Machine Learning Model Development based on Brazil's Covid-19 Data Set

Dr. Kamel Alikhan Siddiqui
Associate Prof, LIET, Lords Institute of Engineering and Technology(Autonomous), Hyderabad TS, India
Dr. C. V Narasimhulu
Principal LIET, Lords Institute of Engineering and Technology(Autonomous), Hyderabad TS, India
Dr. K Nagi Reddy
HOD IT, LIET, Lords Institute of Engineering and Technology(Autonomous), Hyderabad TS, India
Mrs. Rayeez Fatima
Asst. Prof (SWCET), Lords Institute of Engineering and Technology(Autonomous), Hyderabad TS, India

Abstract: Brazil is one of the countries most affected by the COVID-19 pandemic, with more than 16 million confirmedcases and 454,429 confirmed deaths by May 26, 2021 (according to the Johns Hopkins Coronavirus Resource Center). Brazil was and still is one of the countries most impacted by the first wave of Covid-19 which first recorded case on 26th February 2020 and reached community transmission from 20th March 2019 onwards to dates, that caught Brazil unprepared and unable to response due to the strain on hospital capacity such as the intense and lengthy request for ICU (incentive care unit) beds, professionals, personal protection equipment and healthcare resources. A data science team at Sirio Libanês, a top-tier hospital in Brazil, decided to use ML to help reduce the strain on hospital's ICU beds where the objective is to develop a ML model to predict if a patient of confirmed COVID-19 case would require admission to the ICU. With that objective in mind, the team has collected a decent amount of clinical data from patients i.e. the features of COVID patients, and the target (those been admitted to ICU). The paper describes the complete Model design using features of contemporary Machine Learning Models to interpret the Collected Data from the hospital to produce a decent report as a solution to the above mentioned hindrances at crucial times.

The dataset is released on Kaggle platform with full data description at the following URL (https://www.kaggle.com/S%C3%ADrio-Libanes/covid19) by the team seeking interesting solutions and findings from the public

Index terms: Python for Data Science and Machine Learning, Test and Train Data sets, Random Forest, Clustering model and Accuracy and Precision findings.

I. INTRODUCTION

In this paper, we are challenged to perform a full lifecycle ML model development according to the objective of the dataset, which includes the following elements [1], [2]:

1) Perform Exploratory Data Analysis (EDA), and establish hypotheses of predictive insights you expect to glean from the dataset.
2) Perform data preparation for ML informed by the EDA findings.
3) Develop ML model according to hypotheses of predictive insights you gleaned from the dataset. You are required to evaluate at least 3 ML algorithms and assess associated issues i.e. hyper parameters tuning, performance metrics, model complexity (under fitting/over fitting) etc. Finally, provide a recommendation of the best algorithm for your ML model.
4) Documentation:
   i) Jupyter-Notebook include all coding and technical report i.e. explanations, justifications, reasoning etc. for every finding, strategically decision, action, and choice made. Jupyter-notebook provide a comprehensive documentation capability by using

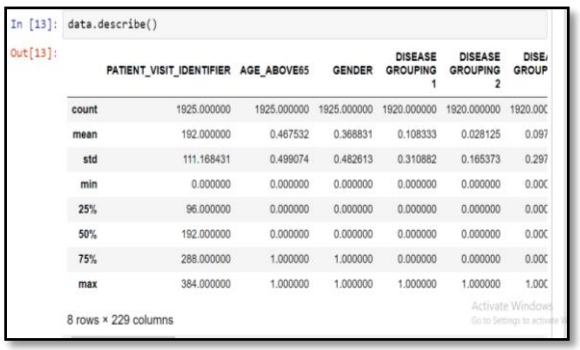
the Markdown (https://www.datacamp.com/community/tutorials/markdown-in-jupyter-notebook).

ii) Executive Summary Report include an overview of the entire lifecycleof the ML model development, written with target audience in mind such as high-level stakeholders, decision makers, directors.

II. EDA: Exploratory Data Analysis:

A ML model always begins with a Data preprocessing phase in where the loading of the important libraries are taken into consideration. Here numpy, pandas and matplotlib.pyplot forms the basis of the model design to interpret the data as per the convenience of the designer. The data_frame.head(5) describes the complete data types of the first 5 rows which is useful to find the missing values and the data types associated with individual variables. There are 1925 rows and 231 columns associated with the data fame provided by the Hospital Sirio Libanês, Brazil. The useful function for the same is data.shape from python [2].

In most digital data gathering stages the amount is huge and hence prior measurement of max, mean, standard, count with lower and upper quartiles are really important. Data.describe provides the important measuring schemes for the following as per the table mentioned below.



In [13]: `data.describe()`

Out[13]:

	PATIENT_VISIT_IDENTIFIER	AGE_ABOVE65	GENDER	DISEASE GROUPING 1	DISEASE GROUPING 2	DISE/ GROUP
count	1925.000000	1925.000000	1925.000000	1920.000000	1920.000000	1920.00(
mean	192.000000	0.467532	0.368831	0.108333	0.028125	0.097
std	111.168431	0.499074	0.482613	0.310882	0.165373	0.297
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.00(
25%	96.000000	0.000000	0.000000	0.000000	0.000000	0.00(
50%	192.000000	0.000000	0.000000	0.000000	0.000000	0.00(
75%	288.000000	1.000000	1.000000	0.000000	0.000000	0.00(
max	384.000000	1.000000	1.000000	1.000000	1.000000	1.00(

8 rows × 229 columns

Table: data.describe

The table candidly infers the complete recording of the admitted patients and of Sirio Libanês. From the above table the partition of the patients in terms of Age was derived under the range of 10 to 90+. The total number of patients was found to be 381

Here, ICU feature is the target label. Proportion of patients admitted/ not admitted to ICU [2]. The first and foremost Data processing is performed by knowing the Data types of each unique variable in the chart using data.dtypes which yield the list below:

```
In [12]: data.dtypes

Out[12]: PATIENT_VISIT_IDENTIFIER        int64
         AGE_ABOVE65                     int64
         AGE_PERCENTIL                  object
         GENDER                          int64
         DISEASE GROUPING 1            float64
                                          ...
         RESPIRATORY_RATE_DIFF_REL     float64
         TEMPERATURE_DIFF_REL          float64
         OXYGEN_SATURATION_DIFF_REL    float64
         WINDOW                         object
         ICU                             int64
         Length: 231, dtype: object
```

Fig: Data types of each variable

III. ICU Analysis:

The function ICU_prop_main=data['ICU'].value_counts().reset_index()explore the new data of ICU patients into account of admitted patients.

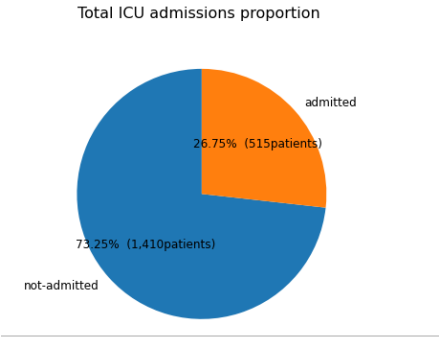


Fig: ICU Proportion

Here total 73.25% of not-admitted and 26.75% admitted ICU patients are there in the current dataset.

The identifier useful for calculating and finding the frequency of the patients above 65 who has visited to the Hospital is very useful in finding the next relations and assumptions towards the services [3]. Here the Patients who has visited more than 12 times and lesser are being shown for the perusal.

prop_65=data.groupby('WINDOW')['PATIENT_VISIT_IDENTIFIER'].count().reset_index() provides the dataset for providing the information needed to plot a pie chart as shown.

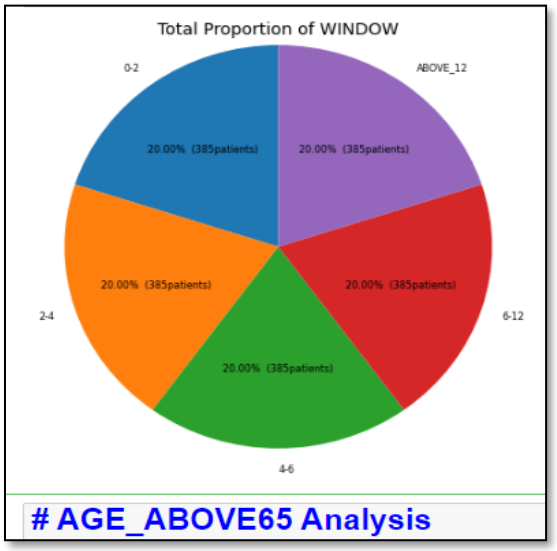


Fig: Pie Chart for Age_Above_65 frequency to the Hospital

Now the age of the patients that are below and above the age of 65 among the complete data frame can be taken out by prop_65 data set as follows, autopct= lambda p : '{:.2f}% ({:,.0f}patients)'.format(p,p * sum(prop_65['PATIENT_VISIT_IDENTIFIER'])/100)).

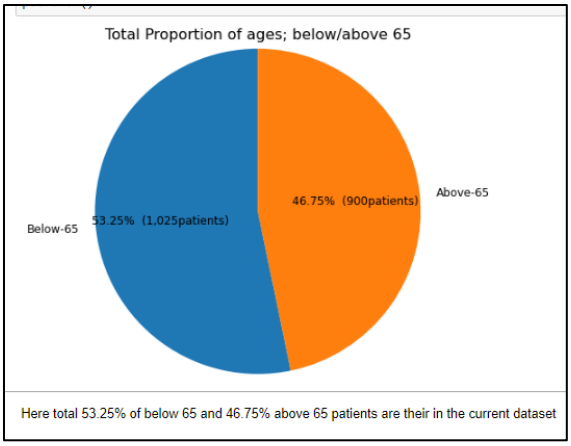


Fig: Patients above 65 years old and below.

In the similar fashion the proportion of the age below and above 65 can be represented using bar plots of Age Percentile using AGE_prop_percentil =AGE_prop_percentil.groupby('AGE_PERCENTIL')['PATIENT_VISIT_IDENTIFIER'].count().reset_index().

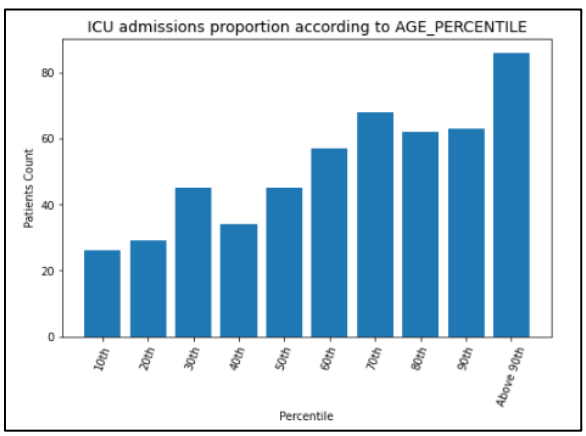
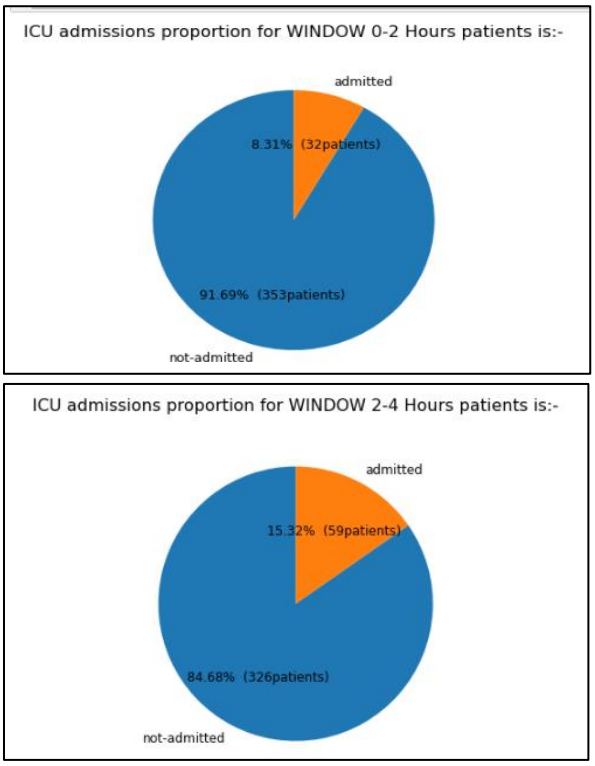


Fig: ICU Admission based onAGE_PERCENILE.

IV. ICU vs WINDOW Analysis:

One of the important consideration in the Hospitals is the Admission time analysis on the ICU occupancy and the availability of the space for the next patients. The majority of cases cannot be dealt with or tries to mislead the information by not portraying the accurate timing analysis of the time keeping of the patients flowing in and out in the ICU andEmergencies [2]. One of the unique method is to correctly keeping the information as soon the patients are brought in. one of the data science tool Fractional Delay (FD) filter has two principal properties is by performing Windowing Analysis by looping the time in terms of steps such as 0-2 hrs., 2-5hrs, 5 -9 hrs.Etc. as follows:

```
# proportion of patients admitted/ not admitted to ICU
for i in data['WINDOW'].unique():
   ICU_prop = data[data['WINDOW'] == i]
#group by ICU admissions
ICU_prop_main=ICU_prop.groupby('ICU')['PATIENT_VISIT_IDENTIFIER'].count().reset_index()
```



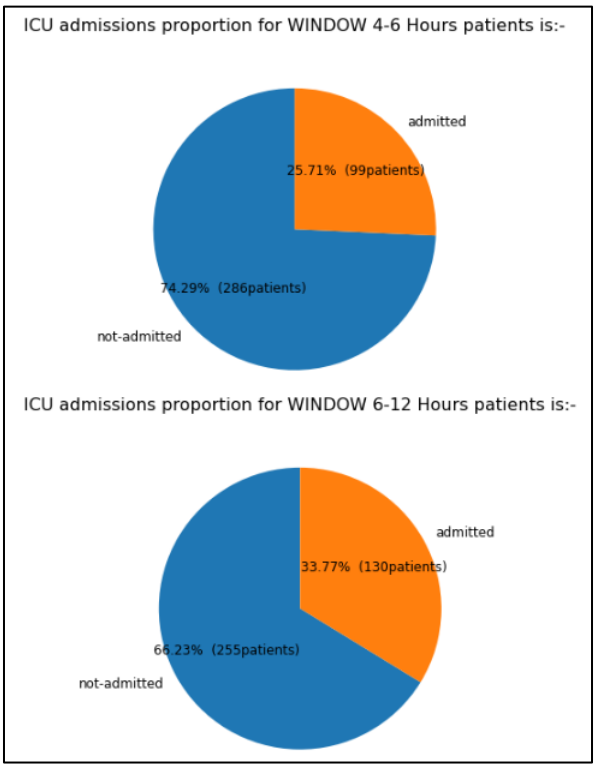


Fig: ICU Windowing hours

V. ICU vs AGE_ABOVE65patients analysis

The ICU occupancy for the above aged 65 patients can also be seen by keeping data set as shown under above age 65 in comparison with the ICU occupancy.

```
# Proportion of patients admitted/ not admitted to ICU
for i in data['AGE_ABOVE65'].unique():
    ICU_prop = data[data['AGE_ABOVE65'] == i]
prop_65=ICU_prop.groupby('ICU')['PATIENT_VISIT_IDENTIFIER'].count().reset_index()
```

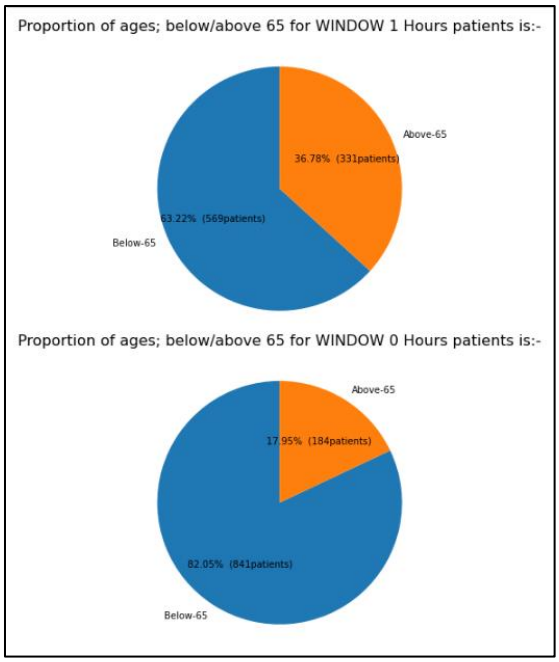


Fig: ICU vs Age_65

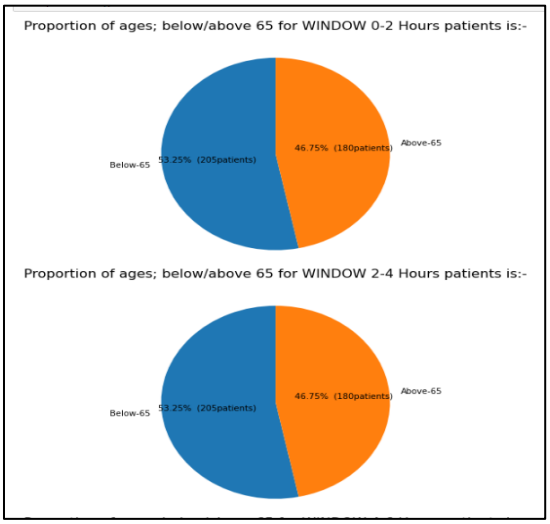
VI. WINDOWvs. AGE_ABOVE65

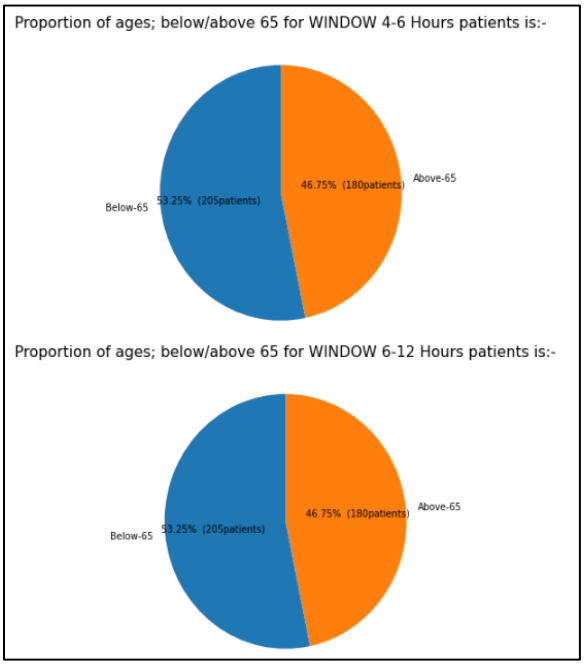
As we have seen the formulation of Window column is showing the object type and has given the range of 0 to 12 hours format for the rows under consideration. Now the patient'sunder 65 years of age needs to be kkept in this category for further analysis [3].

```
# proportion of patients admitted/ not admitted to ICU
for i in data['WINDOW'].unique():
   ICU_prop = data[data['WINDOW'] == i]

prop_65=ICU_prop.groupby('AGE_ABOVE65')['PATIENT_VISIT_IDENTIFIER'].count().reset_index
()
```

The illustration can be detailed as shown:





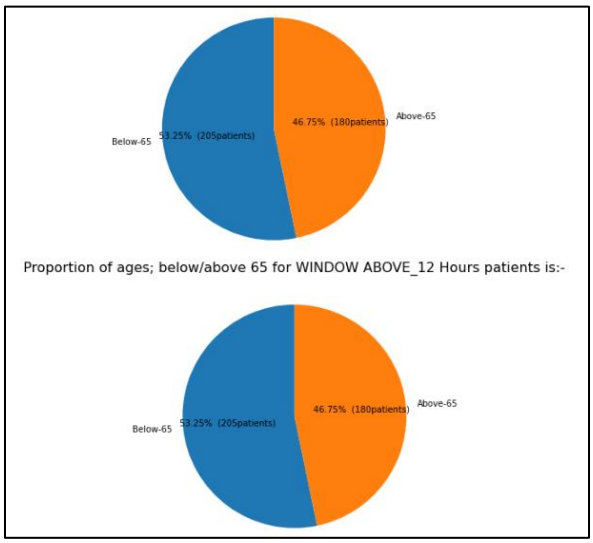


Fig: Window vs. Age_above65 with 2, 4,6,12 hours

The next column for indicating that the patients are eventually admitted or not can be traced by working on the following

create new column to indicate if a patient eventually went to ICU (ICU_SUM)

df_admitted=(data.groupby("PATIENT_VISIT_IDENTIFIER")["ICU"].sum()>0).reset_index()*1

df_admitted.columns=["PATIENT_VISIT_IDENTIFIER", "ICU_SUM"]

dataset_admitted = pd.merge(data,

df_admitted, on = "PATIENT_VISIT_IDENTIFIER")

V. FEATURE SELECTION:

The data set can be readjusted by knowing the missing value, we found that there are some values that needs to be filled and the refilling is performed by the method of ffilna().

The following assumptions have been made with the model:

#drop rows with ICU == 1 i.e. drop data when the target variable is present, as stipulated by dataset. We get

(1410, 232)

#keeping only window 0-2 data. We get

(353, 232)

#drop unnecessary columns.

#convert categorical columns.

#drop duplicated columns.

#columns were values are equal.

#check for empty or null cells. We get

(array([], dtype=int64), array([], dtype=int64))

For such feature selection the reduction in dataset variables can be performed by checking the correlations with target column.

corr_df=final_data.corrwith(final_data["ICU_SUM"]).We get

AGE_ABOVE65 0.292719

GENDER -0.115633

DISEASE GROUPING 1 0.072626

DISEASE GROUPING 2 0.088044

DISEASE GROUPING 3 0.123178
...
AGE_PERCENTIL_60th -0.017375
AGE_PERCENTIL_70th 0.025044
AGE_PERCENTIL_80th 0.138513
AGE_PERCENTIL_90th 0.148680
AGE_PERCENTIL_Above
90th0.198389
Length: 95, dtype: float64

The complete description of the corelation can be performed by using corr_df.describe()
And we get
count 94.000000
mean 0.026424
std 0.132713
min -0.193364
25% -0.041248
50% 0.021372
75% 0.074761
max 1.000000
dtype: float6.

VI. MODEL EVALUATION:

The selected model can be evaluated by creating two sets of Data frames one for Training-X and the other for Testing-Y.

```
X_data = selected_final_data.drop(['ICU_SUM'], axis = 1)
Y_data = selected_final_data[['ICU_SUM']].
```

The Classifier selected for the model is Random Forest Classifier from Skit Learn Library. The Training and Testing module thus so formed are fit under the module and are ready for Evaluation.

```
#train test split
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test =train_test_split(X_data, Y_data, test_size=0.30, random_state=1)

#FIT model
from sklearn.ensemble import RandomForestClassifier
model =RandomForestClassifier(max_depth=6, random_state=0)
model.fit(X_train,Y_train)
```

VII. CONCLUSION:

The selected features for the modeled design for the data set provided by the Hospital of Sirio Libanês Brazil which was able to deliver efficient amount of Time saving and Accuracy for the

Hospital departments to utilize the Hospital domain services efficiently. However the bigger effect of utilization was adopted by the ICU and other Emergency departmensts based on the accurate date in flow and outflow like discharge of the patients was analyzed [4]

```
#model prediction
y_pred = model.predict(X_test)
#accuracy, precision and recall
from sklearn import metrics
print("Accuracy:{:.6f}".format(metrics.accuracy_score(Y_test, y_pred)))
print("Precision:{:.6f}".format(metrics.precision_score(Y_test, y_pred)))
print("Recall:{:.6f}".format(metrics.recall_score(Y_test, y_pred)))
```

The Accuracy, Precession and Recall factors can be noted for the complete model deign, and hence found to be the following:

Accuracy: 0.622642
Precision: 0.612903
Recall: 0.404255

The Accuracy was noted to be the 62% while the Precession is 61% and Recall being 40%.

The model now being absolute completed can be regressed analyzed again by considering the null values to be taken into consideration and importing the standard or mean of the total values.

VIII.REFERENCES

[1] Patients, Authors Till D. Frank Content type: Regular Paper published: 04 April 2022.

[2] Adaptive k-center and diameter estimation in sliding windows
Paolo Pellizzoni Andrea Pietracaprina, Geppino PucciContent type: Regular Paper, Published: 02 April 2022

[3] An interaction-based method for detecting overlapping community structure in real-world networks Authors: Pawan Kumar Ravins Dohare, Content type: Regular Paper Published: 11 March 2022.

[4] Special Issue on AI and Data Science in COVID-19:A Commentary on this Pub Early-Career View on Data Science Challenges: Responsibility, Rigor, and Accessibility by Shuang Frost, Aleksandrina Goeva, William Seaton, Sara Stoudt, and Ana Trisovic Published on Apr 11, 2022.

★★★