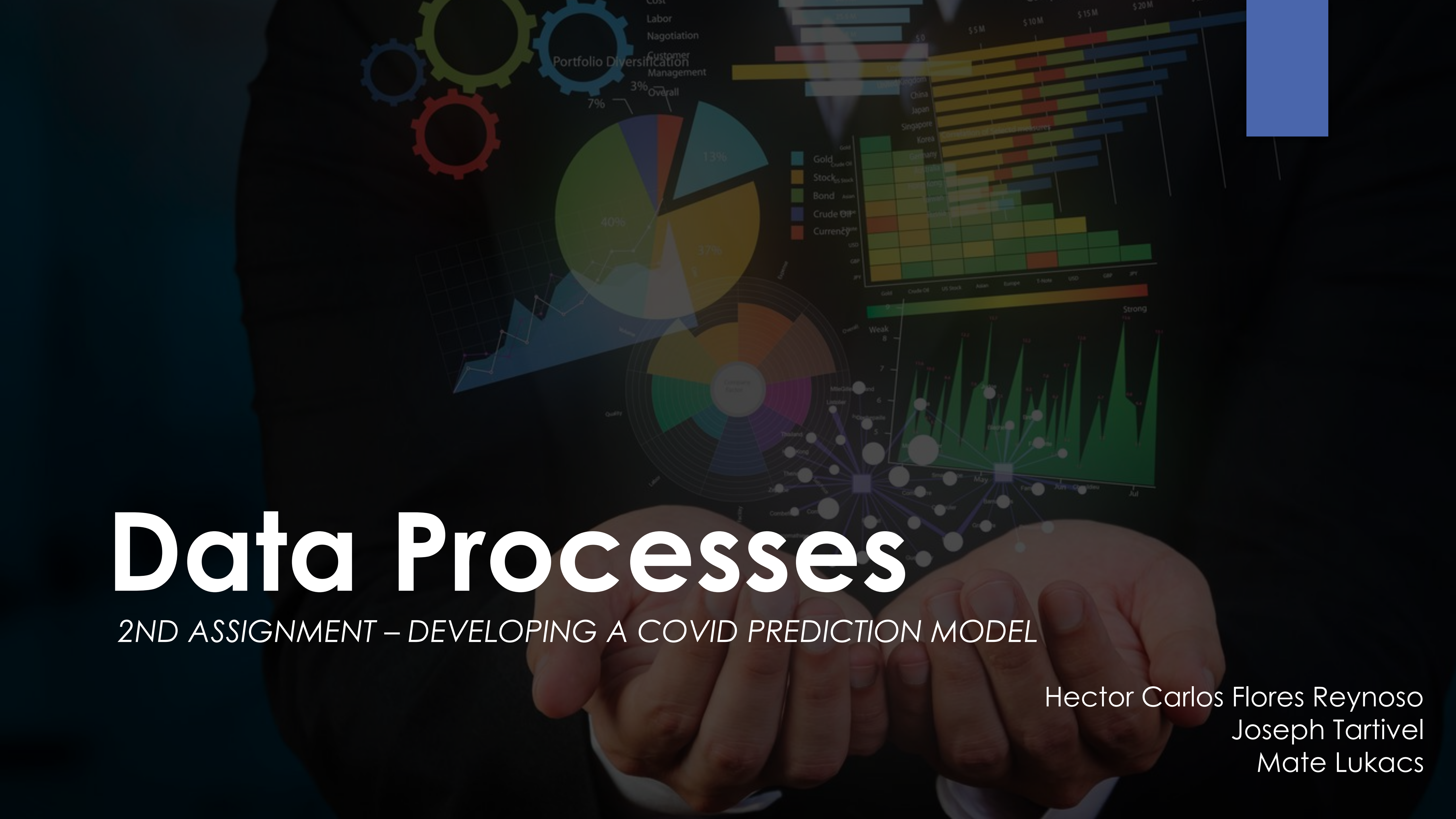


Data Processes

2ND ASSIGNMENT – DEVELOPING A COVID PREDICTION MODEL

Hector Carlos Flores Reynoso
Joseph Tartivel
Mate Lukacs

Introduction

- **Hospital Pressure:** Rapid, accurate COVID-19 patient identification is critical
- **Testing Issues:** Traditional methods are slow, straining hospital resources
- **AI Solution:** Machine learning predicts COVID-19 using patient dataurces
- **Efficiency Boost:** Optimizes bed use, testing, and patient isolation
- **Better Care:** Enhances screening, reduces strain, improves patient outcomes

Business Goal and KPI

- **Business/Mining Goal:**
  - Reduce the total screening time to be hospitalized after taking a COVID-19 test (Time to receive your results) against RT-PCR tests.
  - Develop a predictive model to identify patients in need of hospitalization based on admission characteristics and PCR results.

- **KPI:**
  - **Average hospitalization** waiting time after a PCR-test
  - **Accuracy in Positive Case Detection:** Achieve at least 95% recall for identifying COVID-positive individuals to ensure minimal missed diagnoses in high-risk area

Solution and process

EDA Analysis

Hospital 1 Analysis

- 54 columns of data with different data types
  - Integers
  - Floats
  - Strings (object)
  - Datetime
- Continuous, Nominal and Ordinal values
- Missing 1720 total values

Hospital 2 Analysis

- 54 columns of data with different data types
  - Integers
  - Floats
  - Strings (object)
  - Datetime
- Continuous, Nominal and Ordinal values
- Missing 1333 total values
- 1 column (age) with unexpected value

Hospital 1 vs Hospital 2

- 50 similar column names with different data types (**floats vs ints**)

- 4 different columns names in Hospital 1

  - **basvurutarihi (Admission Date)** # 2021-03-01
  - **gender_k=female_e=male** # K
  - **nationality** # Mexico
  - **patient_id.1** # 11850006

- 4 different columns names in Hospital 2

  - **admission_date** # 2021-03-01 00:00:00
  - **admission_id** # 45.0
  - **country_of_residence** # T.C.
  - **sex** # K

Data Preprocessing

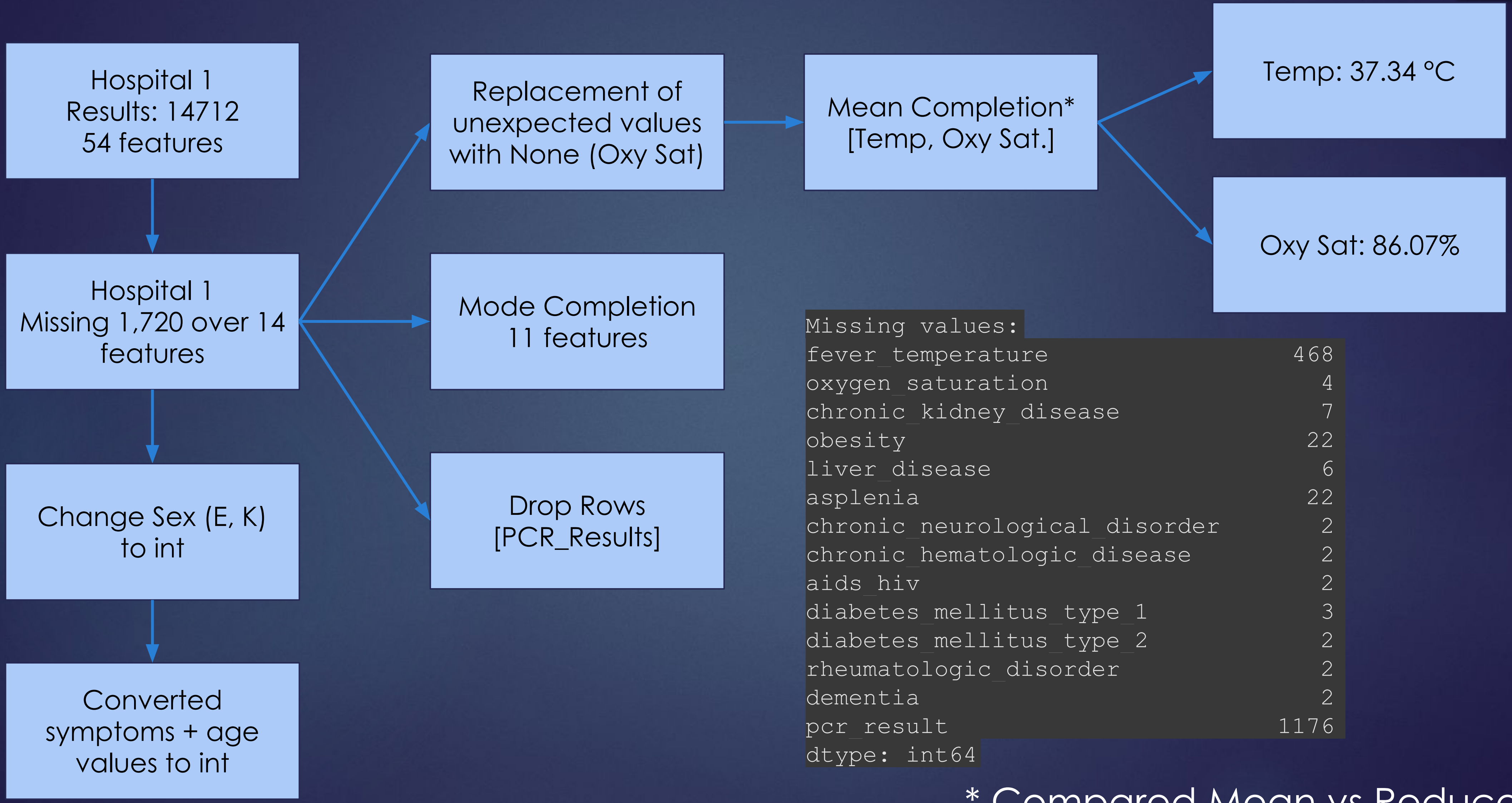
Column Mapping

Hospital 1		Hospital 2
`basvurutarihi`	→	`admission_date`
`patient_id.1`	→	`admission_id`
`gender_k=female_e=male`	→	`sex`
`nationality`	←	`country_of_residence`

Drop Values:

- Drop values with empty nationalities # Population Density plays a big role

Data Preprocessing: Hospital 1



Hospital 1
Results: 14712
54 features

Hospital 1
Missing 1,720 over 14 features

Change Sex (E, K) to int

Converted symptoms + age values to int

Replacement of unexpected values with None (Oxy Sat)

Mode Completion
11 features

Drop Rows
[PCR_Results]

Mean Completion*
[Temp, Oxy Sat.]

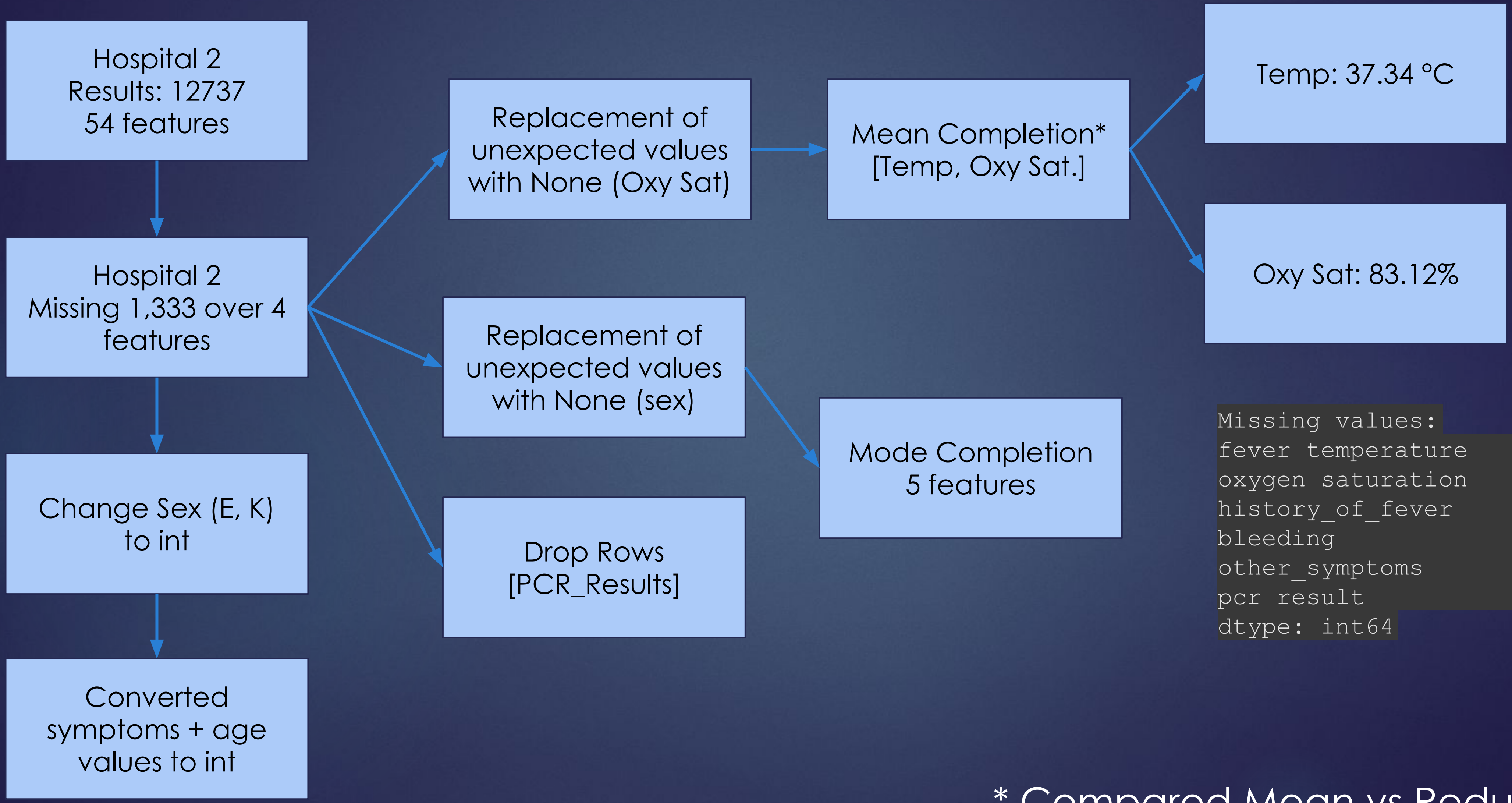
Temp: 37.34 °C

Oxy Sat: 86.07%

```
Missing values:
fever_temperature                     468
oxygen_saturation                       4
chronic_kidney_disease                  7
obesity                                22
liver_disease                           6
asplenia                               22
chronic_neurological_disorder           2
chronic_hematologic_disease             2
aids_hiv                                2
diabetes_mellitus_type_1                3
diabetes_mellitus_type_2                2
rheumatologic_disorder                  2
dementia                                2
pcr_result                           1176
dtype: int64
```

* Compared Mean vs Reduced Mean

Data Preprocessing: Hospital 2



Hospital 2
Results: 12737
54 features

Hospital 2
Missing 1,333 over 4 features

Change Sex (E, K) to int

Converted symptoms + age values to int

Replacement of unexpected values with None (Oxy Sat)

Replacement of unexpected values with None (sex)

Drop Rows [PCR_Results]

Mean Completion* [Temp, Oxy Sat.]

Mode Completion
5 features

Temp: 37.34 °C

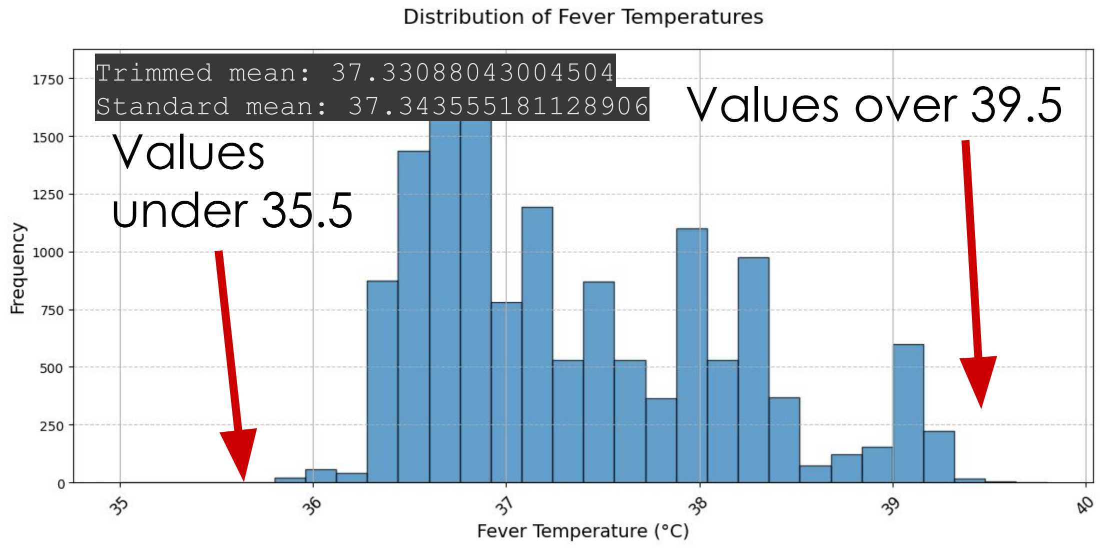
Oxy Sat: 83.12%

```
Missing values:
fever_temperature     1219
oxygen_saturation        4
history_of_fever         5
bleeding                36
other_symptoms          36
pcr_result              33
dtype: int64
```

* Compared Mean vs Reduced Mean

Reduced Mean vs Computed Mean

- Continuous values such a **Temperature** and **Oxygen Saturation** need to be computed using mean. However extreme outliers can cause an invalid statistic. Should they be considered?

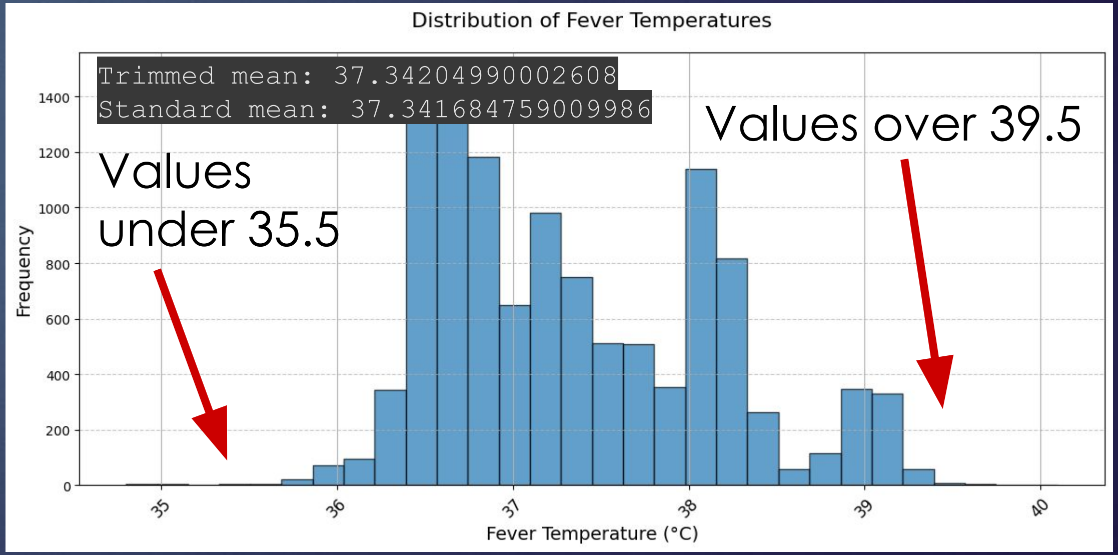


```
Trimmed mean: 37.33088043004504
Standard mean: 37.343555181128906
```

Values under 35.5

Values over 39.5

Hospital 1



```
Trimmed mean: 37.34204990002608
Standard mean: 37.341684759009986
```

Values under 35.5

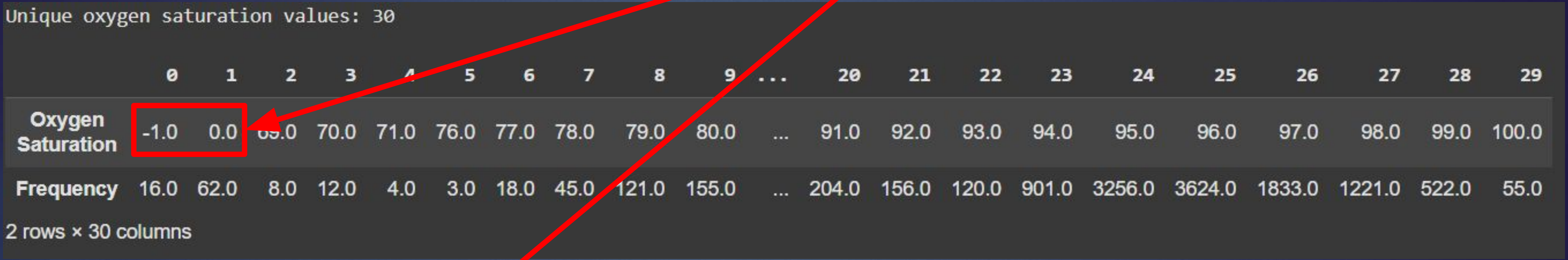
Values over 39.5

Hospital 2

Reduced Mean vs Computed Mean

-1%, 0% of saturations means your dead

Hospital 1

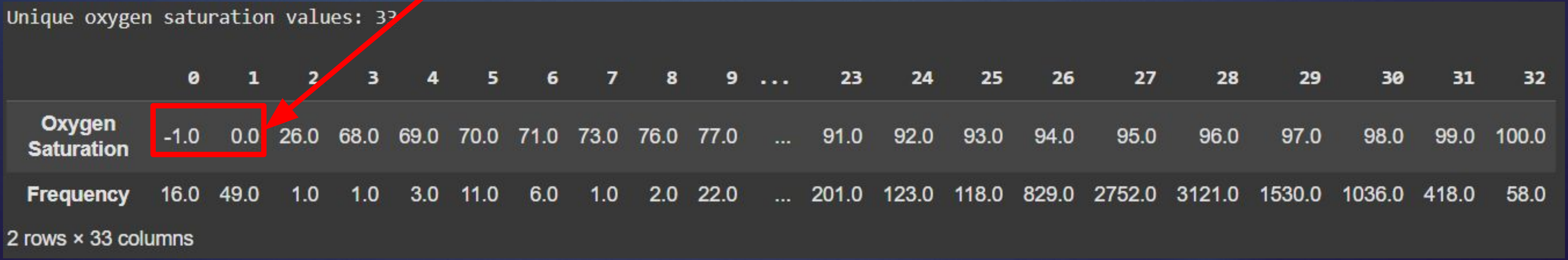


```
Unique oxygen saturation values: 30
```

	0	1	2	3	4	5	6	7	8	9	...	20	21	22	23	24	25	26	27	28	29
Oxygen Saturation	-1.0	0.0	69.0	70.0	71.0	76.0	77.0	78.0	79.0	80.0	...	91.0	92.0	93.0	94.0	95.0	96.0	97.0	98.0	99.0	100.0
Frequency	16.0	62.0	8.0	12.0	4.0	3.0	18.0	45.0	121.0	155.0	...	204.0	156.0	120.0	901.0	3256.0	3624.0	1833.0	1221.0	522.0	55.0

2 rows × 30 columns

Hospital 2

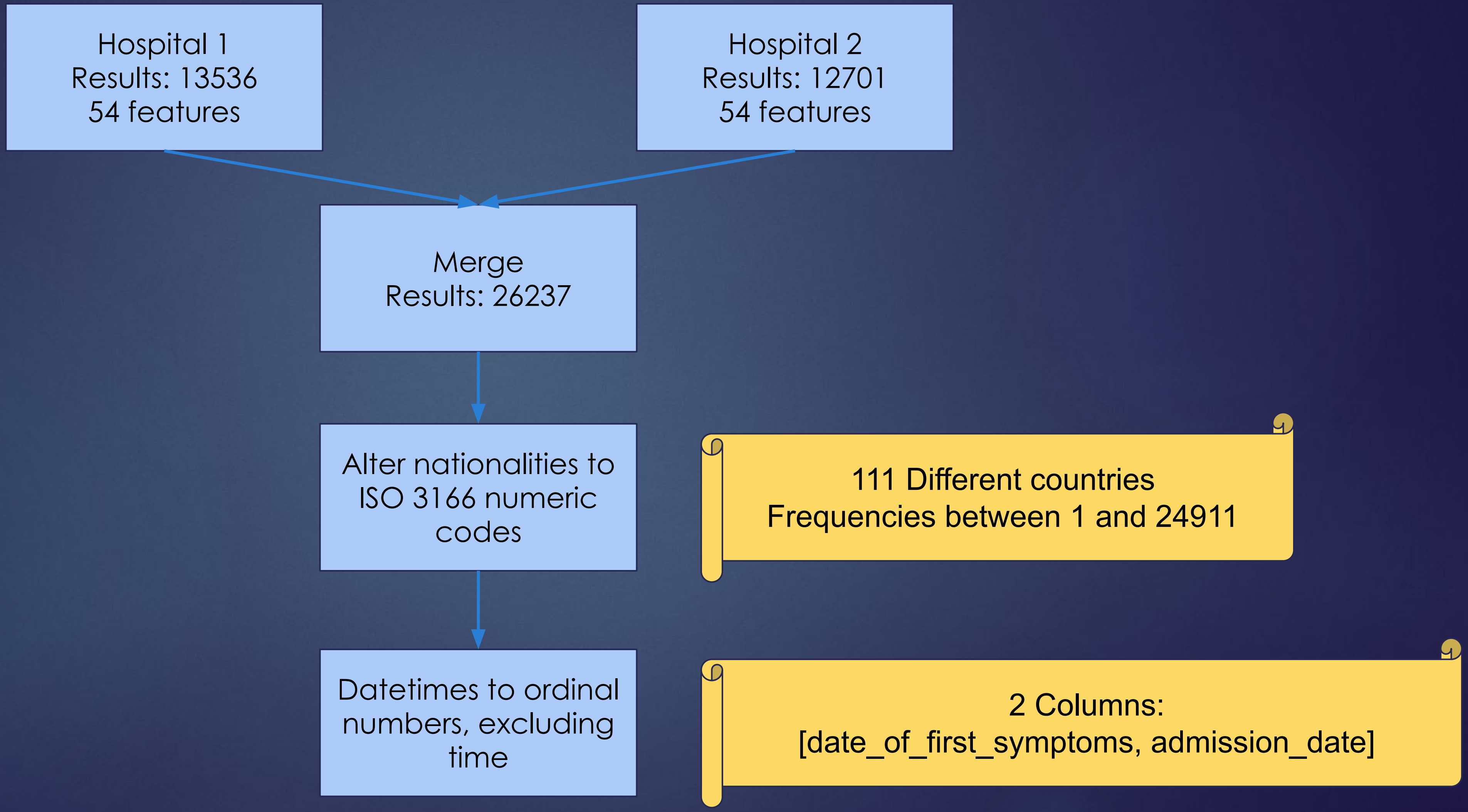


```
Unique oxygen saturation values: 33
```

	0	1	2	3	4	5	6	7	8	9	...	23	24	25	26	27	28	29	30	31	32
Oxygen Saturation	-1.0	0.0	26.0	68.0	69.0	70.0	71.0	73.0	76.0	77.0	...	91.0	92.0	93.0	94.0	95.0	96.0	97.0	98.0	99.0	100.0
Frequency	16.0	49.0	1.0	1.0	3.0	11.0	6.0	1.0	2.0	22.0	...	201.0	123.0	118.0	829.0	2752.0	3121.0	1530.0	1036.0	418.0	58.0

2 rows × 33 columns

Data Preprocessing: Merge



Hospital 1
Results: 13536
54 features

Hospital 2
Results: 12701
54 features

Merge
Results: 26237

Alter nationalities to ISO 3166 numeric codes

111 Different countries
Frequencies between 1 and 24911

Datetimes to ordinal numbers, excluding time

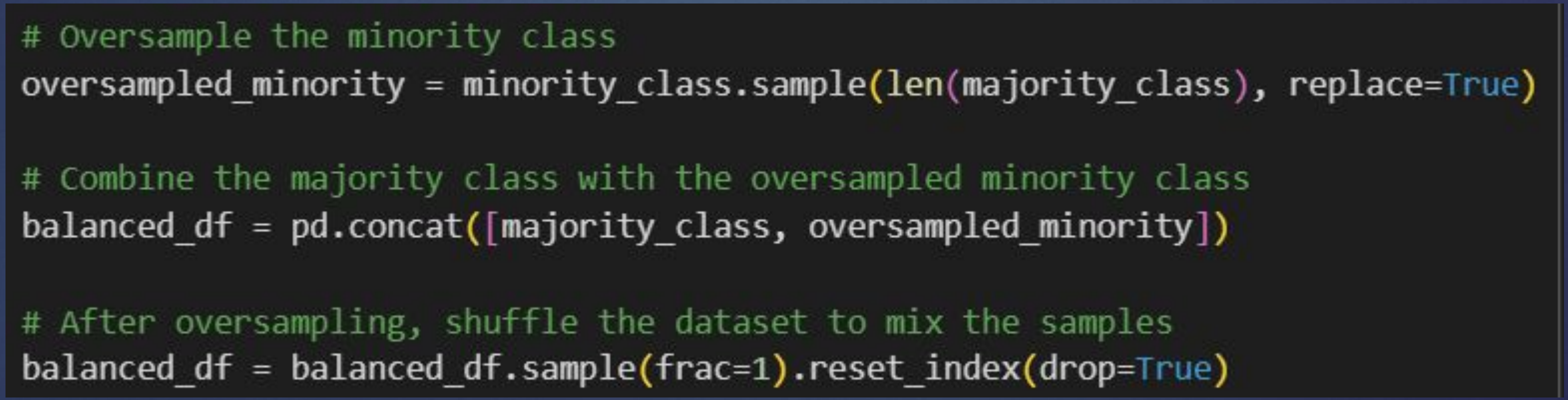
2 Columns:
[date_of_first_symptoms, admission_date]

Data Processing: Analysis

Total number of rows in dataset: **26237**

- Number of Negative PCR-Results: **4,027**
- Number of Positive PCR-Results: **22,210**

- Balanced using **oversampling**



```
# Oversample the minority class
oversampled_minority = minority_class.sample(len(majority_class), replace=True)

# Combine the majority class with the oversampled minority class
balanced_df = pd.concat([majority_class, oversampled_minority])

# After oversampling, shuffle the dataset to mix the samples
balanced_df = balanced_df.sample(frac=1).reset_index(drop=True)
```

Modeling

- Removed obvious columns that are related to COVID-19
  - pacient_id
  - admission date
  - admission id
  - pcr result # That the y-value

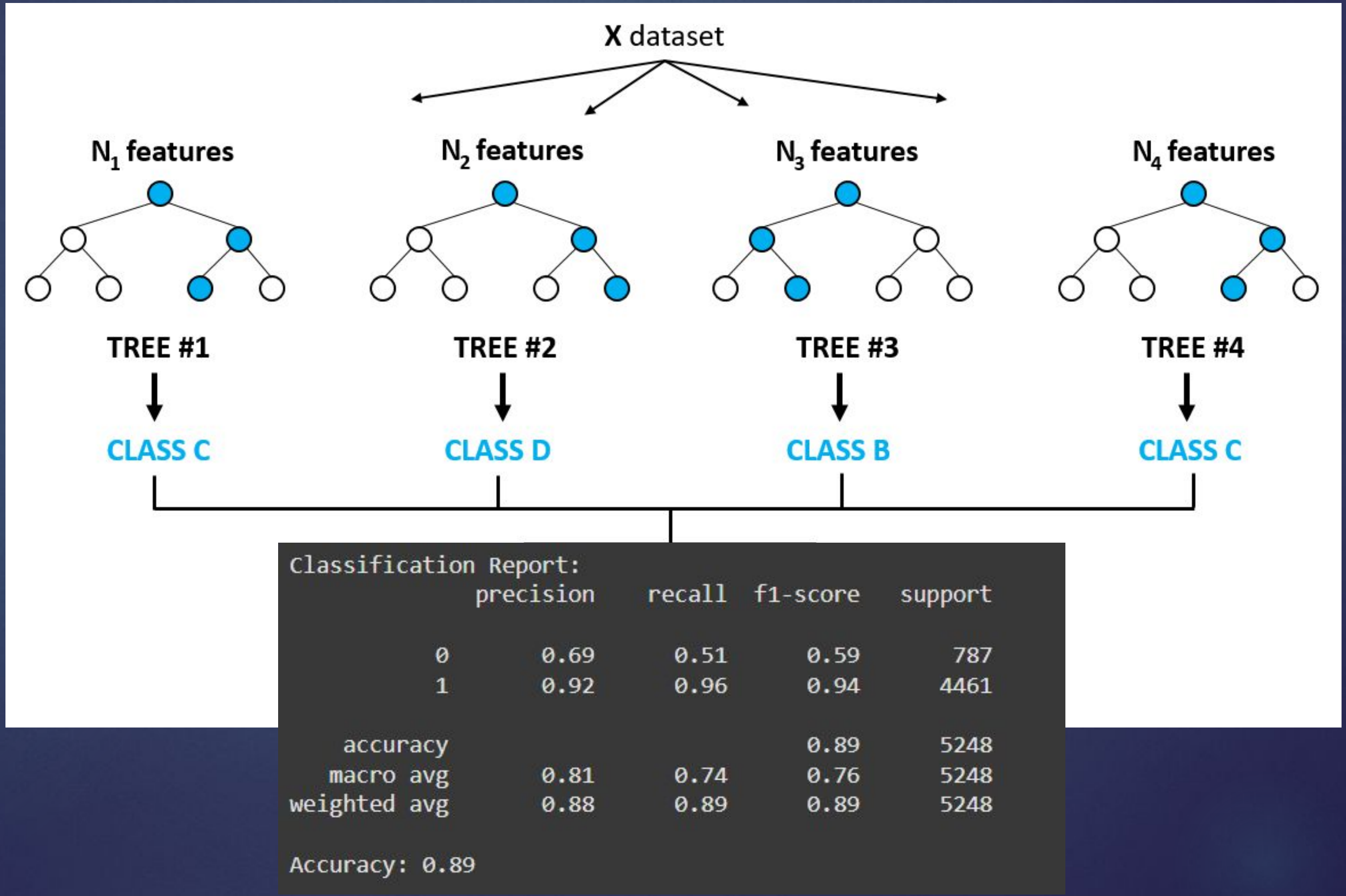
- Divided **80% train** and **20% test**

- **Pearson Correlation** analysis should that we have a range of values, meaning that many values don't have linear relationships. So ***linear models are discarded.***
  - **Min**: 3.860067504733165e-05
  - **Max:** 0.1625352289290226

Modeling

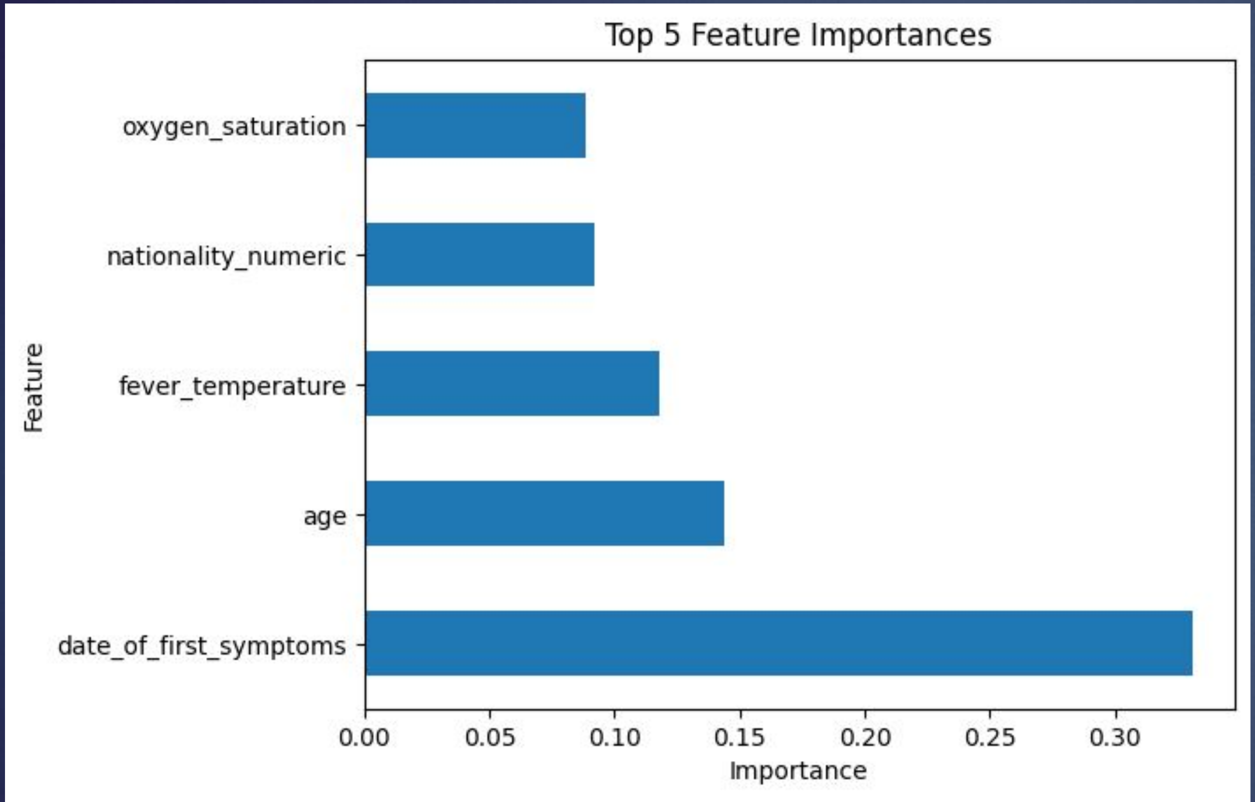
Trained with RandomForrestClassifier with 42 **random_states** and 100 **n_estimators,** to maximize the number of best alternatives

80% - Train
20% - Test



```
Classification Report:
              precision    recall  f1-score   support

           0       0.69      0.51      0.59       787
           1       0.92      0.96      0.94      4461

    accuracy                           0.89      5248
   macro avg       0.81      0.74      0.76      5248
weighted avg       0.88      0.89      0.89      5248

Accuracy: 0.89
```

Findings



- Feature importance (nominal, ordinal and continuous) play an important role.
- It was good to drop N/A nationalities since it plays an important role
- As expected age and Oxy. Satu are of great importance
- Values such as date_of_symptoms can be heavily related since there can be more data in certain periods.

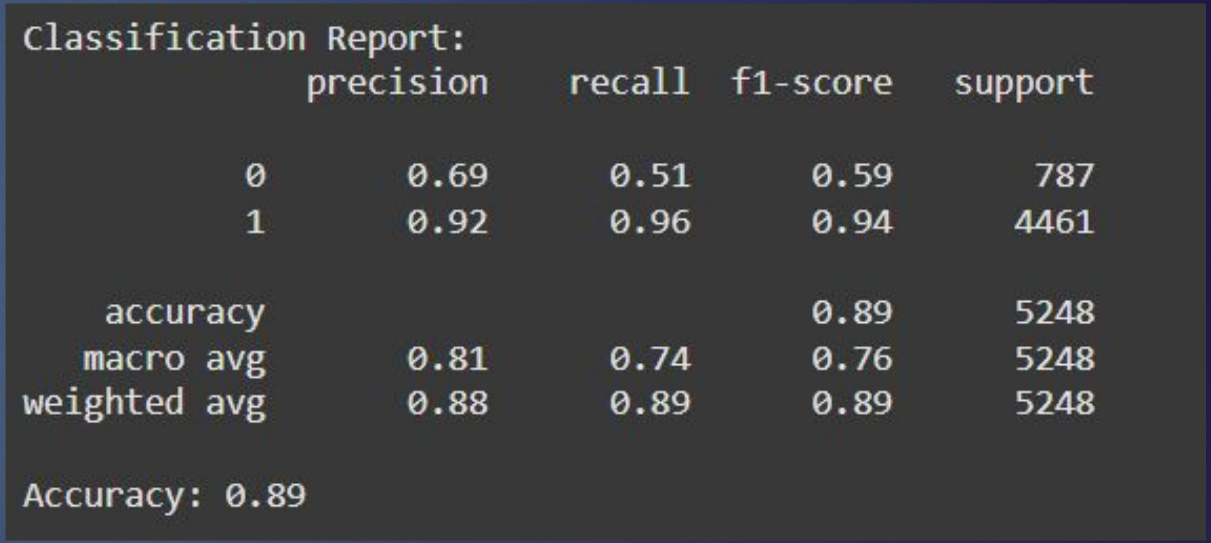
Findings

Weighted Average F1-Score: 89%

- Accounting for class imbalance (**splitting data**), the model has a robust real-world utility..

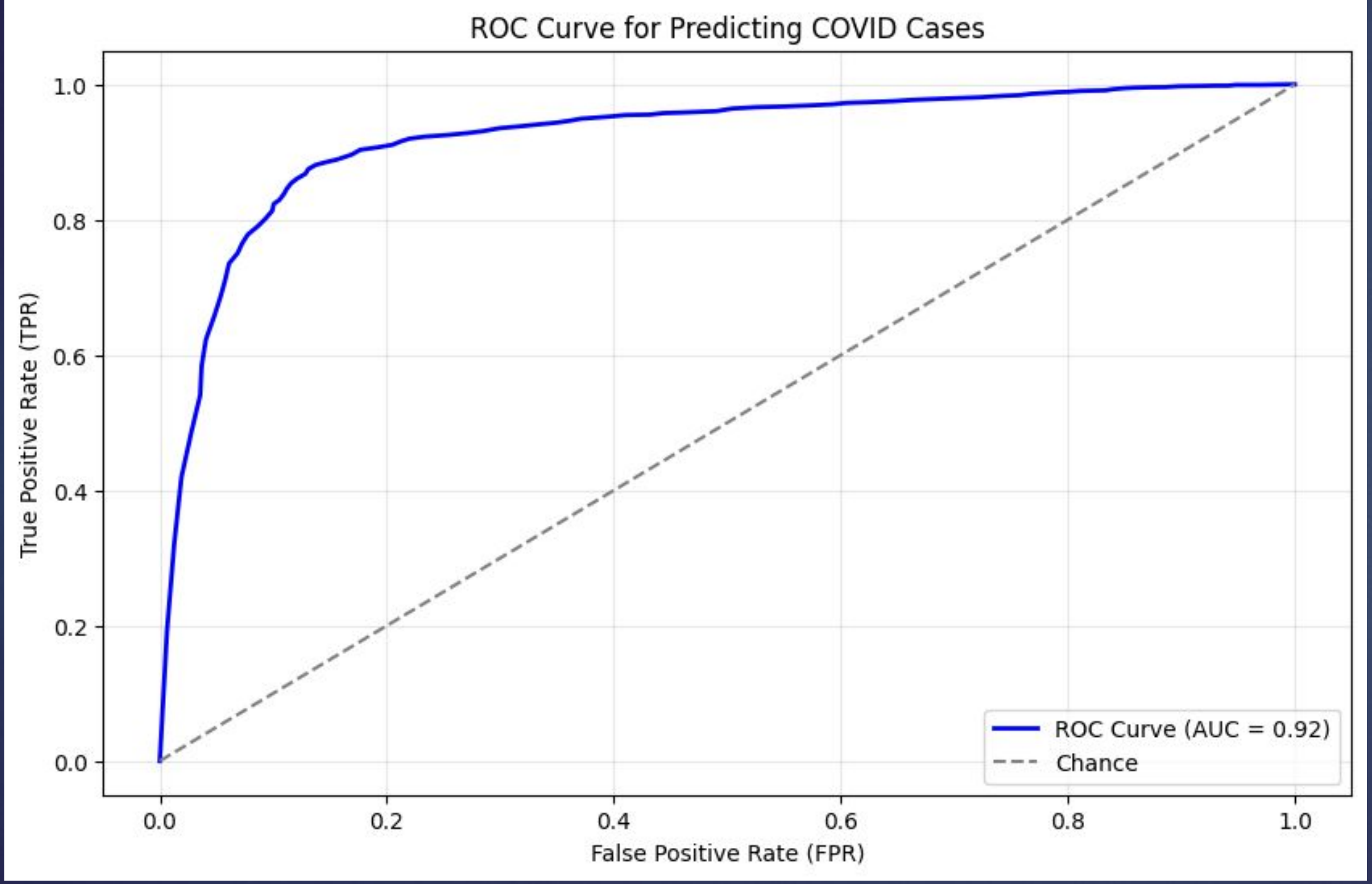
Model Accuracy: 89%

- The model is highly effective at identifying PCR-positive cases, with minimal false negatives.
- Performance for PCR-negative cases is moderate, with room for improvement in reducing false positives and negatives.



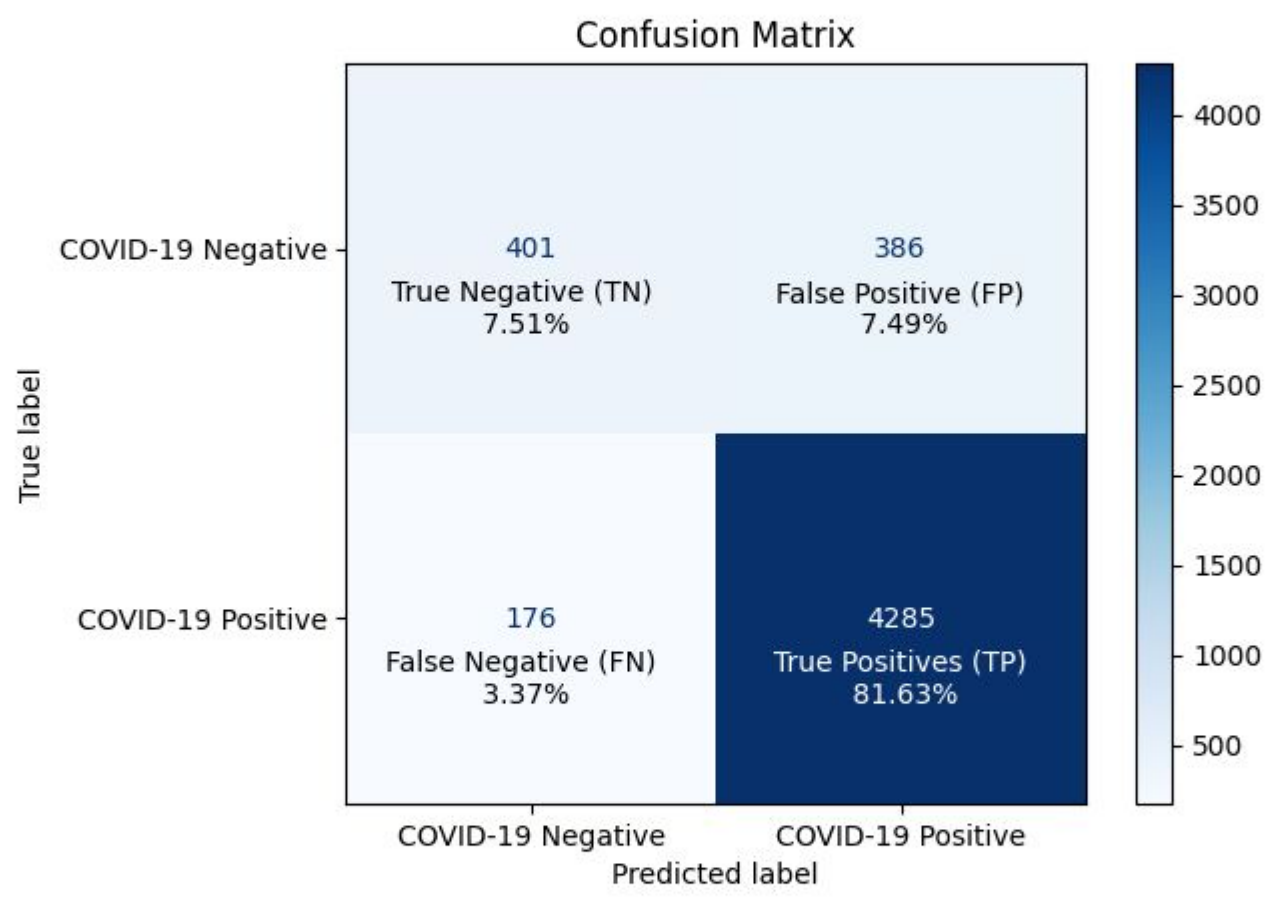
```
Classification Report:
              precision    recall  f1-score   support

           0       0.69      0.51      0.59       787
           1       0.92      0.96      0.94      4461

    accuracy                           0.89      5248
   macro avg       0.81      0.74      0.76      5248
weighted avg       0.88      0.89      0.89      5248

Accuracy: 0.89
```

Findings - ROC Curve



- 92% ROC Curve suggest indicates a very strong performance for a classification model in distinguishing classes on unseen data.
- This is backed by the 89% F1 Weighted score, even when not taking imbalance into consideration

Findings



Confusion Matrix

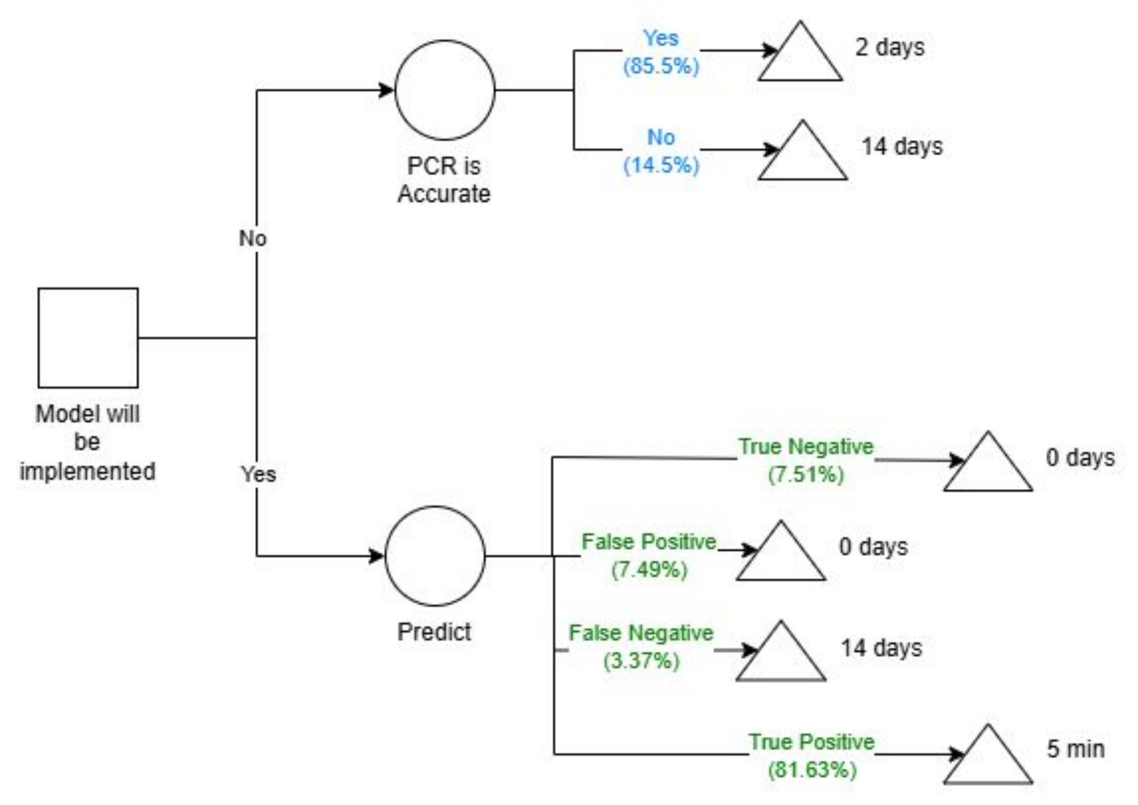
True label \ Predicted label	COVID-19 Negative	COVID-19 Positive
COVID-19 Negative	401 True Negative (TN) 7.51%	386 False Positive (FP) 7.49%
COVID-19 Positive	176 False Negative (FN) 3.37%	4285 True Positives (TP) 81.63%

Using a confusion matrix we can calculate the true impact to the customer.

Using the **20% test data**, we can simulate a scenario where we can predict how many days will it take for a single patient to be hospitalized

Findings

Assuming that a standard 1st Generation **RT-PCR** test has a accuracy of 85% with a mean waiting time between 1-2 days we can calculate the impact if the model is implemented*

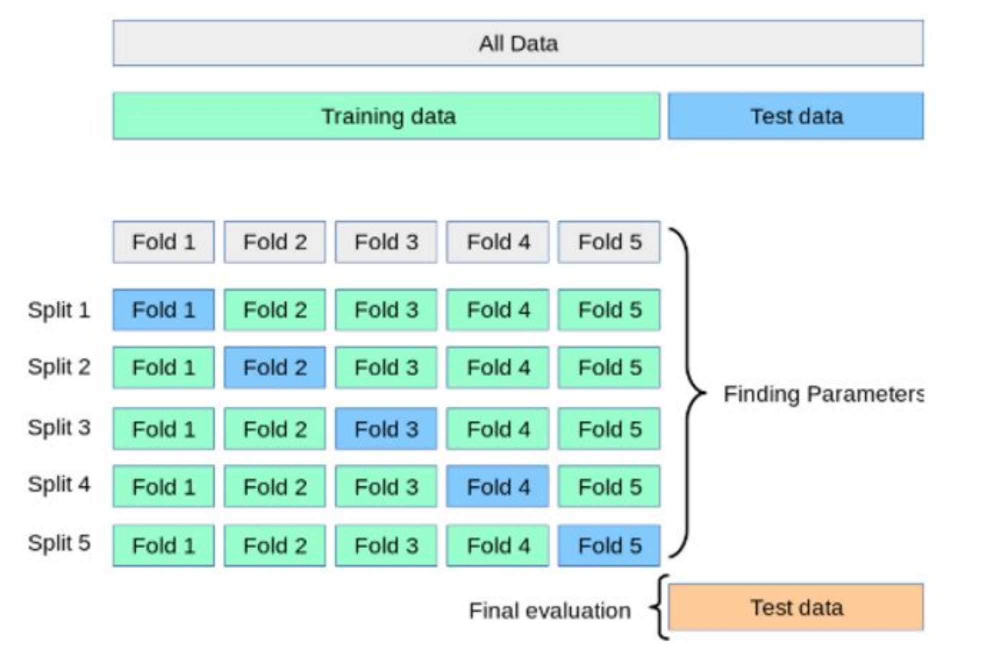


- The average hospitalization rate using a **prediction model** is **~11 hours** accounting for all possible scenarios
- The average hospitalization rate using the **traditional PCR-Test** approach is **~1.5 days** accounting for all possible scenarios

* Oliveira, M.C., Scharan, K.O., Thomés, B.I. *et al.* Diagnostic accuracy of a set of clinical and radiological criteria for screening of COVID-19 using RT-PCR as the reference standard. *BMC Pulm Med* 23, 81 (2023). https://doi.org/10.1186/s12890-023-02369-9

Future training notes

- For future implementation we need to focus on improving PCR_Negative detection to enhance recall and balance across both classes
- When breaking the datasets, we need to use techniques such as **K-Types** to address proper shuffling to avoid improper balance.


All Data
Training data
Test data
Fold 1 Fold 2 Fold 3 Fold 4 Fold 5
Split 1 Fold 1 Fold 2 Fold 3 Fold 4 Fold 5
Split 2 Fold 1 Fold 2 Fold 3 Fold 4 Fold 5
Split 3 Fold 1 Fold 2 Fold 3 Fold 4 Fold 5
Split 4 Fold 1 Fold 2 Fold 3 Fold 4 Fold 5
Split 5 Fold 1 Fold 2 Fold 3 Fold 4 Fold 5
Finding Parameters
Final evaluation
Test data

Conclusion

- Implementing the model reduces the average hospitalization time from **~1.5 days** to **~11 hours** or a **87.31% decrease.**
- Factors such as **age, oxygen saturation and fever play** a great importance making crucial to get those values right
- The business objective was achieved, since we got a 87.31% reduction which is great overall, while increasing the accuracy against traditional methods.
- KPI's such as recall in positive test where successfully achieved with a 1% difference (95% to 96%)
- Even though we got a 69% precision when predicting false results, the model is still implementable, which is backed by the ROC-Curve with a 92%. This is a good indicator to test the model in real life

Summary

- We started by exploring the data, understanding the values
- Transforming it, cleaning and standardizing the values
- Data was imbalance, so oversampling was applied to level the bias
- Given that they where low correlation values, we decided to implement alternative non-linear models such as Random Forest to train the data
- The model yield a 89% accuracy against a 85.5% accuracy in RT-PCR testing.
- Further breakdown can be done by partitioning data more appropriately using K-Fold techniques
- We computed the ROC-Curve against the test data with a 92%.