	13	17	8	5	15
SARS-Cov-2 exam result	negative	negative	negative	negative	negative
Patient addmited to regular ward (1=yes, 0=no)	0	0	0	0	0
Patient addmited to semi-intensive unit (1=yes, 0=no)	0	0	0	0	0
...	...	...	...	...	...
HCO3 (arterial blood gas analysis)	NaN	NaN	NaN	NaN	NaN
pO2 (arterial blood gas analysis)	NaN	NaN	NaN	NaN	NaN
Arteiral Fio2	NaN	NaN	NaN	NaN	NaN
Phosphor	NaN	NaN	NaN	NaN	NaN
ctO2 (arterial blood gas analysis)	NaN	NaN	NaN	NaN	NaN

Exploratory Data Analysis with Visualization

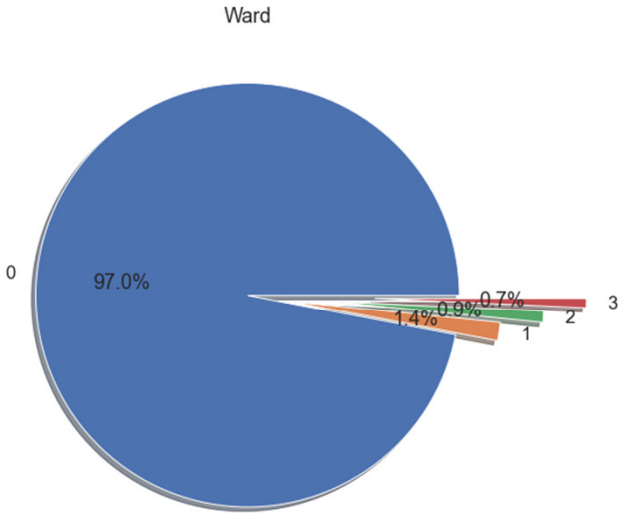
1. What are patients allocation size in different wards?
2. Does all patients with Positive SARS-Cov-2 Exam Results being assigned with a ward?
3. How many patients are warded?
4. Which age quantile has the highest number of patients in the ward?

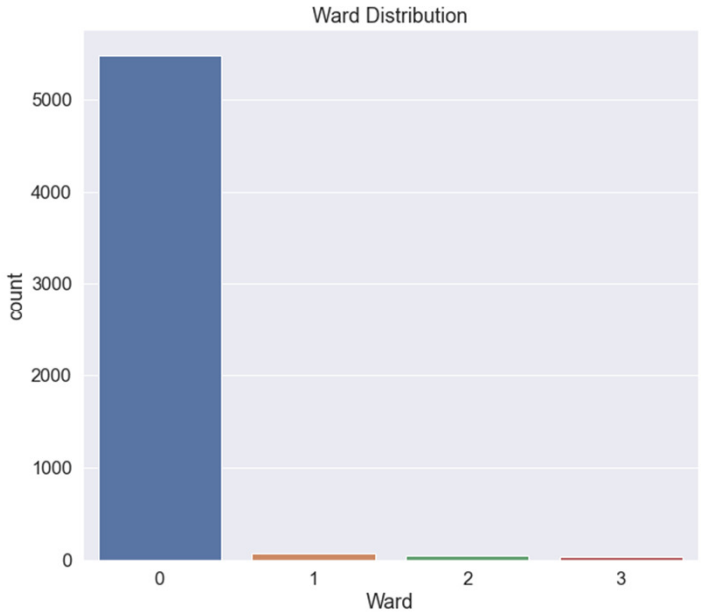
Finding Limitations:

No data provided for the hospital wards capacity,
Only age demographic is available in this data

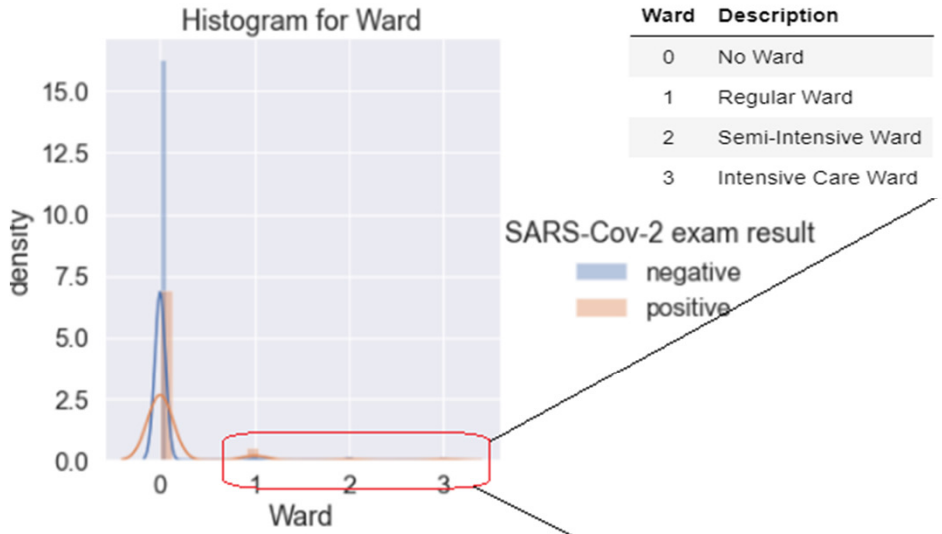
Patients allocation size in different wards

Based on the dataset, there are total 5643 patient recorded, and 97% are not warded, 1.4% are admitted to regular ward, 0.9% are admitted to semi-intensive wards and the remaining 0.7% are admitted to intensive care unit.

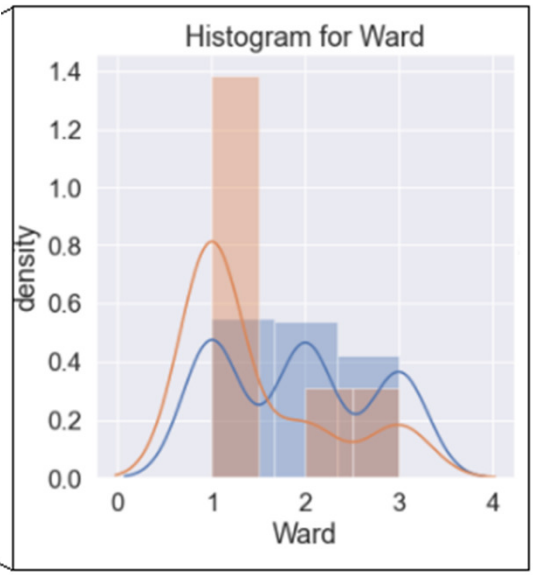




Does all patients with Positive SARS-Cov-2 Exam Results being assigned with a ward?



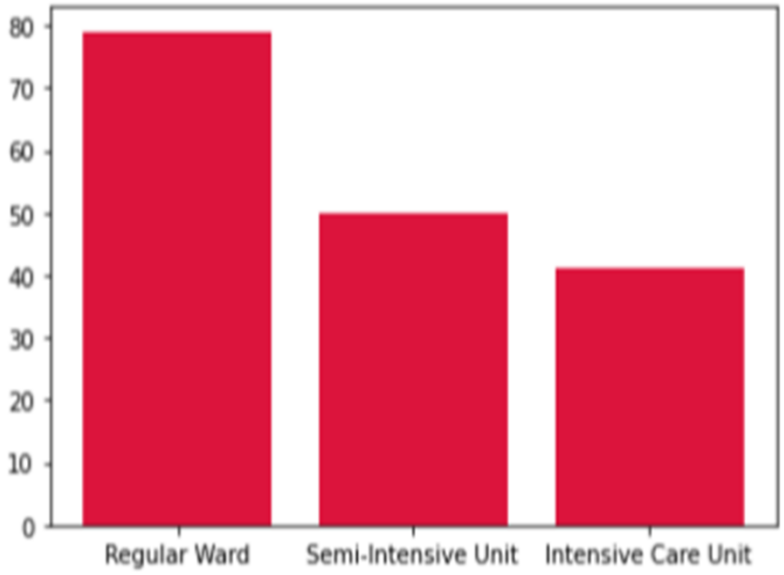
Ward	Description
0	No Ward
1	Regular Ward
2	Semi-Intensive Ward
3	Intensive Care Ward



Most of the patients with positive SARS-Cov-2 exam result, are being placed in the regular ward or not being assigned to a ward.

It suggest that majority of the patients, with positive results, does not have life threatening symptoms.

How many patients are warded?



Observation:

1) Patient admitted to regular ward = 79
   *43 are Covid (+), 36 are Covid (-)

1) Patient admitted to Semi-Intensive Unit = 50
   *8 are Covid (+), 42 are Covid (-)

1) Patient admitted to Intensive Care Unit = 41
   *7 are Covid (+), 34 are Covid (-)

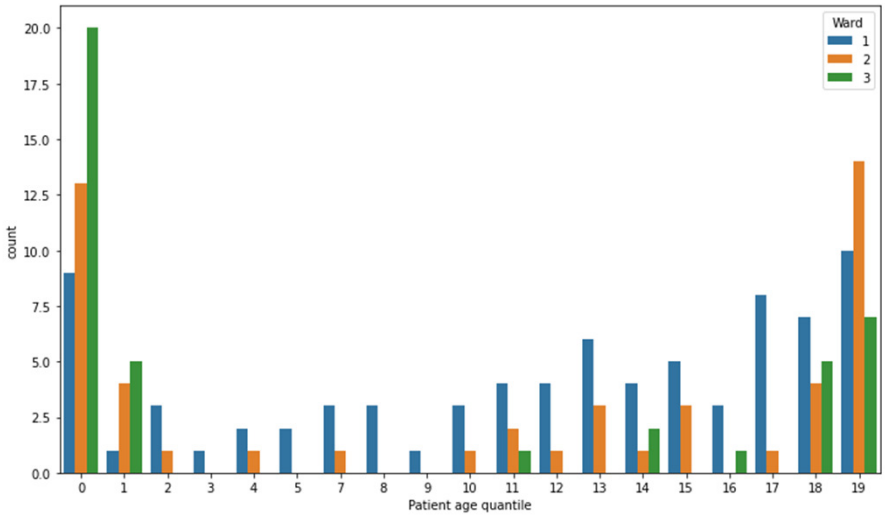
Insights:

Not all ICU warded patient are Covid (+), can be due to other diseases and sickness, and not all visitors to hospital will be warded as well due to severity.

Which age quantile has the highest number of patients in the wards?

What is a Quantile? https://www.statisticshowto.com/quantile-definition-find-easy-steps/

In simple terms, a quantile is where a sample is divided into equal sized, adjacent, subgroups. It can also refer to dividing a probability distribution into areas of equal probability



Observing from the chart,

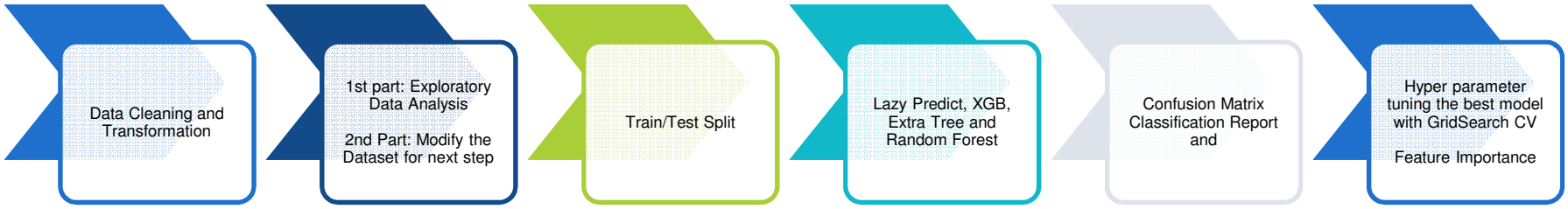
1. 0 age quantiles has the highest number of patients admitting to Ward 0 (ICU).

1. 19 age quantiles has the highest number in regular and semi-intensive.

Insights:

There are mostly young ones and elderly patient are warded, probably due to they are more “fragile” and need more medical attention.

Methodology and Workflow



- Missing Values Analysis
- Drop Columns
- Outliers Observation
- Adding Test Indicators

- Summarize dataset
- Data Visualization
- Observe for pattern
- Fill missing number
- Prepare features to use
- Create new columns with the patterns observed

- Check dataset distribution
- Stratify/Smote imbalanced data
- Justify the splitting method

- Lazy predict to select
- Perform, observe and compare model results
- Conduct oversampling and undersampling

- Evaluate the performance matrix and pick the best model

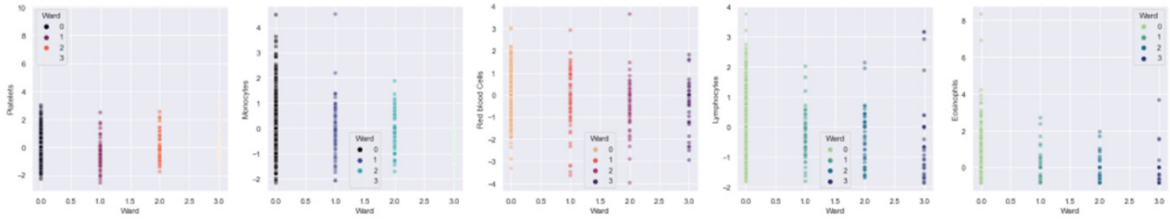
Data Cleaning and Transformation

Observation Stage
- 88% missing values
- Not all in binaries
- Unidentified Lab test

Actions Taken
- Drop unwanted columns and >95% missing values
- Convert to binary
- Grouping Lab Results

Observation : As you can see from the missing number matrix above, you can easily see the missing number distributions. The first 6 columns have datas in all the rows, but it still does not provide us with enough details to decide which columns to drop.

Outliers Observation



Data Cleaning and Transformation

3.5 Add Respiratory Tract Infection test indicator

After Observation, Influenza A,Influenza B and Respiratory Syncytial Virus results appear together.

https://pubmed.ncbi.nlm.nih.gov/18820587/

n [19]:
```
#Set New Column to Zero
covid_new['RTI']=0
# Set Column to One for Rows that fulfil the criteria
covid_new.loc[pd.notnull(covid_new['Influenza A']) & pd.notnull(covid_new['Influenza B'])
              & pd.notnull(covid_new['Respiratory Syncytial Virus']),'RTI']=1
# display updated DataFrame
covid_new.head()
```

ut[19]:

ordetella pertussis	Metapneumovirus	Parainfluenza 2	Neutrophils	Urea	Proteina C reativa mg/dL	Creatinine	Potassium	Sodium	Influenza B, rapid test	Influenza A, rapid test	Strepto A	Ward	CBC	HH	RTI
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0	0
detected	not_detected	not_detected	-0.619086	1.198059	-0.147895	2.089928	-0.305787	0.862512	negative	negative	NaN	0	1	1	1
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0	0
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0	0
detected	not_detected	not_detected	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0	1

3.6 Add Polymerase Chain Reaction(PCR) test indicator

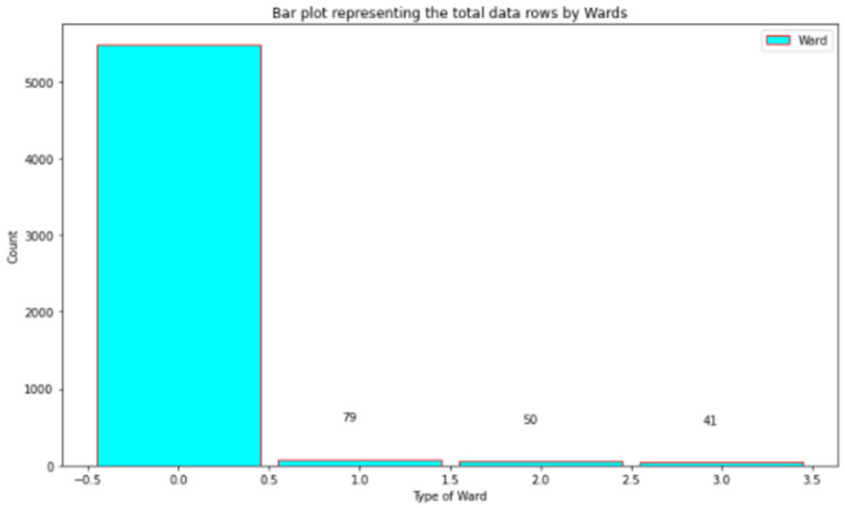
Data Cleaning and Transformation

Ward	1.00
NAT	0.37
KT	0.37
CREAT	0.35
UREAT	0.35
Proteina C reativa mg/dL	0.34
MONT	0.34
CBC	0.34
HH	0.34
MPV	0.34
NEUT	0.23
Hematocrit	0.23
Respiratory Syncytial Virus	0.23
Hemoglobin	0.23
Leukocytes	0.21
PCR	0.21
RTI	0.21
Red blood cell distribution width (RDW)	0.20
Red blood Cells	0.16
Lymphocytes	0.15

Name: Ward, dtype: float64

Observation : The red rectangle box highlights those lab tests are highly correlated with the wards.

Imbalance Data



Ward = 0, n = 5474, Percentage = 96.988%
Ward = 2, n = 50, Percentage = 0.886%
Ward = 1, n = 79, Percentage = 1.400%
Ward = 3, n = 41, Percentage = 0.726%

Stratify split +Smote Train result

In [68]:
```
# target=covid_final['y_test']
counter = Counter(y_test )
for k,v in counter.items():
    per = v / len(target) * 100
    print('Ward = %d, n = %d, Percentage = %.3f%%' % (k, v, per))
```

```
Ward = 0, n = 1095, Percentage = 19.401%
Ward = 2, n = 10, Percentage = 0.177%
Ward = 3, n = 8, Percentage = 0.142%
Ward = 1, n = 16, Percentage = 0.283%
```

Smote result

]:
```
smote = SMOTE()

X_smote, Y_smote = smote.fit_resample(X, y)
# target=covid_final['y_test']
counter = Counter(Y_smote )
for k,v in counter.items():
    per = v / len(target) * 100
    print('Ward = %d, n = %d, Percentage = %.3f%
X_smote_train, X_smote_test, y_smote_train, y_sm
```

```
Ward = 0, n = 5474, Percentage = 96.988%
Ward = 2, n = 5474, Percentage = 96.988%
Ward = 1, n = 5474, Percentage = 96.988%
Ward = 3, n = 5474, Percentage = 96.988%
```

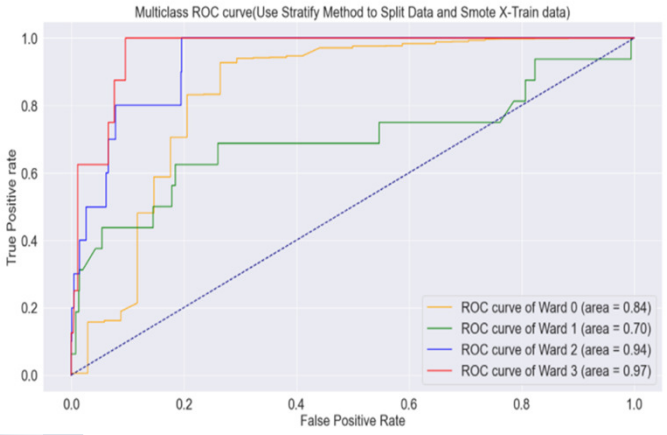
Imbalance Data Con't

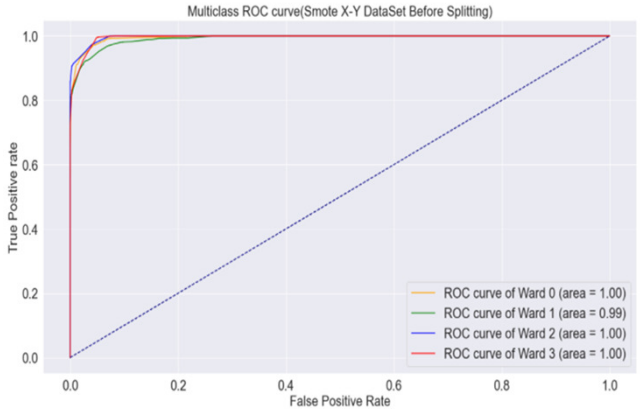
Method Chosen :

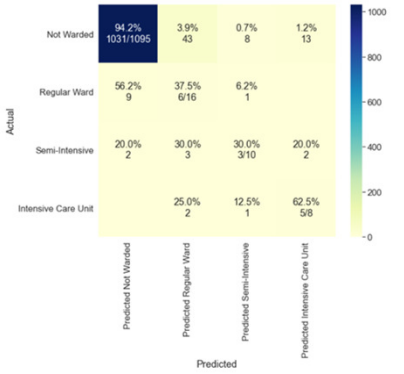
Smote whole dataset before splitting to train and test dataset.

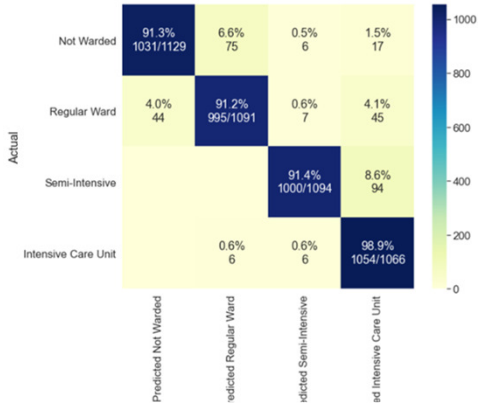
Factor 1 : Improves ROC and AUC score.

Factor 2 : The accuracy improved among the wards.

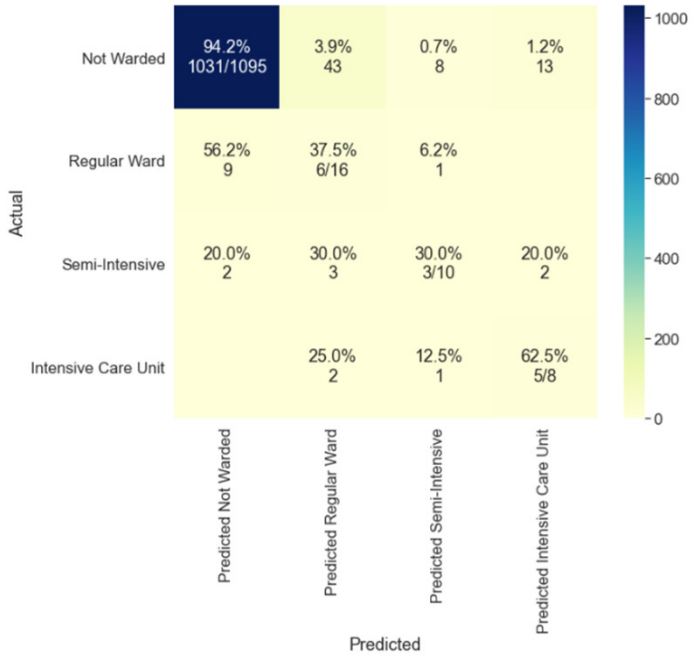


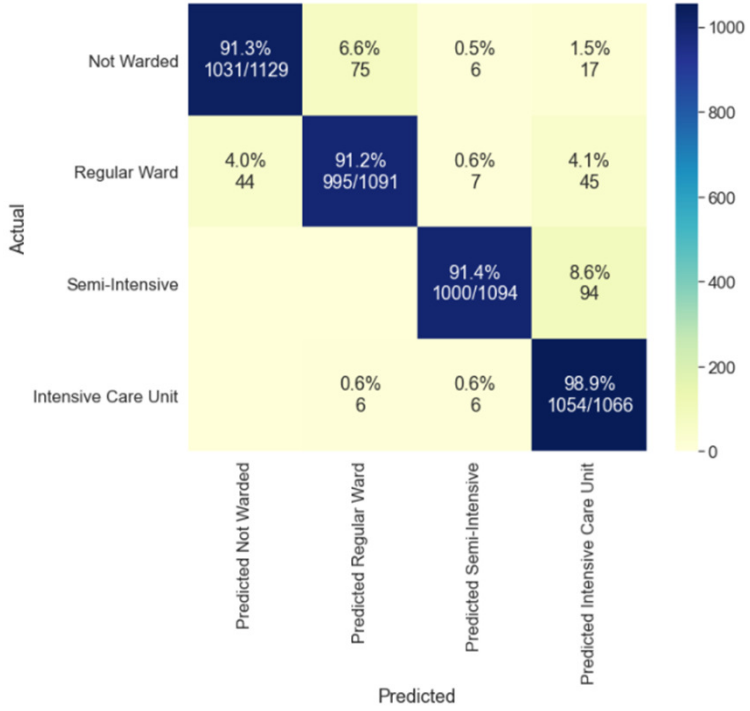






Tell the differences





Model Building

Multi-Class Classification Approach -

- Predict 4 classes - No ward = 0 , Regular Ward = 1, Semi-Intensive Unit = 2, Intensive Care Unit = 3
- Combine all classes into column
- Smote the whole dataset
- Split the Smoted dataset into Train and Test data
- Use Lazy predict to shortlist 3 models to train
- Train selected classifier models
- Use the above classifiers to predict for the test data
- Measure balanced accuracy with Confusion Matrix ,Classification Report (F1 Score, Recall, Precision),ROC AUC graph
- Feature importance

Shortlist Model with Lazy Predict

1. All the results are similar, except for the time taken to complete each prediction.
2. Pick 3 models from the top 4 models that have the best balanced accuracy.

Model	Accuracy	Balanced Accuracy	ROC AUC	F1 Score	Time Taken
XGBClassifier	0.93	0.93	None	0.93	11.33
LGBMClassifier	0.93	0.93	None	0.93	2.62
ExtraTreesClassifier	0.93	0.93	None	0.93	1.44
RandomForestClassifier	0.93	0.93	None	0.93	2.81
BaggingClassifier	0.92	0.92	None	0.92	1.51

XGBoost Classifier

Original Parameter :{"objective" : ['multi:softprob']}

Grid Search Parameter

{"max_depth" : list(range(1,20)), "n_estimators" : list(range(50,401,25)) ,

"objective" : ['multi:softprob'], "eval_metric" : ['mlogloss']}

Best parameters {"max_depth" : 18, "n_estimators" : 400 }

Observation : No improvement is observed, despite the fact that different parameters are used.

Before Tune

	precision	recall	f1-score	support
0	0.95	0.92	0.93	1129
1	0.93	0.90	0.91	1091
2	0.98	0.92	0.95	1094
3	0.87	0.98	0.93	1066
accuracy			0.93	4380
macro avg	0.93	0.93	0.93	4380
weighted avg	0.93	0.93	0.93	4380

Accuracy : 0.9292237442922374
Balanced Accuracy : 0.9296621526079807

After Tune

	precision	recall	f1-score	support
0	0.95	0.92	0.93	1129
1	0.92	0.90	0.91	1091
2	0.98	0.92	0.95	1094
3	0.87	0.98	0.92	1066
accuracy			0.93	4380
macro avg	0.93	0.93	0.93	4380
weighted avg	0.93	0.93	0.93	4380

Accuracy : 0.9292237442922374
Balanced Accuracy : 0.9296621526079807

Extra Trees Classifier

Original Parameter :*No Parameter define*

Grid Search Parameter :

{"criterion" : ['entropy','gini'] , "n_estimators" : list(range(50,401,25)), "min_samples_split" : list(range(1,20))}

Best parameters {'criterion': 'entropy', 'min_samples_split': 2, 'n_estimators': 375}

Observation : There is slight improvement in accuracy and balanced accuracy score.

Before Tune

	precision	recall	f1-score	support
0	0.95	0.91	0.93	1129
1	0.92	0.90	0.91	1091
2	0.98	0.92	0.95	1094
3	0.87	0.98	0.92	1066
accuracy			0.93	4380
macro avg	0.93	0.93	0.93	4380
weighted avg	0.93	0.93	0.93	4380

Accuracy : 0.9257990867579908
Balanced Accuracy : 0.9262979485173

After Tune

	precision	recall	f1-score	support
0	0.95	0.91	0.93	1129
1	0.92	0.90	0.91	1091
2	0.98	0.92	0.95	1094
3	0.87	0.98	0.92	1066
accuracy			0.93	4380
macro avg	0.93	0.93	0.93	4380
weighted avg	0.93	0.93	0.93	4380

Accuracy : 0.9262557077625571
Balanced Accuracy : 0.926755615283948

Improvement:
Accuracy : +0.000457
Balanced Accuracy : +0.000458

Random Forest Classifier

Original Parameter : {criterion="entropy",max_depth=10,n_estimators=10,max_features="auto"}

Grid Search Parameter :

{"criterion" : ['entropy','gini'] , "max_depth" : list(range(1,15)),

'max_features': ['auto','sqrt','log2'], 'n_estimators' :list(range(1,20))}

Best parameters : {'criterion': 'entropy', 'max_depth': 14, 'max_features': 'auto', 'n_estimators': 15}

Observation : All the scores have shown significant improvement.

Before Tune

	precision	recall	f1-score	support
0	0.87	0.89	0.88	1129
1	0.91	0.82	0.86	1091
2	0.96	0.91	0.94	1094
3	0.87	0.97	0.92	1066
accuracy			0.90	4380
macro avg	0.90	0.90	0.90	4380
weighted avg	0.90	0.90	0.90	4380

Accuracy : 0.9
Balanced Accuracy : 0.9004663978679517

After Tune

	precision	recall	f1-score	support
0	0.94	0.90	0.92	1129
1	0.92	0.89	0.90	1091
2	0.97	0.92	0.94	1094
3	0.87	0.98	0.93	1066
accuracy			0.92	4380
macro avg	0.92	0.92	0.92	4380
weighted avg	0.92	0.92	0.92	4380

Accuracy : 0.9223744292237442
Balanced Accuracy : 0.9229188409161369

Improvement:
Accuracy : +0.022374
Balanced Accuracy : +0.022452

Final Model Selection Summary

Select Model : XGBoost Classifier

1. Higher balanced accuracy
2. Higher ROC and AUC score
3. It has higher true positive and lower false positive prediction.

Drop Model : Extra Trees Classifier and Random Forest Classifier

1. Lower balanced accuracy
2. Lower ROC and AUC score than the selected model.

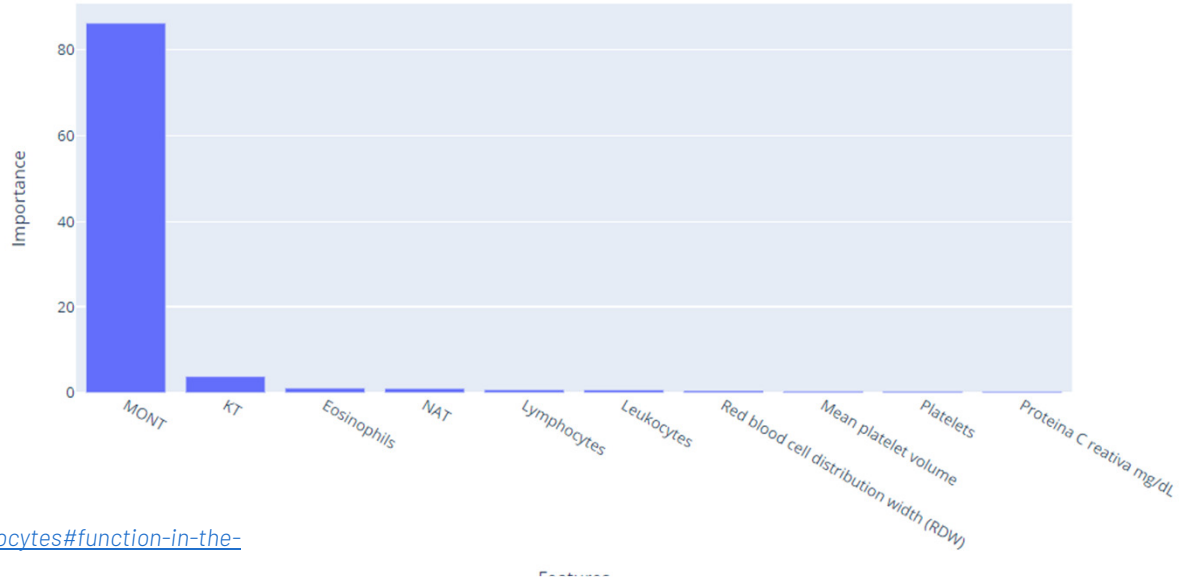
Feature Importance

Monocytes is identified to have the most influence when comes to determine the type of wards for the infected patients.

- Monocytes are a type of leukocyte (white blood cell) and they are powerful defenders that are vital in protecting the body against infection.

This shows that our hypertuned model has given us an accurate result, allowing us to learn about this feature and to investigate into it.

	Features	Importance
54	MONT	86.20
53	KT	3.77
52	Eosinophils	1.06
51	NAT	0.99
50	Lymphocytes	0.70
49	Leukocytes	0.67
48	Red blood cell distribution width (RDW)	0.48
47	Mean platelet volume	0.40
46	Platelets	0.37
45	Proteina C reativa mg/dL	0.35



https://www.healthline.com/health/absolute-monocytes#function-in-the-body

Feature Selection

- The performance of the model generally decreases with the number of selected features, except when the number of features reached 45, which actually increase the accuracy by 0.05%.
    - It is proven that that accuracy improved after reducing the columns to 45 and the auc curve for ward 3 actually improved by 0.01.

Ward	AUC Before Columns Reduction	AUC After Columns Reduction	Variance
0	0.99	0.99	0.00
1	0.99	0.99	0.00
2	1.00	1.00	0.00
3	0.99	1.00	0.01

Ward	Before Columns Reduction	After Columns Reduction	Variance
Accuracy	0.9292237442922374	0.9351598173515981	+0.0059361
Balanced Accuracy	0.9296621526079807	0.9356553371152411	+0.0059932

```
Thresh=0.000, n=55, Accuracy: 92.92%
Thresh=0.000, n=55, Accuracy: 92.92%
Thresh=0.000, n=55, Accuracy: 92.92%
Thresh=0.000, n=55, Accuracy: 92.92%
Thresh=0.000, n=55, Accuracy: 92.92%
Thresh=0.000, n=55, Accuracy: 92.92%
Thresh=0.000, n=55, Accuracy: 92.92%
Thresh=0.000, n=55, Accuracy: 92.92%
Thresh=0.000, n=47, Accuracy: 92.90%
Thresh=0.000, n=46, Accuracy: 92.90%
Thresh=0.000, n=45, Accuracy: 92.95%
Thresh=0.000, n=44, Accuracy: 92.90%
Thresh=0.000, n=43, Accuracy: 92.88%
Thresh=0.000, n=42, Accuracy: 92.83%
Thresh=0.000, n=41, Accuracy: 92.76%
Thresh=0.000, n=40, Accuracy: 92.79%
Thresh=0.000, n=39, Accuracy: 92.24%
Thresh=0.000, n=38, Accuracy: 91.78%
Thresh=0.000, n=37, Accuracy: 90.05%
Thresh=0.000, n=36, Accuracy: 90.14%
Thresh=0.000, n=35, Accuracy: 88.70%
Thresh=0.000, n=34, Accuracy: 88.58%
Thresh=0.001, n=33, Accuracy: 88.20%
Thresh=0.001, n=32, Accuracy: 88.22%
Thresh=0.001, n=31, Accuracy: 88.15%
Thresh=0.001, n=30, Accuracy: 82.49%
Thresh=0.001, n=29, Accuracy: 82.60%
Thresh=0.001, n=28, Accuracy: 82.56%
```

Recap and Conclusions

A quick recap

- Managing large missing values, smote the entire data to tackle the imbalance data, develop multi-class target and understanding viruses and grouping them.
- We saw little to no predictions initially, but after we perform thorough cleaning, it has significantly improves our data quality.
- Lazy Predict has been used to identify machine learning models and evaluating the performance metrics.
- After hyperturne the chosen models, we use Max Voting to combine multiple machines to make prediction and vote the best.
- XGboost is the best model and was hypertuned before conducting Feature Importance. The final result we gotten is 0.93%

If we were to do it again, we will try this approach

- After we had treated the data, we would go with a Bayesian Neural Network (BNN), given that it would take all the inputs into consideration and predict the wards.
- BNN is said to take a probabilistic approach to deep learning that account for uncertainty due to measurement error, noise in the label or insufficient data availability for the model to learn effectively.
- PyMC3 Library to fit BNN and is something to explore and also implement cutting edge inference algorithms.

Thanks!

Any questions?

You can find us at:

- **https://github.com/lilinchen84**
- **https://github.com/limsweeming**
- **https://github.com/auyuhjiin**