CODING DOJO ACADEMY

الأكاديمية السعودية الرقمية
SAUDI DIGITAL ACADEMY

Spark Project

DaQuest Team

Today's Agenda

1 Introduction

2 Business Problem

3 Data Review

4 Data Preprocessing

5 Exploratory Data Analysis (EDA)

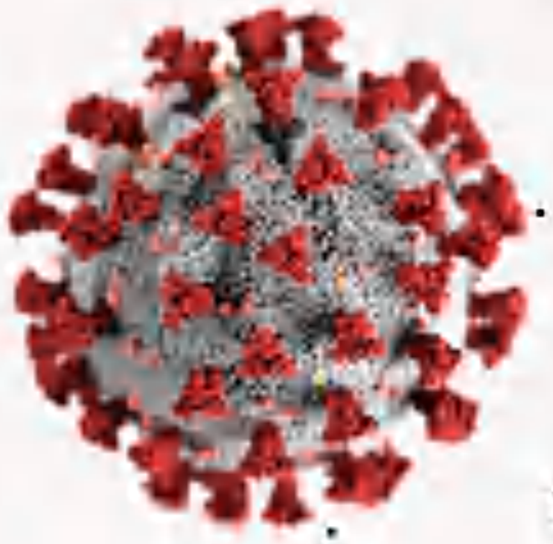
6 Machine Learning Models

Introduction

In our project, we practice building machine learning models with Spark to solve a specific business problem.

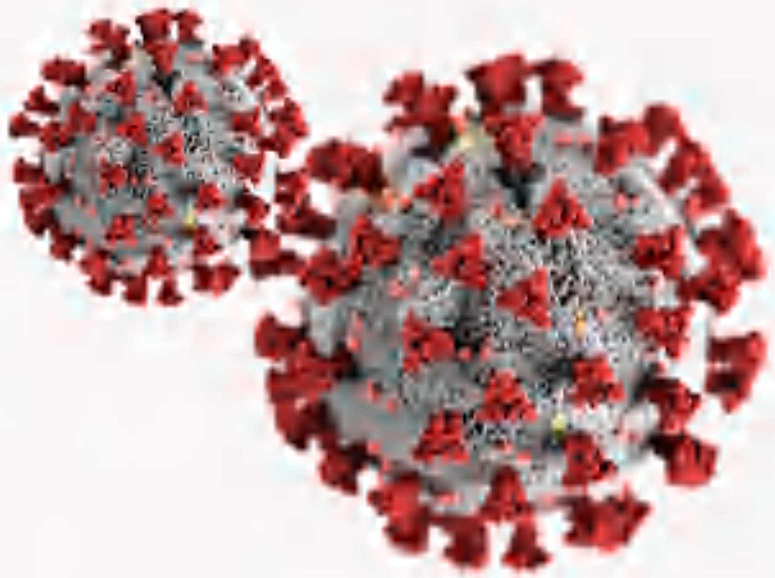
Business Problem

We wanted to forecast the likelihood that a person would be detected by Covid 19 based on a few characteristics using machine learning and Spark.



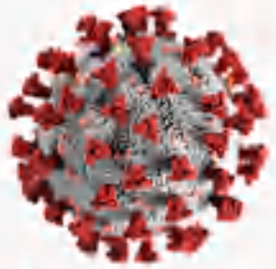
Data Review

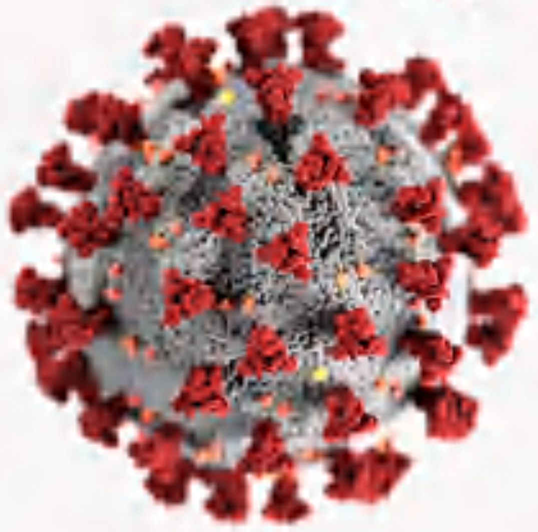
- The dataset was provided by the Mexican government .
- The raw dataset consists of 21 unique features and 1,048,576 unique patients.
- This dataset has 21 columns and 1048575 rows.



Data Review

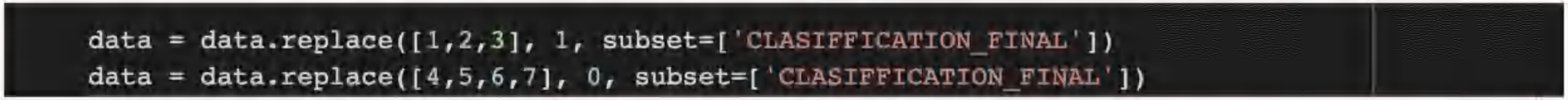
sex	female or male
age	of the patient.
classification	covid test findings. Values 1-3 mean that the patient was diagnosed with covid in different degrees. 4 or higher means that the patient is not a carrier of covid or that the test is inconclusive.
patient type	hospitalized or not hospitalized.
pneumonia	whether the patient already have air sacs inflammation or not.
pregnancy	whether the patient is pregnant or not.
diabetes	whether the patient has diabetes or not.
copd	Indicates whether the patient has Chronic obstructive pulmonary disease or not.
asthma	whether the patient has asthma or not.
inmsupr	whether the patient is immunosuppressed or not.
hypertension	whether the patient has hypertension or not.
cardiovascular	whether the patient has heart or blood vessels related disease.
renal chronic	whether the patient has chronic renal disease or not.
other disease	whether the patient has other disease or not.
obesity	whether the patient is obese or not.
tobacco	whether the patient is a tobacco use
usmr	Indicates whether the patient treated medical units of the first, second or third level.
medical unit	type of institution of the National Health System that provided the care.
intubed	whether the patient was connected to the ventilator
icu	Indicates whether the patient had been admitted to an Intensive Care Unit
death	indicates whether the patient died or recovered





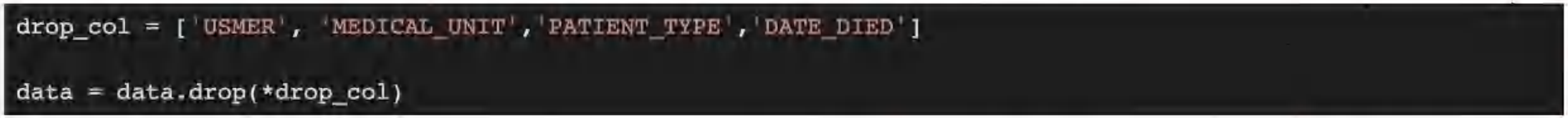
Data Preprocessing

- replace the value 1,2,3 by 1 and 4,5,6,7 by 0 (Values 1-3 mean that the patient was diagnosed with covid in different degrees, 4 or higher means that the patient is not a carrier of covid or that the test is inconclusive)



```
    data = data.replace([1,2,3], 1, subset=['CLASIFFICATION_FINAL'])
    data = data.replace([4,5,6,7], 0, subset=['CLASIFFICATION_FINAL'])
```

- drop un unnecessary columns

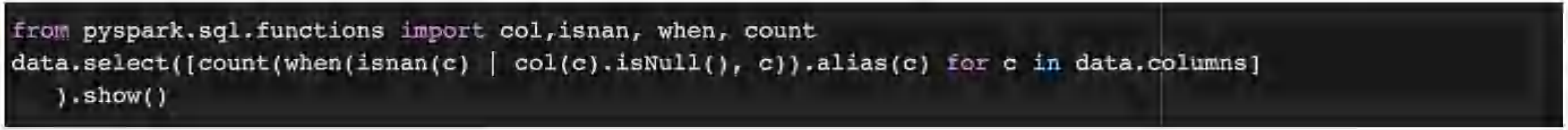


```
drop_col = ['USMER', 'MEDICAL_UNIT','PATIENT_TYPE','DATE_DIED']

data = data.drop(*drop_col)
```

Cleaning Data

- check the null value

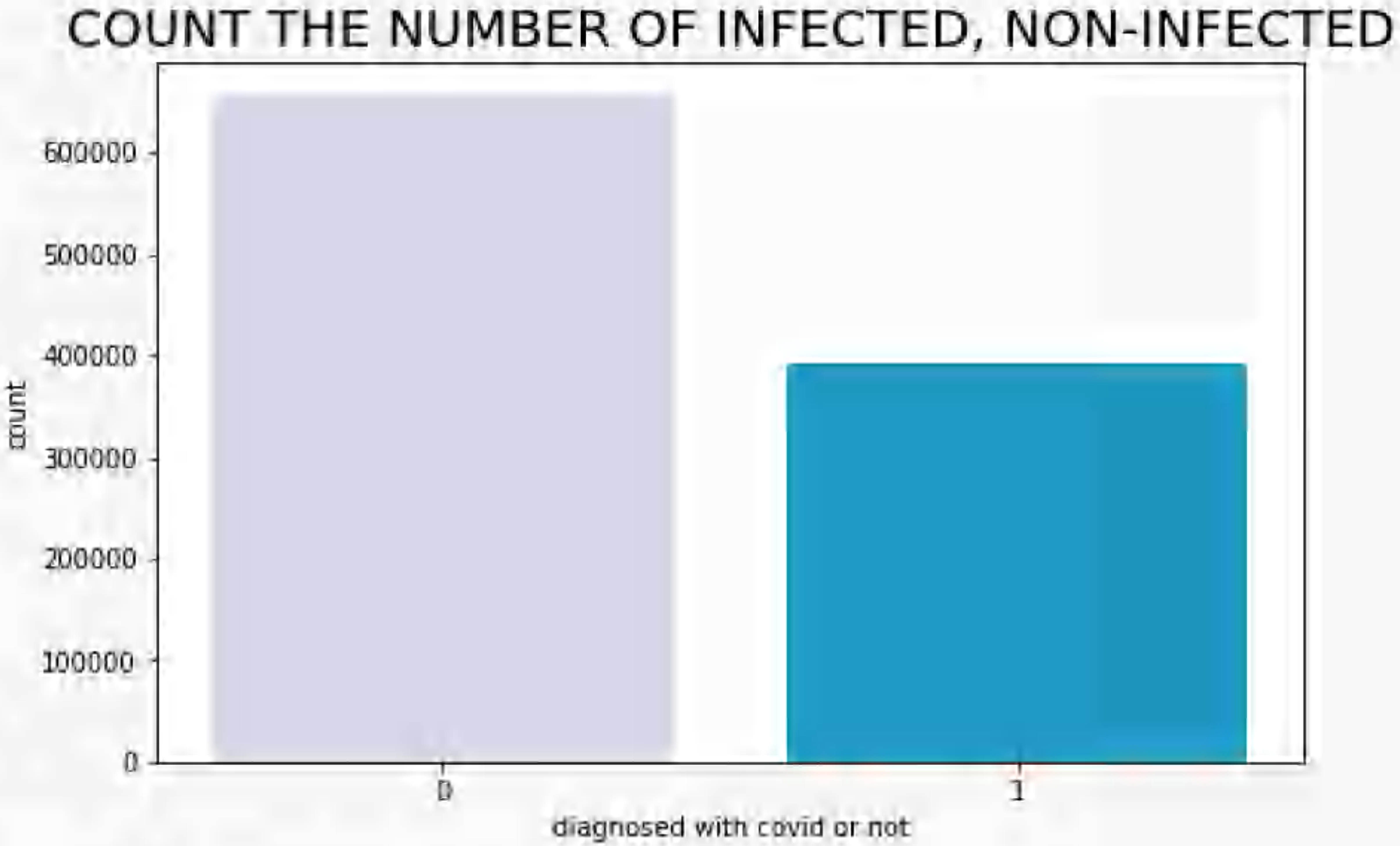


```
from pyspark.sql.functions import col,isnan, when, count
data.select([count(when(isnan(c) | col(c).isNull(), c)).alias(c) for c in data.columns]
   ).show()
```

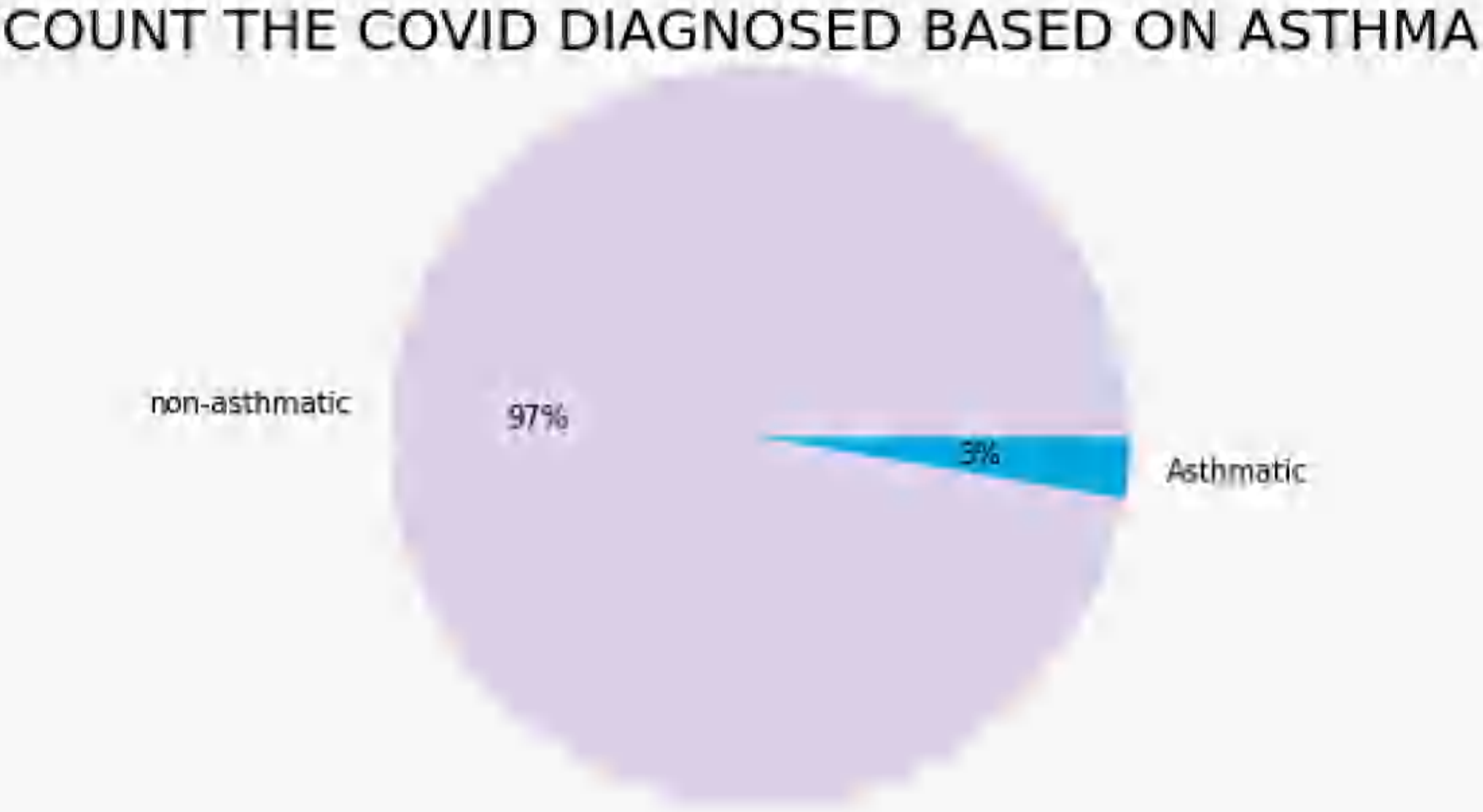
- we didn't have any null value .

Exploratory Data Aanalysis (EDA)

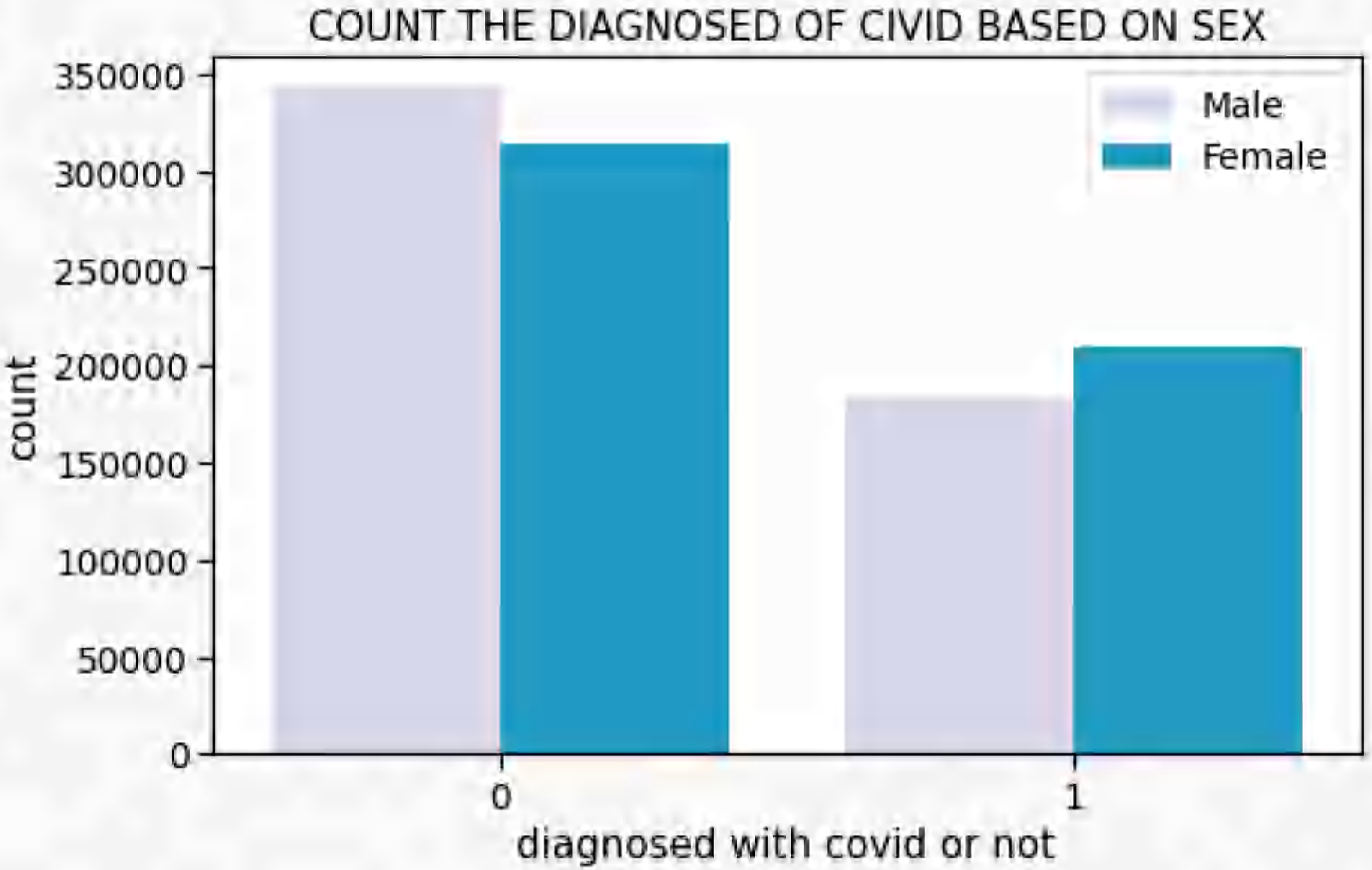
1st Plot


COUNT THE NUMBER OF INFECTED, NON-INFECTED
count
600000
500000
400000
300000
200000
100000
0
0
1
diagnosed with covid or not

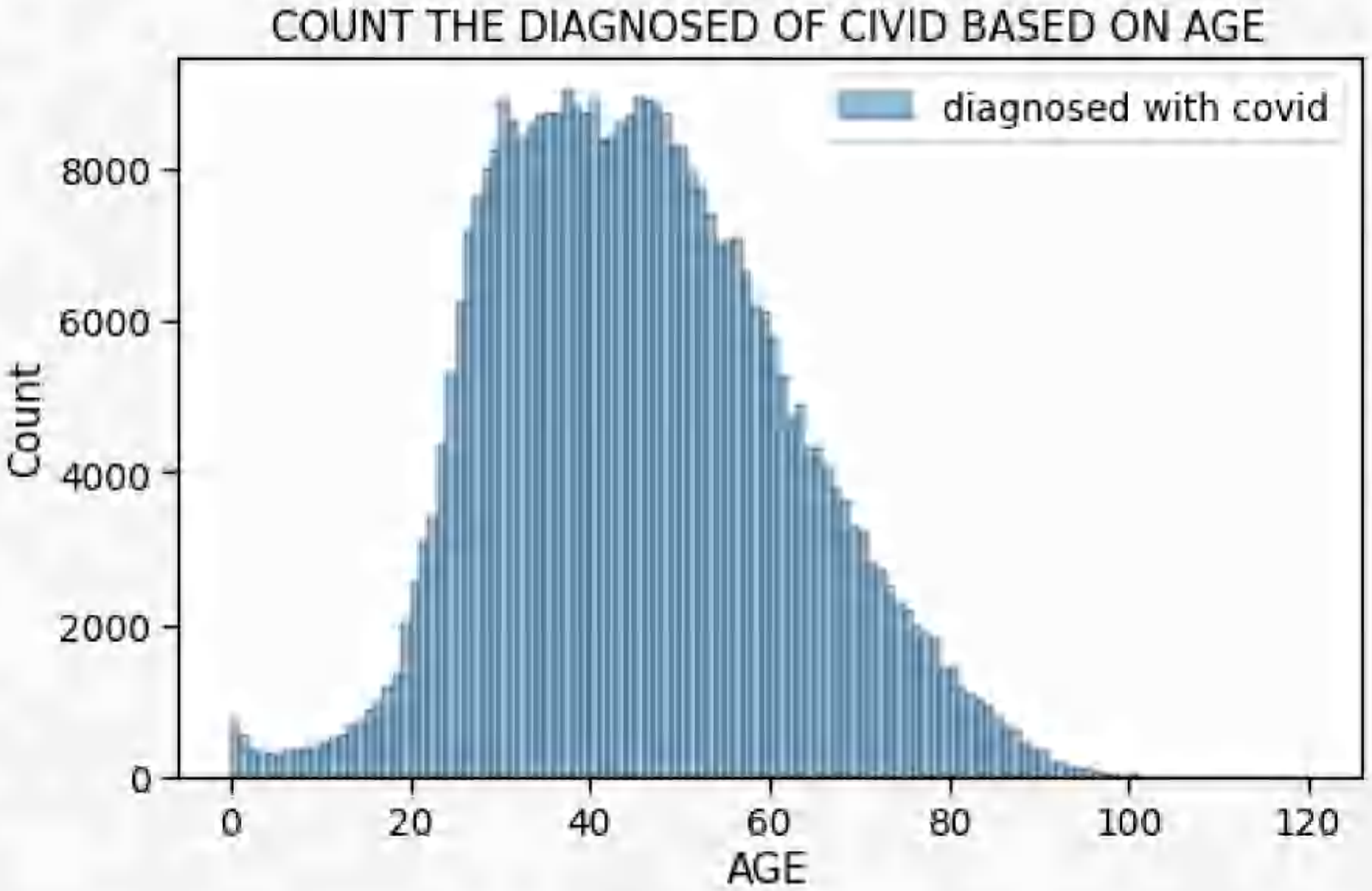
2nd Plot


COUNT THE COVID DIAGNOSED BASED ON ASTHMA
non-asthmatic
97%
3%
Asthmatic

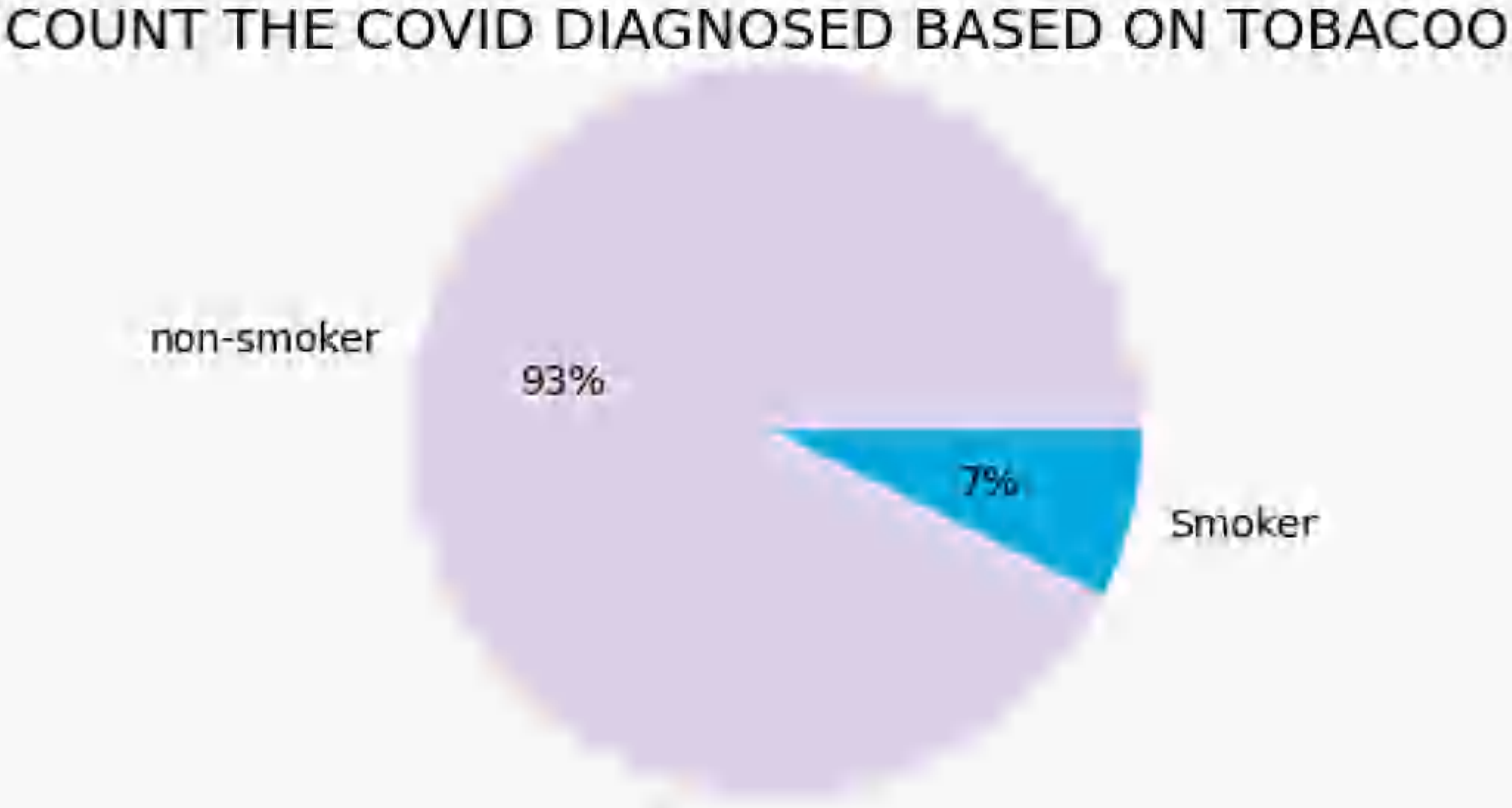
3rd Plot


COUNT THE DIAGNOSED OF CIVID BASED ON SEX
Male
Female
count
350000
300000
250000
200000
150000
100000
50000
0
0
1
diagnosed with covid or not

4th Plot

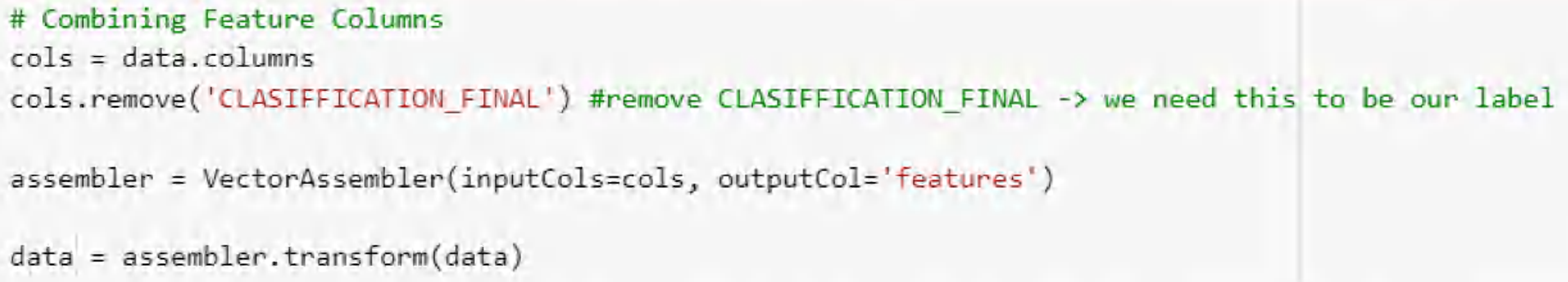

COUNT THE DIAGNOSED OF CIVID BASED ON AGE
diagnosed with covid
Count
0
2000
4000
6000
8000
AGE
0
20
40
60
80
100
120

Fifth plot


COUNT THE COVID DIAGNOSED BASED ON TOBACOO
non-smoker
93%
7%
Smoker

Machine Learning Models

- Combining Feature Columns

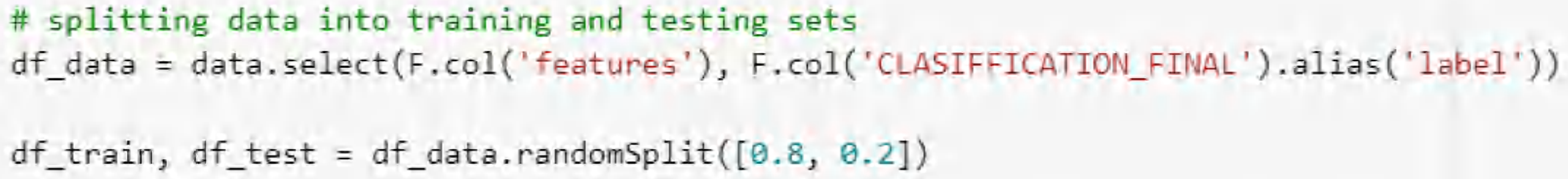


```
# Combining Feature Columns
cols = data.columns
cols.remove('CLASIFFICATION_FINAL') #remove CLASIFFICATION_FINAL -> we need this to be our label

assembler = VectorAssembler(inputCols=cols, outputCol='features')

data = assembler.transform(data)
```

- splitting data into training and testing sets



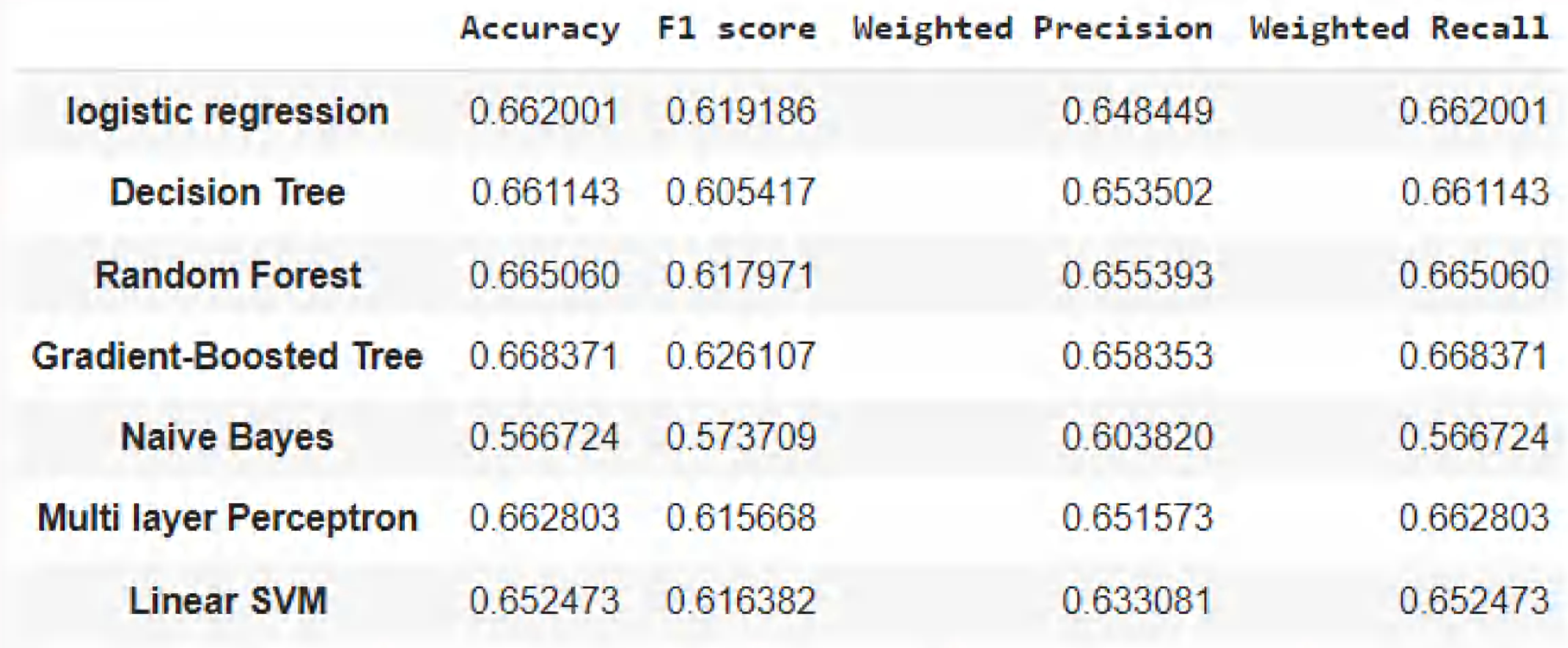
```
# splitting data into training and testing sets
df_data = data.select(F.col('features'), F.col('CLASIFFICATION_FINAL').alias('label'))

df_train, df_test = df_data.randomSplit([0.8, 0.2])
```

Model Building

- Logistic Regression
- Decision Tree Classifier
- Random Forest Classifier
- Gradient-Boosted Tree Classifier
- NaiveBayes
- Multi layer Perceptron Classifier
- Linear Support Vector Machine

Results



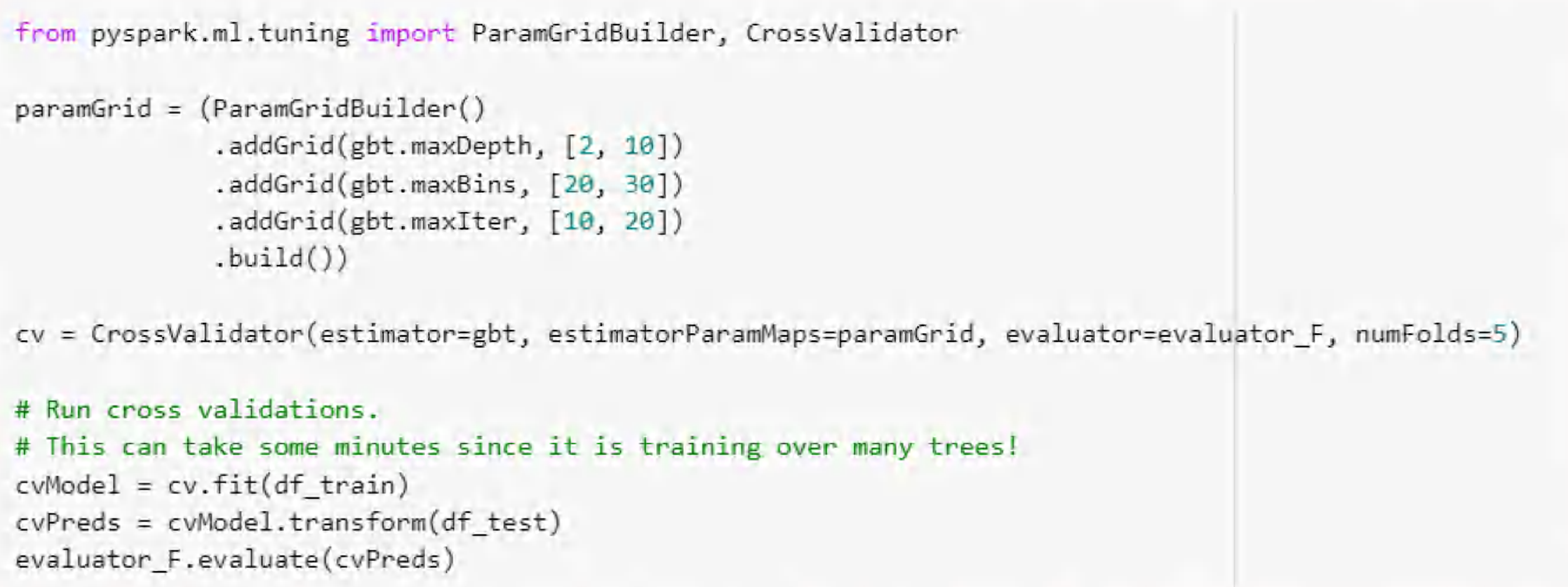
	Accuracy	F1 score	Weighted Precision	Weighted Recall
logistic regression	0.662001	0.619186	0.648449	0.662001
Decision Tree	0.661143	0.605417	0.653502	0.661143
Random Forest	0.665060	0.617971	0.655393	0.665060
Gradient-Boosted Tree	0.668371	0.626107	0.658353	0.668371
Naive Bayes	0.566724	0.573709	0.603820	0.566724
Multi layer Perceptron	0.662803	0.615668	0.651573	0.662803
Linear SVM	0.652473	0.616382	0.633081	0.652473

Model Selection

Since Our data is not balanced, so in choosing the best model, we must focus on the F1 score.

Gradient-Boosted Tree has the highest F1 (0.626283), We choose it as the best model.

Tuning Gradient-Boosted Tree model with the ParamGridBuilder and the CrossValidator



```python
from pyspark.ml.tuning import ParamGridBuilder, CrossValidator

paramGrid = (ParamGridBuilder()
             .addGrid(gbt.maxDepth, [2, 10])
             .addGrid(gbt.maxBins, [20, 30])
             .addGrid(gbt.maxIter, [10, 20])
             .build())

cv = CrossValidator(estimator=gbt, estimatorParamMaps=paramGrid, evaluator=evaluator_F, numFolds=5)

# Run cross validations.
# This can take some minutes since it is training over many trees!
cvModel = cv.fit(df_train)
cvPreds = cvModel.transform(df_test)
evaluator_F.evaluate(cvPreds)
```

0.6258819211953449

Thank you!

Have a great day ahead.