



# PROJECT COVID 19

Larissa Santana  
11.09.2020

# STAGES

01

---

## Contextualization

Problem definition and goals.

02

---

## Exploratory Analysis

Analysis and data preprocessing.

03

---

## Classification Model and Results

Description of chosen models and their results.

04

---

## Suggestions

Suggestions for future projects.



# 01 Contextualization

## Problem and goal:

The data are from patients of Albert Einstein Hospital that were in ICU and after physical evaluation took tests for COVID-19.

The goal of this project is using patient data, discover which features are relevant for positive or negative diagnosis of COVID-19.



# 02

## Exploratory Analysis

Type of data:  
Numerical and  
Categorical

Missing data

Feature analysis:  
Inclusion, exclusion or  
modification



# Stages Analysis



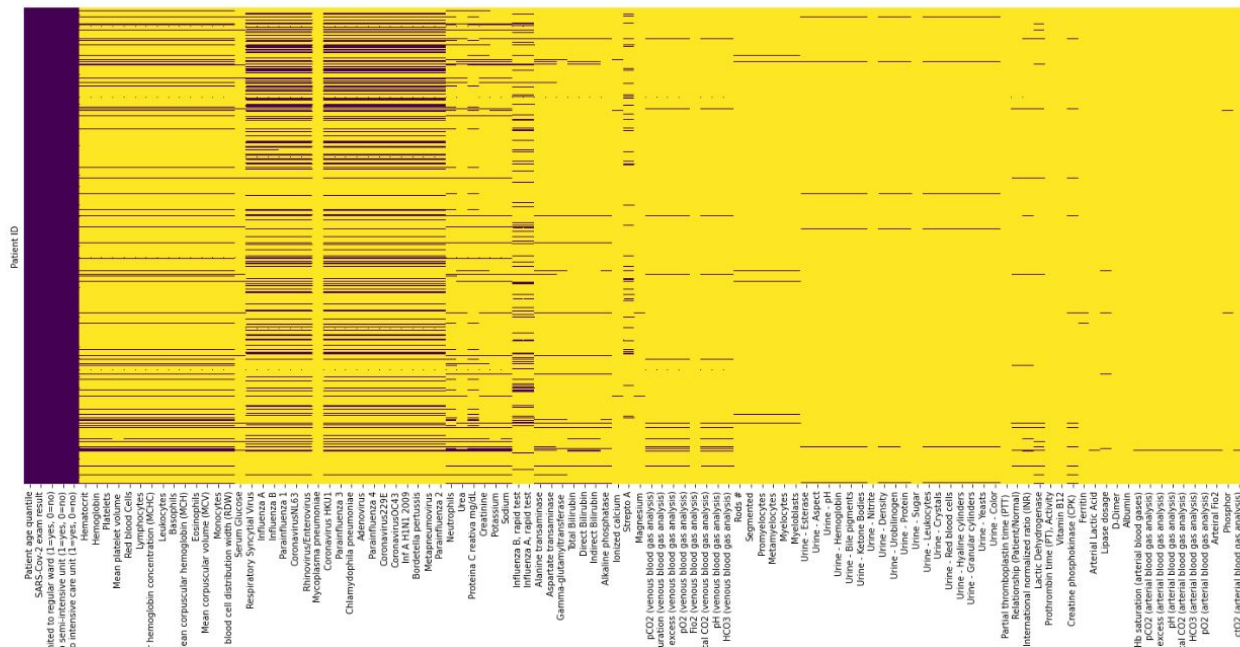
The diagram shows a cluster of white-outlined hexagons on a teal-to-blue gradient background. A horizontal white line passes through the center of the cluster. A small white hexagon with a diamond-shaped arrow pointing upwards is positioned on this line. A small teal dot is located above the text 'STAGE I'.

## STAGE I

Analysis of the number of null  
values.

```
In [194]: plt.figure(figsize=(25,10))
sns.heatmap(train.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

```
Out[194]: <matplotlib.axes._subplots.AxesSubplot at 0x1f4cddb02b0>
```



# Stages Analysis

# Stages Analysis

## STAGE 1

Analysis of the number of null values.

## STAGE 2

Evaluation of the correlation between the number of exams and type of medical care.



```
In [31]: uti = train[train['Patient admitted to intensive care unit (1=yes, 0=no)']==1]
```

```
In [33]: plt.figure(figsize=(25,10))  
sns.heatmap(uti.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

```
Out[33]: <matplotlib.axes._subplots.AxesSubplot at 0x1f4be0bffd0>
```



de exames era  
de internação.

```
In [34]: regular = train[train['Patient admitted to regular ward (1=yes, 0=no)']==1]
```

```
In [37]: plt.figure(figsize=(25,10))  
sns.heatmap(regular.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

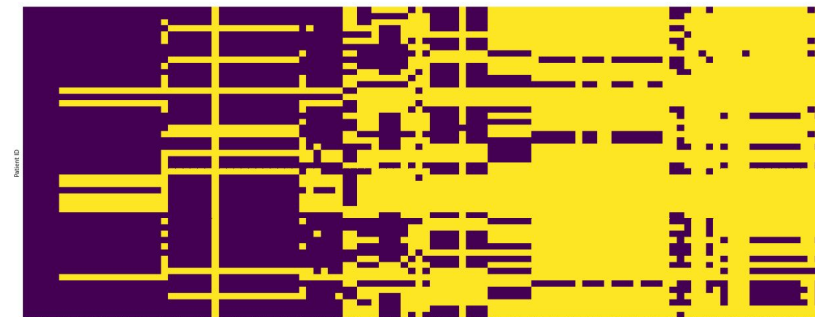
```
Out[37]: <matplotlib.axes._subplots.AxesSubplot at 0x1f4bd0432b0>
```



```
semi = train[train['Patient admitted to semi-intensive unit (1=yes, 0=no)']==1]
```

```
plt.figure(figsize=(25,10))  
sns.heatmap(semi.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1face20ba00>
```



# Stages Analysis



# Stages Analysis

## STAGE 1

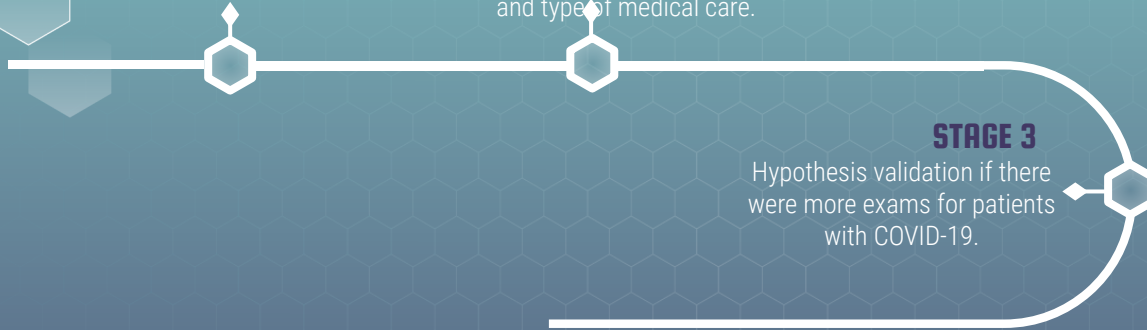
Analysis of the number of null values.

## STAGE 2

Evaluation of the correlation between the number of exams and type of medical care.

## STAGE 3

Hypothesis validation if there were more exams for patients with COVID-19.



# Stages Analysis

```
In [5]: positives = train[train['SARS-Cov-2 exam result']=='positive']['SARS-Cov-2 exam result']  
positives.count()
```

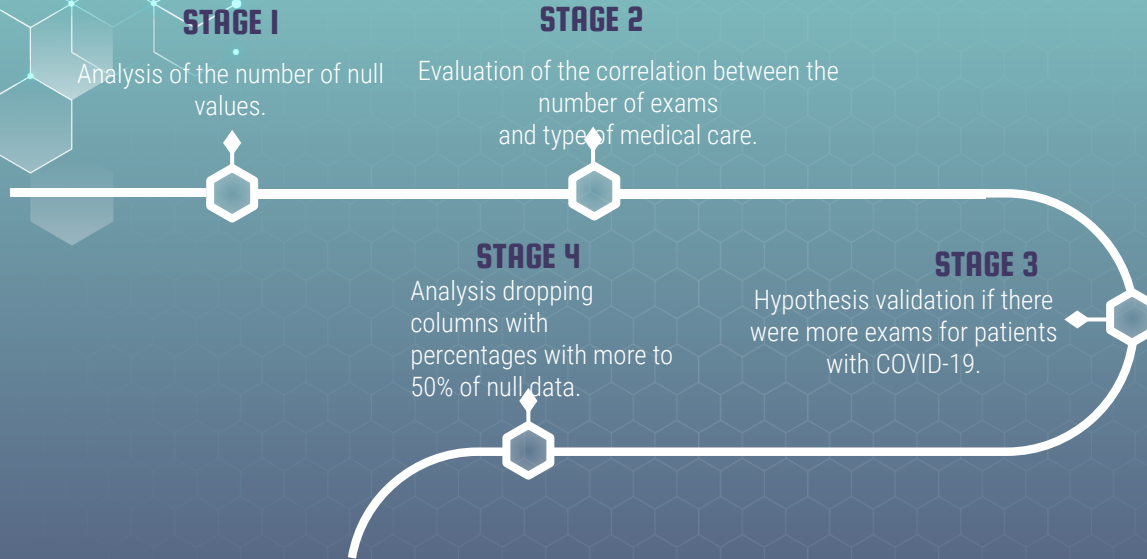
```
Out[5]: 558
```

```
In [198]: #Avaliando se os que são positivos possuem mais informação
```

```
positives = train[train['SARS-Cov-2 exam result']=='positive']  
positives.isnull().sum()
```

```
Out[198]: Patient age quantile                                0  
SARS-Cov-2 exam result                                      0  
Patient admitted to regular ward (1=yes, 0=no)            0  
Patient admitted to semi-intensive unit (1=yes, 0=no)     0  
Patient admitted to intensive care unit (1=yes, 0=no)     0  
Hematocrit                                                  475  
Hemoglobin                                                  475  
Platelets                                                  475  
Mean platelet volume                                       477  
Red blood Cells                                            475  
Lymphocytes                                                475  
Mean corpuscular hemoglobin concentration (MCHC)          475  
Leukocytes                                                 475  
Basophils                                                  475  
Mean corpuscular hemoglobin (MCH)                         475  
Eosinophils                                                475  
Mean corpuscular volume (MCV)                             475  
Monocytes                                                  475  
Red blood cell distribution width (RDW)                   475  
Serum Glucose                                              525
```

# Stages Analysis



# Stages Analysis

```
In [75]: percent_missing = train.isnull().sum() * 100 / len(train)
```

```
In [76]: percent_missing_50 = percent_missing > 50
```

```
In [80]: percent_missing_50
```

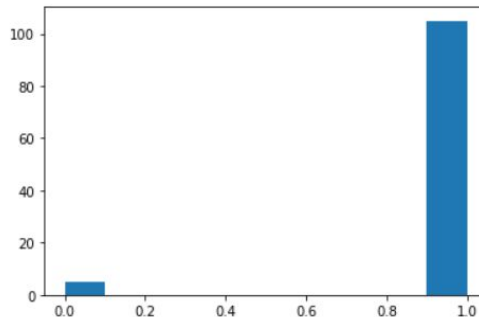
```
Out[80]: Patient age quantile                False
SARS-Cov-2 exam result                    False
Patient admitted to regular ward (1=yes, 0=no)  False
Patient admitted to semi-intensive unit (1=yes, 0=no) False
Patient admitted to intensive care unit (1=yes, 0=no) False
...
HCO3 (arterial blood gas analysis)          True
pO2 (arterial blood gas analysis)           True
Arterial Fio2                             True
Phosphor                                    True
ctO2 (arterial blood gas analysis)          True
Length: 110, dtype: bool
```

```
In [87]: def boolstr_to_floatstr(v):
          if v == 'True':
              return '1'
          elif v == 'False':
              return '0'
          else:
              return v
```

```
In [88]: new_percent_missing_50 = np.vectorize(boolstr_to_floatstr)(percent_missing_50).astype(float)
```

```
In [91]: plt.hist(new_percent_missing_50)
```

```
Out[91]: (array([ 5.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0., 105.]),
          array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
          <a list of 10 Patch objects>)
```



```
In [106]: data = percent_missing_50.to_frame(name='Dados')
```

# Stages Analysis

```
In [160]: pd.set_option("display.max.rows", None)
data_true = data[data['Dados']==True]
data_true
```

```
Out[160]:
```

	Dados
Hematocrit	True
Hemoglobin	True
Platelets	True
Mean platelet volume	True
Red blood Cells	True
Lymphocytes	True
Mean corpuscular hemoglobin concentration (MCHC)	True
Leukocytes	True
Basophils	True
Mean corpuscular hemoglobin (MCH)	True
Eosinophils	True
Mean corpuscular volume (MCV)	True

```
In [161]: data_true.count()
```

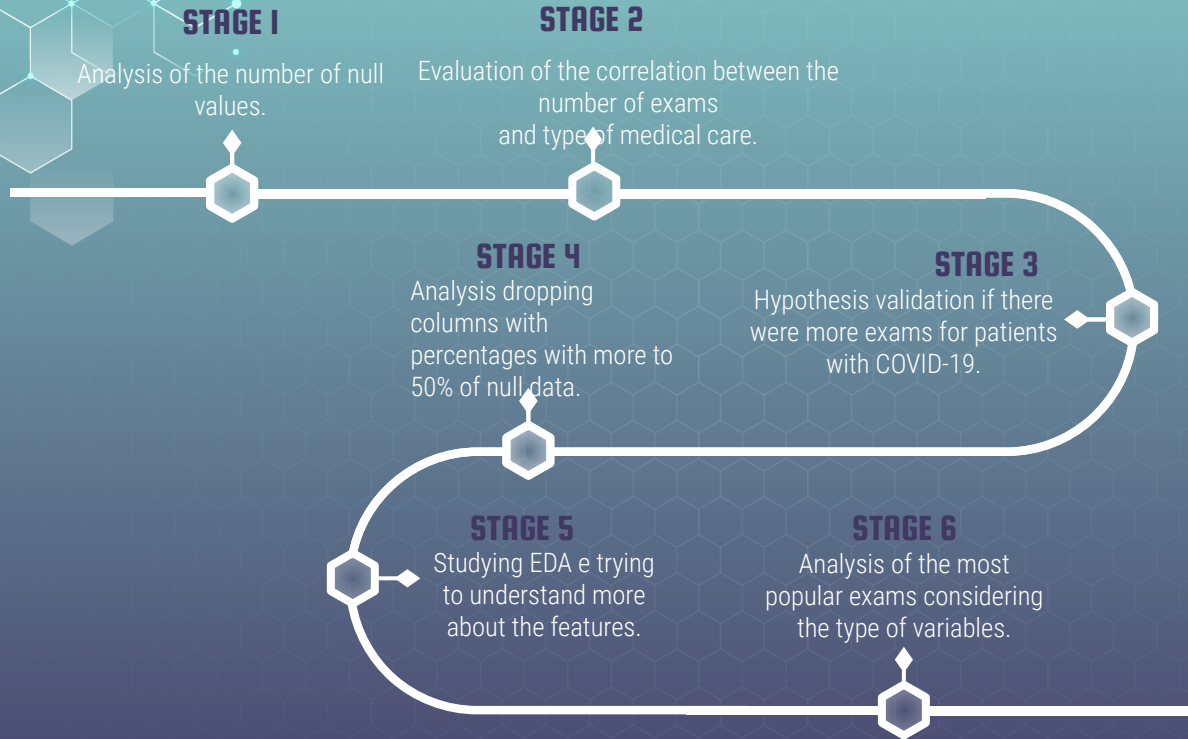
```
Out[161]: Dados    105
dtype: int64
```

```
In [187]: new_train
```

```
Out[187]:
```

Patient ID	Patient age quantile	SARS-Cov-2 exam result	Patient addmitted to regular ward (1=yes, 0=no)	Patient addmitted to semi-intensive unit (1=yes, 0=no)	Patient addmitted to intensive care unit (1=yes, 0=no)
44477f75e8169d2	13	negative	0	0	0
126e9dd13932f68	17	negative	0	0	0
a46b4402a0e5696	8	negative	0	0	0
f7d619a94f97c45	5	negative	0	0	0
d9e41465789c2b5	15	negative	0	0	0
75f16746216c4d1	9	negative	0	0	0
2a2245e360808d7	13	negative	0	0	0
509197ec73f1400	16	negative	0	0	0
8bb9d64f0215244	1	negative	0	1	0
5f1ed301375586c	17	negative	0	0	0

# Stages Analysis



# Stages Analysis

```
In [20]: numerical = train.select_dtypes(include=[np.number])
numerical.isnull().sum().head(40)
```

*#Conseguí ver quais são os testes que tem menos nulos, agora preciso ver se esses testes são aplicados nas mesmas pessoas*

```
Out[20]: Patient age quantile                                0
Patient addmitted to regular ward (1=yes, 0=no)           0
Patient addmitted to semi-intensive unit (1=yes, 0=no)    0
Patient addmitted to intensive care unit (1=yes, 0=no)     0
Hematocrit                                                 5041
Hemoglobin                                                 5041
Platelets                                                  5042
Mean platelet volume                                       5045
Red blood Cells                                           5042
Lymphocytes                                               5042
Mean corpuscular hemoglobin concentration (MCHC)          5042
Leukocytes                                                5042
Basophils                                                 5042
Mean corpuscular hemoglobin (MCH)                        5042
Eosinophils                                               5042
Mean corpuscular volume (MCV)                            5042
Monocytes                                                 5043
Red blood cell distribution width (RDW)                  5042
Serum Glucose                                             5436
Mycoplasma pneumoniae                                    5644
Neutrophils                                              5131
Urea                                                     5247
Proteína C reativa mg/dL                                5138
Creatinine                                               5220
Potassium                                                5273
Sodium                                                   5274
Alanine transaminase                                     5419
Aspartate transaminase                                   5418
Gamma-glutamyltransferase                               5491
Total Bilirubin                                          5462
Direct Bilirubin                                         5462
Indirect Bilirubin                                       5462
```

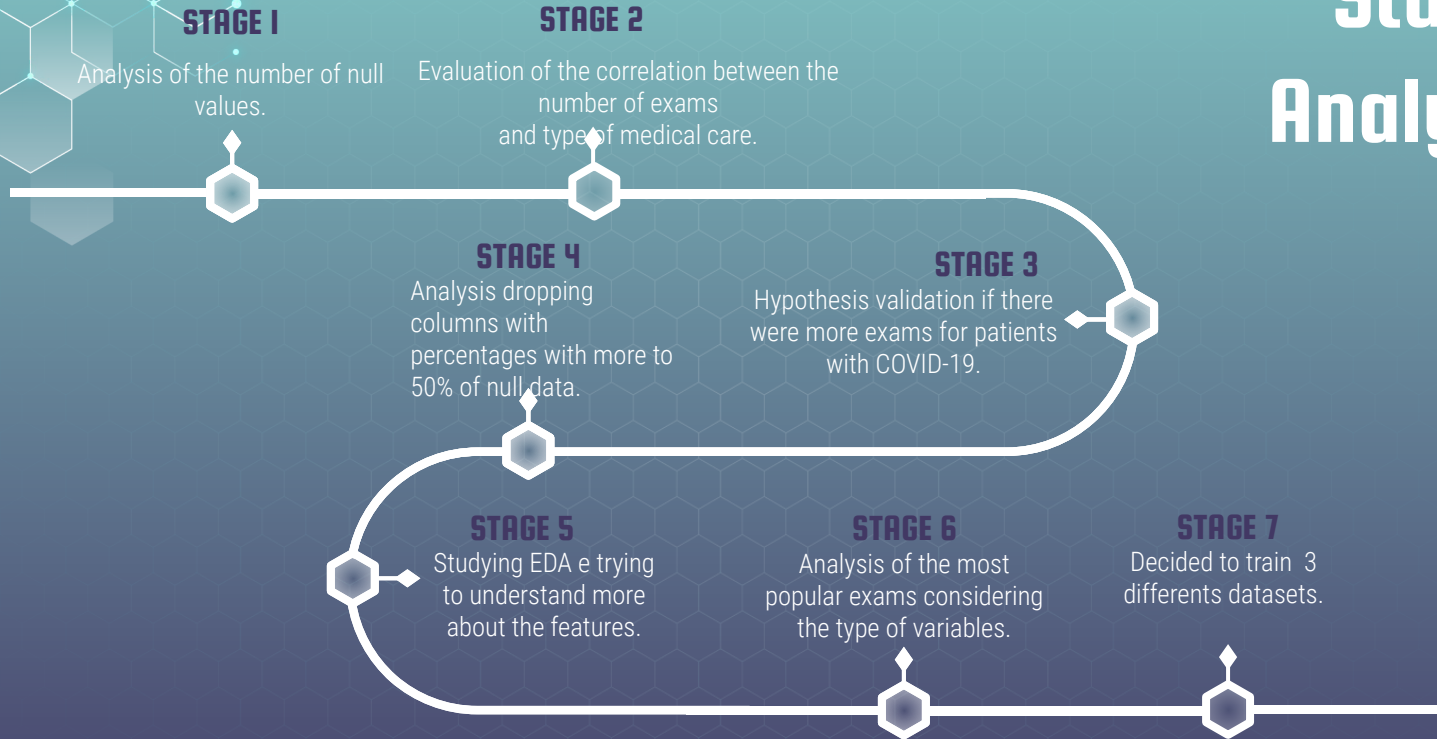
```
In [7]: categorical = train.select_dtypes(include=['object'])
categorical.isnull().sum()
```

*#Assim eu consigo ver quais testes são feitos juntos, posso dropar os de urina, influenza A e B Rapid Test e Strepto A*

```
Out[7]: SARS-Cov-2 exam result                            0
Respiratory Syncytial Virus                             4290
Influenza A                                              4290
Influenza B                                              4290
Parainfluenza 1                                         4292
CoronavirusNL63                                         4292
Rhinovirus/Enterovirus                                 4292
Coronavirus HKU1                                         4292
Parainfluenza 3                                         4292
Chlamydomphila pneumoniae                             4292
Adenovirus                                              4292
Parainfluenza 4                                         4292
Coronavirus229E                                         4292
CoronavirusOC43                                         4292
Inf A H1N1 2009                                         4292
Bordetella pertussis                                    4292
Metapneumovirus                                         4292
Parainfluenza 2                                         4292
Influenza B, rapid test                                4824
Influenza A, rapid test                                4824
Strepto A                                               5312
Urine - Esterase                                        5584
Urine - Aspect                                          5574
Urine - pH                                              5574
Urine - Hemoglobin                                      5574
Urine - Bile pigments                                  5574
Urine - Ketone Bodies                                  5587
Urine - Nitrite                                         5643
Urine - Urobilinogen                                    5575
Urine - Protein                                         5584
Urine - Leukocytes                                      5574
Urine - Crystals                                       5574
Urine - Hyaline cylinders                              5577
Urine - Granular cylinders                             5575
Urine - Yeasts                                          5574
Urine - Color                                           5574
```



# Stages Analysis





# Etapas

# Análises

```
In [4]: def drop_cols(df):
        for column in df:
            if df[column].isnull().sum()>5043:
                train.drop(column,axis=1, inplace=True)
        return df
```

Atenção a quantidade de valores

Atenção a quantidade de exames tra

```
In [27]: numerical = df.select_dtypes(include=[np.number])
        numerical_final = numerical.dropna(axis=0)
        numerical_final
```

Out[27]:

Patient ID	Patient age quantile	Patient admitted to regular ward (1=yes, 0=no)	Patient admitted to semi-intensive unit (1=yes, 0=no)	Patient admitted to intensive care unit (1=yes, 0=no)	Hematocrit	Hemoglobin	Platelets	Red blood Cells	Lymphocytes	Mean corpuscular hemoglobin concentration (MCHC)	Leukocytes	Ba
126e9dd13932f68	17	0	0	0	0.236515	-0.022340	-0.517413	0.102004	0.318366	-0.950790	-0.094610	-0.
8bb9d64f0215244	1	0	1	0	-1.571682	-0.774212	1.429667	-0.850035	-0.005738	3.331071	0.364550	-0.
6c9d3323975b082	9	0	0	0	-0.747693	-0.586244	-0.429480	-1.361315	-1.114514	0.542882	-0.884923	0.
d3ea751f3db9de9	11	0	0	0	0.991838	0.792188	0.072992	0.542763	0.045436	-0.452899	-0.211488	-0.
2c2eae16c12a18a	9	0	0	0	0.190738	-0.147652	-0.668155	-0.127191	0.002791	-1.249524	-1.132592	0.
...	...	...	...	...	...	...	...	...	...	...	...	...
c5b44ff9c7782fd	19	0	0	0	0.190738	0.165628	-0.102873	0.384090	-1.583611	-0.054585	-0.328365	-0.
88cce1444e16f9c	19	0	0	0	-0.289922	-0.523588	0.663397	0.754327	-1.532437	-1.050367	1.569499	0.
2733fac0d3f7138	15	0	0	0	0.717175	1.105468	-0.492289	0.613284	0.002791	1.538664	-0.550988	-0.
acd761fe16b5d0f	17	0	0	0	-3.242548	-2.779203	-1.773594	-3.318285	-1.830953	1.538664	-1.733675	-1.
2697fdccbf7f7f	19	0	0	0	0.694287	0.541564	-0.906829	0.578024	-0.295726	-0.353319	-1.288428	-1.

601 rows × 19 columns

## ETAPA 7

treinar 3 datasets diferentes



```
In [28]: categorical = df.select_dtypes(include=['object'])
categorical_final = categorical.dropna(axis=0)
categorical_final
```

Out[28]:

Patient ID	SARS-Cov-2 exam result	Respiratory Syncytial Virus	Influenza A	Influenza B	Parainfluenza 1	CoronavirusNL63	Rhinovirus/Enterovirus	Coronavirus HKU1	Parainfluenza 3	Chlamy pneumoniae
126e9dd13932f68	negative	not_detected	not_detected	not_detected	not_detected	not_detected	detected	not_detected	not_detected	not_detected
6c9d3323975b082	negative	not_detected	not_detected	not_detected	not_detected	not_detected	not_detected	not_detected	not_detected	not_detected
fe656baa2bfc5dd	negative	detected	not_detected	not_detected	not_detected	not_detected	not_detected	not_detected	not_detected	not_detected
ebdd7c67fcb21b4	negative	not_detected	not_detected	detected	not_detected	not_detected	not_detected	not_detected	not_detected	not_detected
01d324f278f3101	negative	not_detected	not_detected	detected	not_detected	not_detected	not_detected	not_detected	not_detected	not_detected
...	...	...	...	...	...	...	...	...	...	...
cfa1522418dc528	negative	detected	not_detected	not_detected	not_detected	not_detected	not_detected	not_detected	not_detected	not_detected
0edc1a366792b62	negative	not_detected	not_detected	not_detected	not_detected	not_detected	detected	not_detected	not_detected	not_detected
c19b361d6ab2051	negative	detected	not_detected	not_detected	not_detected	not_detected	not_detected	not_detected	not_detected	not_detected
32f55e529808065	positive	not_detected	not_detected	not_detected	not_detected	not_detected	not_detected	not_detected	not_detected	not_detected
f1c30ed80e63858	positive	not_detected	not_detected	not_detected	not_detected	not_detected	not_detected	not_detected	not_detected	not_detected

265 rows × 20 columns

```
In [9]: both = df.dropna()
both
```

126e9dd13932f68	17	negative	0	0	0	0.236515	-0.022340	-0.517413	0.102004	0.318366	...	not_detected	not_detected
6c9d3323975b082	9	negative	0	0	0	-0.747693	-0.586244	-0.429480	-1.361315	-1.114514	...	not_detected	not_detected
ebdd7c67fcb21b4	9	negative	1	0	0	-0.679027	-0.711556	0.952319	-0.321124	-0.875701	...	not_detected	not_detected
01d324f278f3101	16	negative	0	0	0	0.671398	0.290940	0.135801	0.525133	0.173372	...	not_detected	not_detected
54ea170f12c4e76	9	negative	0	0	0	-0.656139	-0.899524	-0.391795	-0.409276	1.862123	...	not_detected	not_detected
...	...	...	...	...	...	...	...	...	...	...	...	...	...
d584d2cf7d09e1e	11	positive	0	0	1	-0.450142	0.040316	-0.492289	-0.409276	-1.378914	...	not_detected	not_detected
0edc1a366792b62	0	negative	0	0	1	-2.418559	-2.152643	0.952319	-1.237902	-1.694489	...	not_detected	not_detected
c19b361d6ab2051	0	negative	0	0	1	-1.182576	-0.836868	-0.693278	-0.462168	-0.671003	...	not_detected	not_detected
32f55e529808065	12	positive	1	0	0	1.152058	0.604220	-0.529975	0.930631	-0.679533	...	not_detected	not_detected
f1c30ed80e63858	14	positive	1	0	0	1.106281	1.042812	-0.253615	0.930631	-0.483364	...	not_detected	not_detected

99 rows × 37 columns

# Etapas Análises



## 02 Exploratory Analysis

### Conclusion:

As a result, I have 3 datasets, which I will work on. My decision not to include nulls in all features was based on the fact that there is a lot of missing data and this could interfere with the model. Also, there are features that I don't know about, so I could include wrong information. The downside of this is that I am excluding features that could be relevant in the diagnosis of COVID, even though they are not commonly done by doctors.





## 02 Exploratory Analysis

### Conclusion:

After the definition of the datasets, I did data transformations that the machine learning model needs:

- Analysis of the type of medical care influences the number of positives or negatives - No influence, drop feature.



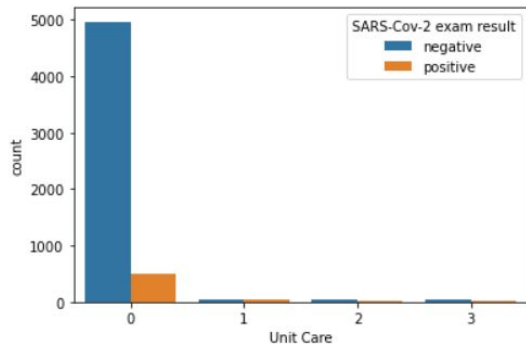
```
In [10]: # Colocando em uma coluna única em qual internação o paciente está
# 0 - None
# 1 - Regular
# 2 - Semi Intensive
# 3 - Intensive

conditions = [(train['Patient addmitted to regular ward (1=yes, 0=no)']==1),
              (train['Patient addmitted to semi-intensive unit (1=yes, 0=no)']==1),
              (train['Patient addmitted to intensive care unit (1=yes, 0=no)']==1),
              (train['Patient addmitted to regular ward (1=yes, 0=no)']==0) & (train['Patient addmitted to semi-intensive unit (1=yes, 0=no)']==1)]
values = [1,2,3,0]

train['Unit Care'] = np.select(conditions,values)

sns.countplot('Unit Care', hue='SARS-Cov-2 exam result', data = train)
```

```
Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x18e67874d00>
```



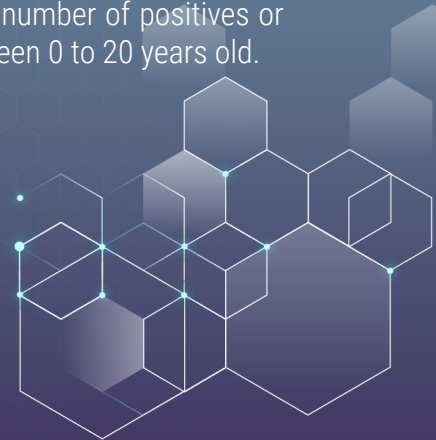


## 02 Exploratory Analysis

### Conclusion:

After the definition of the datasets, I did data transformations that the machine learning model needs:

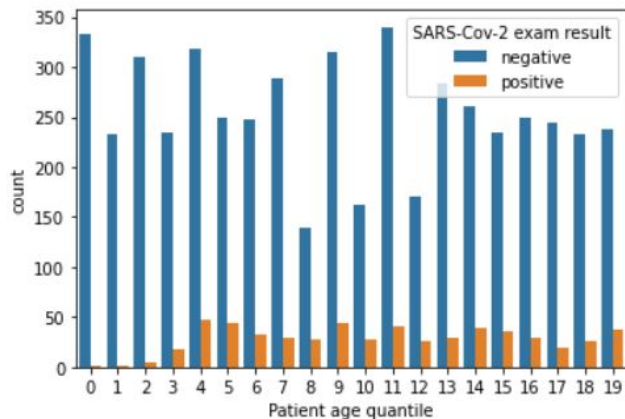
- Analysis of the type of medical care influences the number of positives or negatives - No influence, drop feature.
- Analysis of the age influences the number of positives or negatives - Influence, mostly between 0 to 20 years old.





```
In [17]: sns.countplot('Patient age quantile', hue='SARS-Cov-2 exam result', data = train)
```

```
Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x18e67ccb0a0>
```

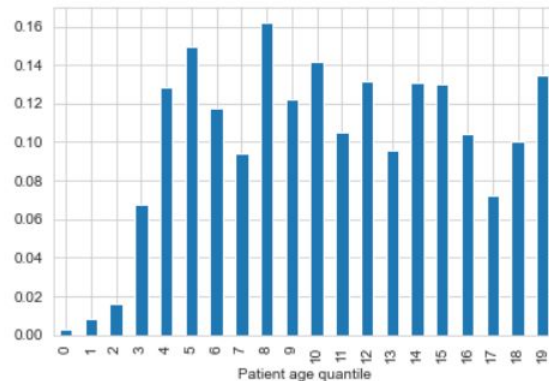


realizei as transformações

```
In [157]: #Influencia da Idade no número de Positivos
```

```
Percentil_Positive.plot.bar()
```

```
Out[157]: <matplotlib.axes._subplots.AxesSubplot at 0x20638cd18b0>
```

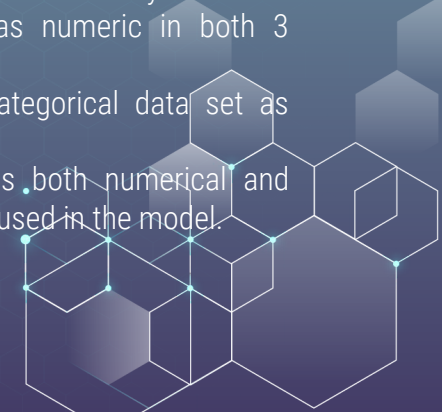




## 02 Exploratory Analysis

### Conclusion:

After the definition of the datasets, I did data transformations that the machine learning model needs:

- Analysis of the type of medical care influences the number of positives or negatives - No influence, drop feature.
  - Analysis of the age influences the number of positives or negatives - Influence, mostly between 0 to 20 years old.
  - I transform the target feature as numeric in both 3 datasets.
  - I transform the features of a categorical data set as numerical.
  - I transform the dataset that has both numerical and categorical data to be ready to be used in the model.
- 



In [159]: *#Início Transformação Variváveis Categorias*

In [20]: *#Transformando em Números as categorias*

```
def cat_to_num(item):  
    if item == 'not_detected' or item == 'negative':  
        return 0  
    elif item == 'positive' or item == 'detected':  
        return 1  
    else:  
        return item
```

In [33]: categorical\_final\_test = categorical\_final  
Data\_Categorical = pd.DataFrame()

```
for column in categorical_final_test:  
    name = str(column)  
    Data_Categorical[name]= categorical_final_test[column].apply(cat_to_num)  
  
Data_Categorical
```

datasets;

- Transformei as features do dataset categorical em

In [54]: *#Criando Saídas em CSV dos Datasets*


```
Data_NumandCat.to_csv('Dataset_Num_Cat.csv',index=False)  
Data_Categorical.to_csv('Dataset_Categorical.csv',index=False)  
Data_Numerical.to_csv('Dataset_Numerical.csv',index=False)
```



03

# Classification Model

As the datasets were small and unbalanced, I decided to use the logistic regression method and then evaluate improvements from it for my case.



# RESULTS – Numerical Dataset

96% 😊 50% ☹️

Precision

90%  
Accuracy

True  
Negatives

True  
Positives

Predicted  
Negatives

Predicted  
Positives

150

12

7

12

# RESULTS – Numerical Dataset

In [54]: *#Feature Importance*

In [32]: *# get importance*

```
importance = logmodel.coef_[0]
```

```
# summarize feature importance
```

```
for i,v in enumerate(importance):
```

```
    print(str(list(Data_Numerical.columns.values)[i]) + ' Score: %.5f' % (v))
```

Patient age quantile Score: 0.10708

Hematocrit Score: 0.12439

Hemoglobin Score: 0.07165

Platelets Score: -0.50992

Red blood Cells Score: 0.11821

Lymphocytes Score: -0.00502

Mean corpuscular hemoglobin concentration (MCHC) Score: -0.04207

Leukocytes Score: -1.23292

Basophils Score: -0.09168

Mean corpuscular hemoglobin (MCH) Score: -0.06136

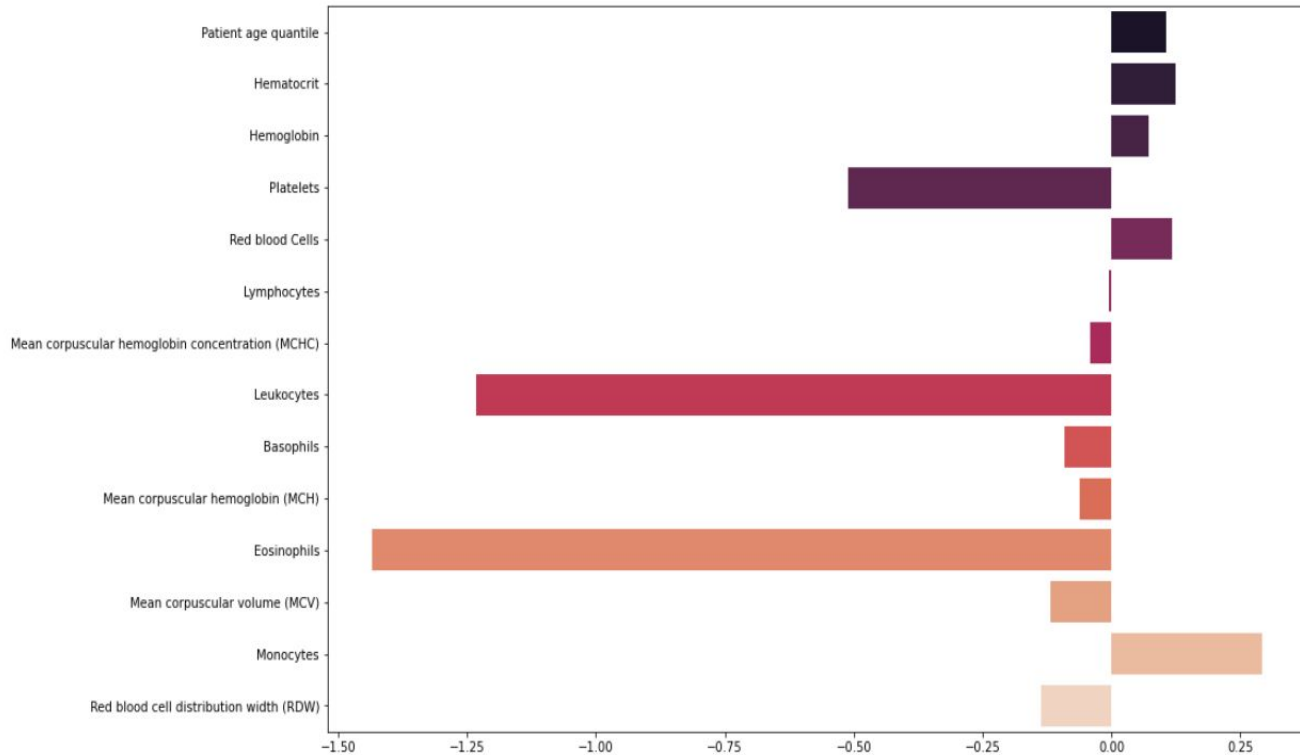
Eosinophils Score: -1.43405

Mean corpuscular volume (MCV) Score: -0.11913

Monocytes Score: 0.29231

Red blood cell distribution width (RDW) Score: -0.13605

# RESULTS – Numerical Dataset



# RESULTS – Categorical Dataset

91% 😊 0% ☹️

Precision

91%  
Accuracy

True  
Negatives

True  
Positives

Predicted  
Negatives

Predicted  
Positives

48

0

5

0

# RESULTS – Categorical Dataset

```
In [15]: # get importance

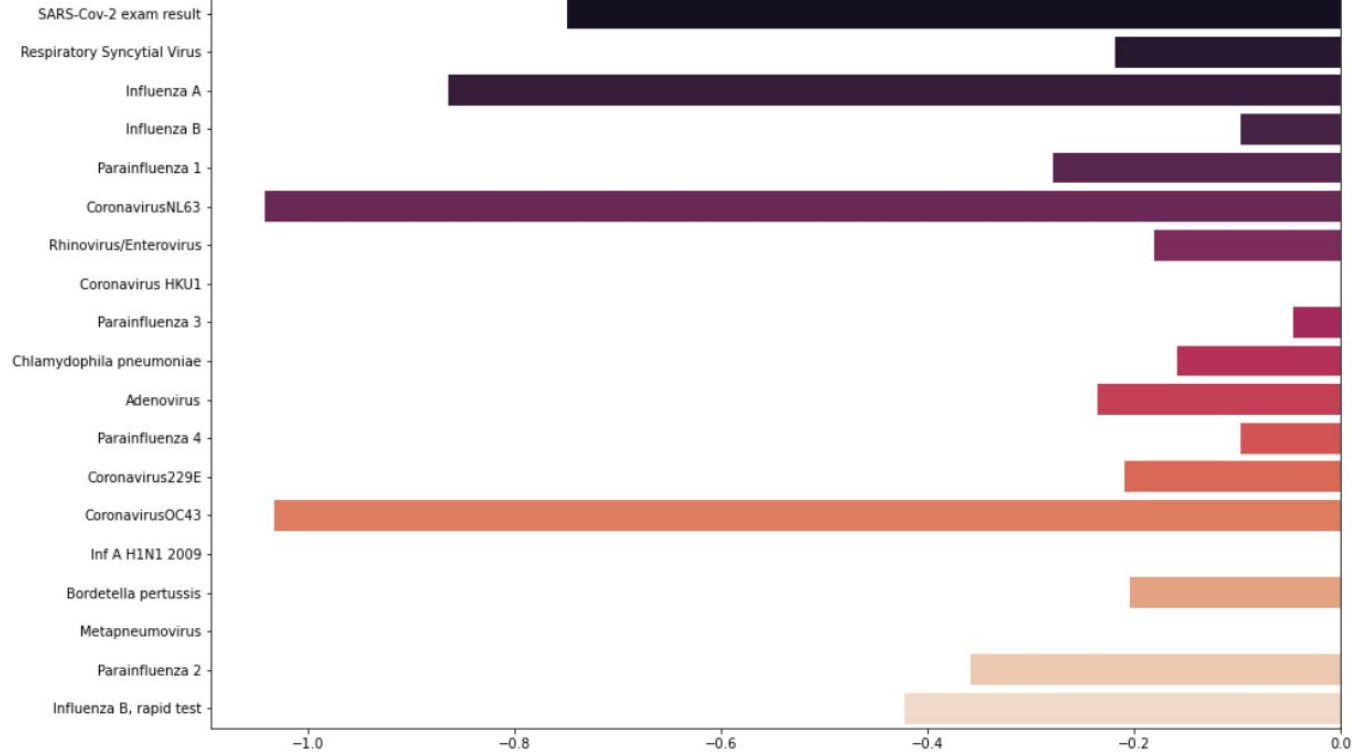
importance = logmodel.coef_[0]

# summarize feature importance
for i,v in enumerate(importance):
    print(str(list(Data_Categorical.columns.values)[i]) + ' Score: %.5f' % (v))
```

```
SARS-Cov-2 exam result Score: -0.74861
Respiratory Syncytial Virus Score: -0.21840
Influenza A Score: -0.86408
Influenza B Score: -0.09741
Parainfluenza 1 Score: -0.27879
CoronavirusNL63 Score: -1.04166
Rhinovirus/Enterovirus Score: -0.18076
Coronavirus HKU1 Score: 0.00000
Parainfluenza 3 Score: -0.04573
Chlamydomphila pneumoniae Score: -0.15840
Adenovirus Score: -0.23611
Parainfluenza 4 Score: -0.09741
Coronavirus229E Score: -0.20913
CoronavirusOC43 Score: -1.03198
Inf A H1N1 2009 Score: 0.00000
Bordetella pertussis Score: -0.20465
Metapneumovirus Score: 0.00000
Parainfluenza 2 Score: -0.35877
Influenza B, rapid test Score: -0.42245
```



# RESULTS – Categorical Dataset





# RESULTS – Mix Dataset

88% 😊 40% ☹️

Precision

80%  
Accuracy

True  
Negatives

True  
Positives

Predicted  
Negatives

Predicted  
Positives

22

3

3

2



# RESULTS – Mix Dataset

In [13]: *# get importance*

```
importance = logmodel.coef_[0]

# summarize feature importance
for i,v in enumerate(importance):
    print(str(list(Data_NumandCat.columns.values)[i]) + ' Score: %.5f' % (v))
```

Patient age quantile Score: 0.12337  
Hematocrit Score: 0.07098  
Hemoglobin Score: 0.07982  
Platelets Score: -0.17702  
Red blood Cells Score: -0.02327  
Lymphocytes Score: -0.39796  
Mean corpuscular hemoglobin concentration (MCHC) Score: 0.14289  
Leukocytes Score: -1.22951  
Basophils Score: 0.24559  
Mean corpuscular hemoglobin (MCH) Score: 0.18691  
Eosinophils Score: -0.55137  
Mean corpuscular volume (MCV) Score: 0.21223  
Monocytes Score: 0.35273  
Red blood cell distribution width (RDW) Score: -0.18024  
Respiratory Syncytial Virus Score: -0.01868  
Influenza A Score: 0.00000  
Influenza B Score: -0.30123  
Parainfluenza 1 Score: 0.00000  
CoronavirusNL63 Score: -0.18708  
Rhinovirus/Enterovirus Score: -0.86277  
Coronavirus HKU1 Score: 0.00000  
Parainfluenza 3 Score: 0.00000  
Chlamydomphila pneumoniae Score: 0.00000  
Adenovirus Score: 0.00000  
Parainfluenza 4 Score: 0.00000  
Coronavirus229E Score: 0.00000  
CoronavirusOC43 Score: -0.58288  
Inf A H1N1 2009 Score: -0.80941  
Bordetella pertussis Score: 0.00000  
Metapneumovirus Score: -0.03437  
Parainfluenza 2 Score: 0.00000  
Influenza B, rapid test Score: -0.22301  
Influenza A, rapid test Score: -0.09826

# RESULTS – Mix Dataset

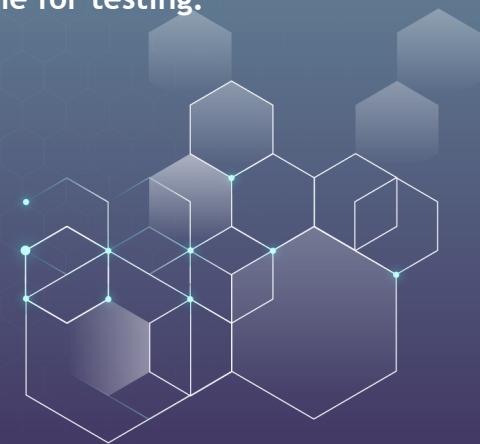




03

# Classification Model 2

I looked for other models to improve performance, I found two modifications of the logistic regression, Firths Logistic Regression and Log  $f(m, m)$ . I used the first one for testing.



# RESULTS – Numerical Dataset

91% 😊 26% ☹️

Precision

85%  
Accuracy

True  
Negatives

True  
Positives

Predicted  
Negatives

Predicted  
Positives

148

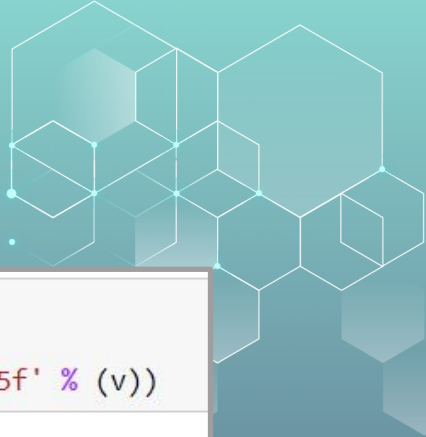
14

14

5



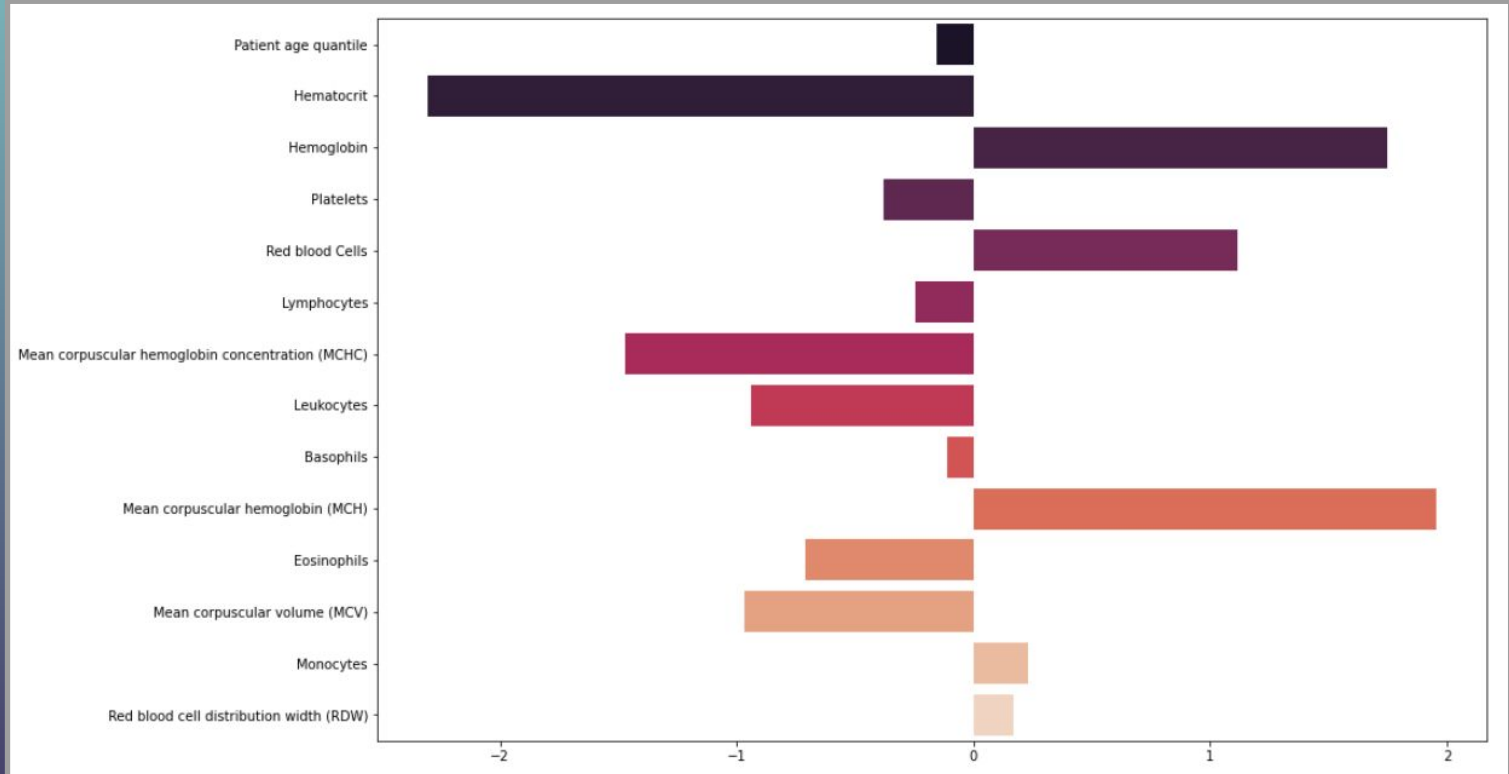
# RESULTS – Numerical Dataset



```
In [15]: # summarize feature importance
for i,v in enumerate(weights):
    print(str(list(Data_Numerical.columns.values)[i]) + ' Score: %.5f' % (v))
```


```
Patient age quantile Score: -0.15329
Hematocrit Score: -2.30512
Hemoglobin Score: 1.74979
Platelets Score: -0.38222
Red blood Cells Score: 1.11313
Lymphocytes Score: -0.24534
Mean corpuscular hemoglobin concentration (MCHC) Score: -1.46935
Leukocytes Score: -0.93720
Basophils Score: -0.10969
Mean corpuscular hemoglobin (MCH) Score: 1.95556
Eosinophils Score: -0.71253
Mean corpuscular volume (MCV) Score: -0.96635
Monocytes Score: 0.23034
Red blood cell distribution width (RDW) Score: 0.16693
```

# RESULTS – Numerical Dataset





## 04 Future

- Check other models that work best with small, unbalanced datasets
  - Look for more correlations with features and patients;
  - Try another approach, such as including non-zero values.
  - Try a decision tree model, likewise XGBoost.
- 



# References

<https://towardsdatascience.com/exploratory-analysis-python-kaggle-data-b0afb6ec1788>

<https://www.dataquest.io/blog/tutorial-add-column-pandas-dataframe-based-on-if-else-condition/>

<https://towardsdatascience.com/a-gentle-introduction-to-exploratory-data-analysis-f11d843b8184>

<https://github.com/dformoso/sklearn-classification/blob/master/Data%20Science%20Workbook%20-%20Census%20Income%20Dataset.ipynb>

<https://medium.com/datadriveninvestor/exploratory-data-analysis-in-python-a3b53fadb421>

<https://towardsdatascience.com/weighted-logistic-regression-for-imbalanced-dataset-9a5cd88e68b>

<https://machinelearningmastery.com/calculate-feature-importance-with-python/>

<https://medium.com/datadriveninvestor/firhs-logistic-regression-classification-with-datasets-that-are-small-imbalanced-or-separated-49d7782a13f1>

<https://medium.com/@remycanario17/log-f-m-m-logit-the-best-classification-algorithm-for-small-datasets-fc92f495bc58>

<https://www.kaggle.com/rafi3a/dealing-with-very-small-dataset-5>