Structured Data Assignment

Thadimudupula Sahith

GUVI

001

Introduction

The dataset in question contains a comprehensive collection of electronic health records belonging to patients who have been diagnosed with a specific disease. These health records comprise a detailed log of every aspect of the patients' medical history including all diagnoses symptoms prescribed drug treatments and medical tests that they have undergone. Each row represents a healthcare record/medical event for a patient and it includes a timestamp for each entry/event thereby allowing for a chronological view of the patient's medical history.

The Data has mainly three columns

1. **Patient-Uid** - Unique Alphanumeric Identifier for a patient
2. **Date** - Date when patient encountered the event.
3. **Incident** - This column describes which event occurred on the day.

Problem Statement

The objective is to construct a prognostic model capable of ascertaining the potential eligibility of a patient for the "Target Drug" within a forthcoming period of 50 days. The provision of this information is of utmost importance for clinicians to make well-informed decisions regarding treatment alternatives.

Problem

The primary challenge is to a binary classification undertaking, wherein the objective is to categorize patients into two distinct groups: those who are deemed eligible (1) or ineligible (0) for the "Target Drug" based on their medical records.

Objective

The purpose of this task is to develop a reliable predictive model that, within the next half-century, can determine whether or not a particular patient is qualified to receive the "Target Drug." The purpose of this model is to provide medical professionals with a tool that will assist them in making better educated treatment decisions.

Potential Applications of Problem Statement

The developed model may have uses in the medical and pharmaceutical fields in the future. It can be utilized to: Assist medical professionals in the process of developing individualized treatment regimens for patients.

Improve the production and distribution of drugs by optimizing their strategies.

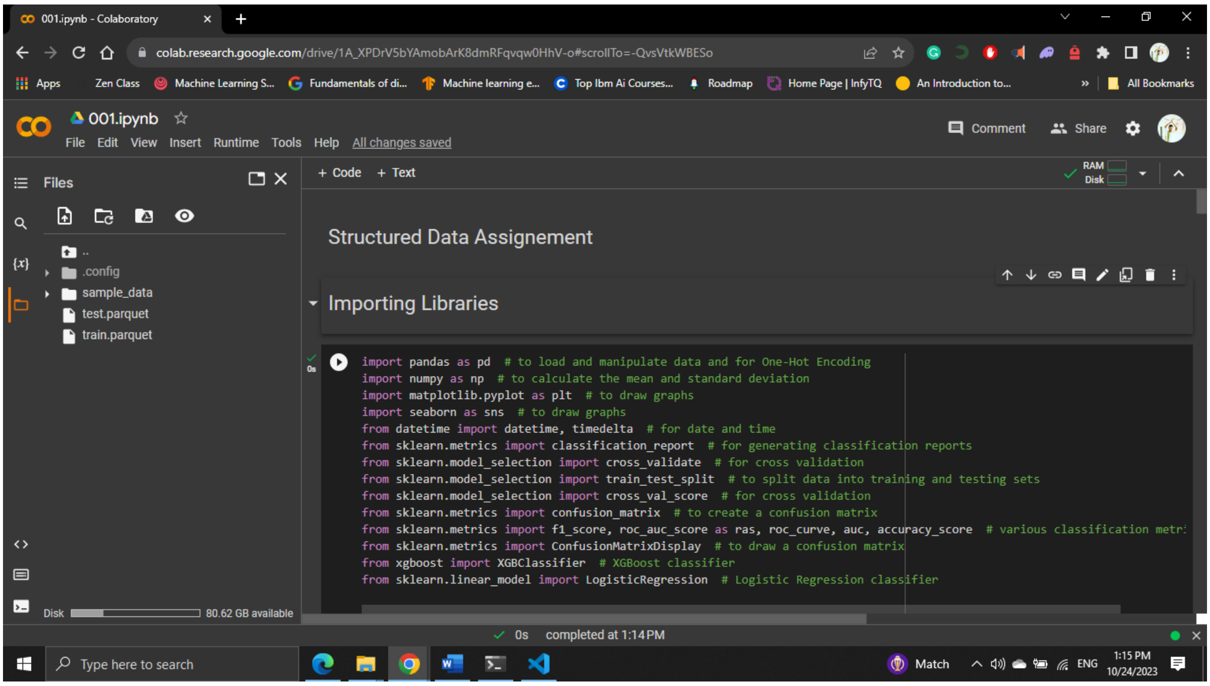
Enhance the quality of care provided to patients and the sector as a whole.

Reduce the risk of adverse reactions to medications.

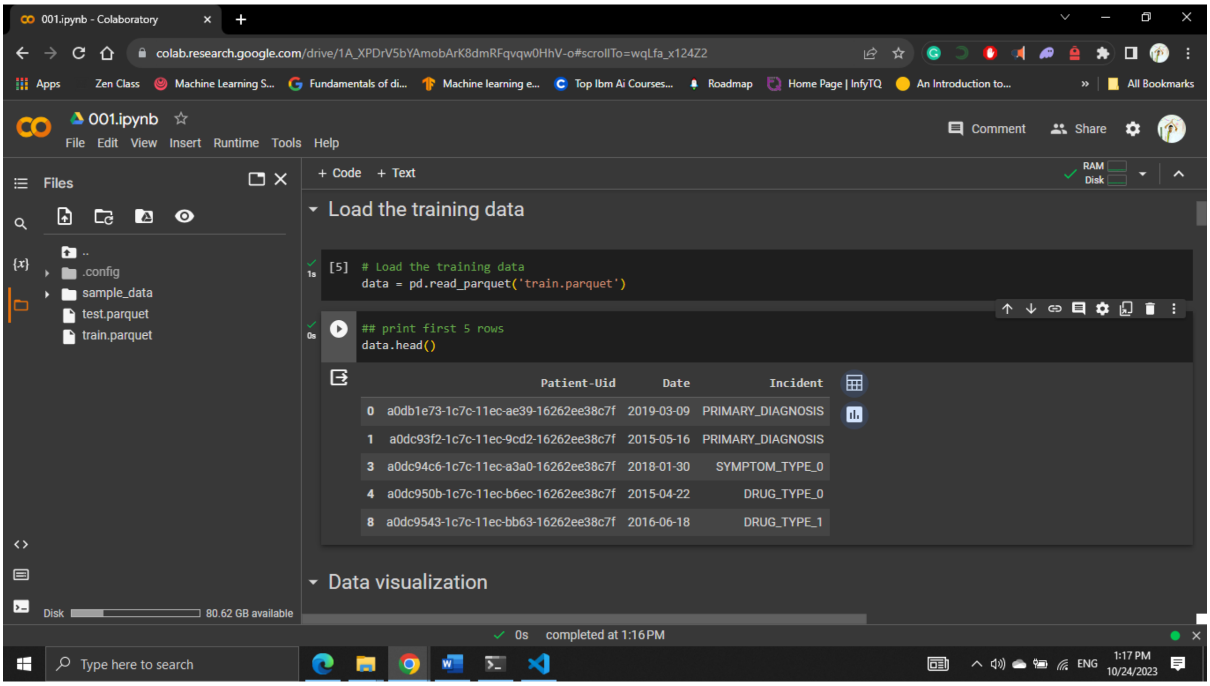
Type of Machine Learning Problem

The task at hand pertains to a binary classification scenario whereby the objective is to make predictions regarding the eligibility of a patient for the "Target Drug" during a span of 30 days. These predictions are based on an analysis of the patient's medical history. The outcome variable is represented by two distinct classes: 1 denotes eligibility for the drug, while 0 signifies ineligibility. Additionally this problem may involve time-series analysis and feature engineering techniques to incorporate temporal aspects into the predictive model.

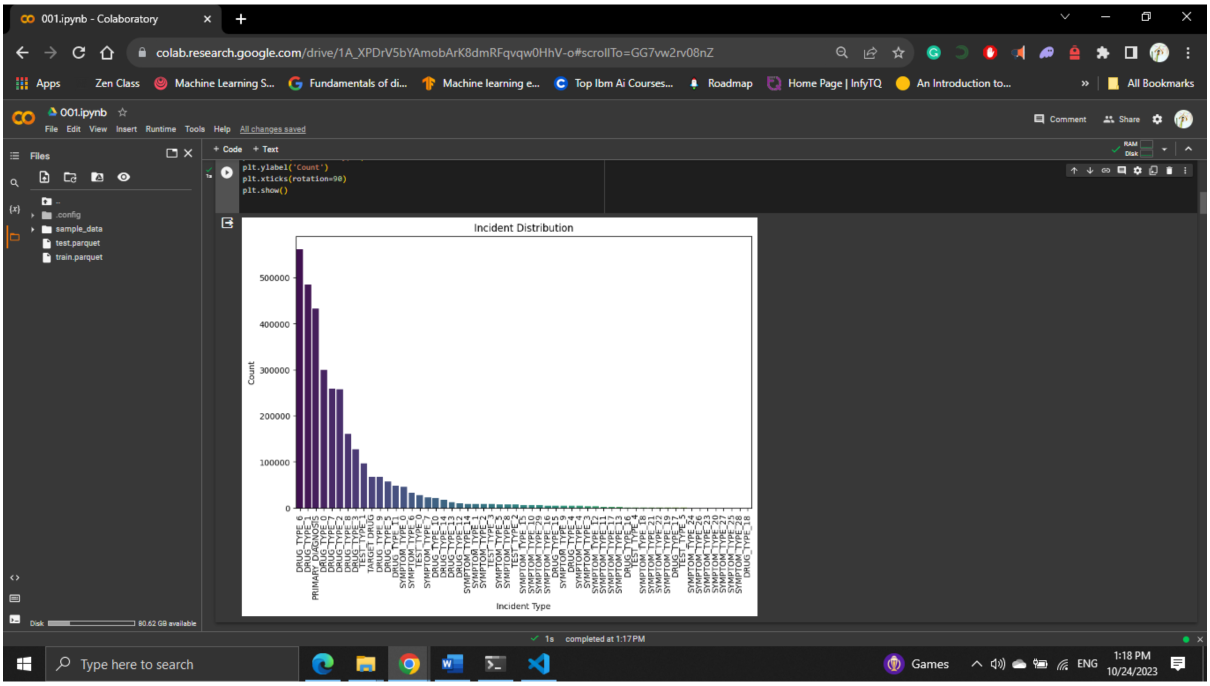
Importing Libraries



Load the training data

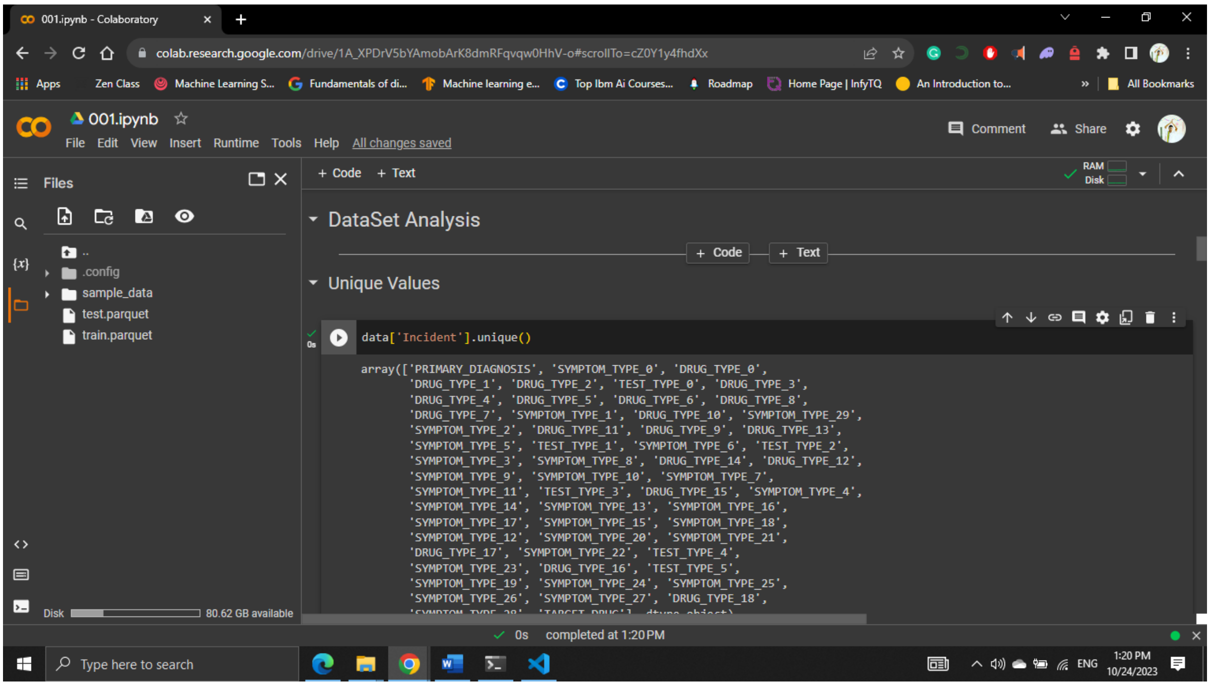


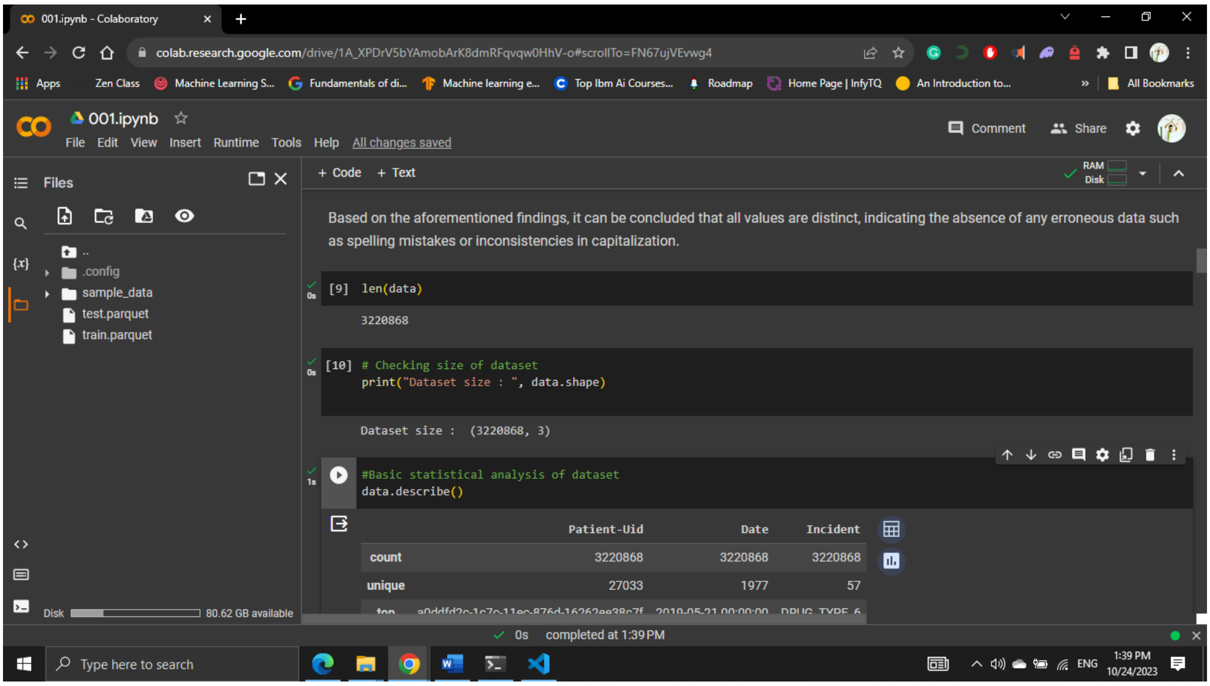
Data visualization

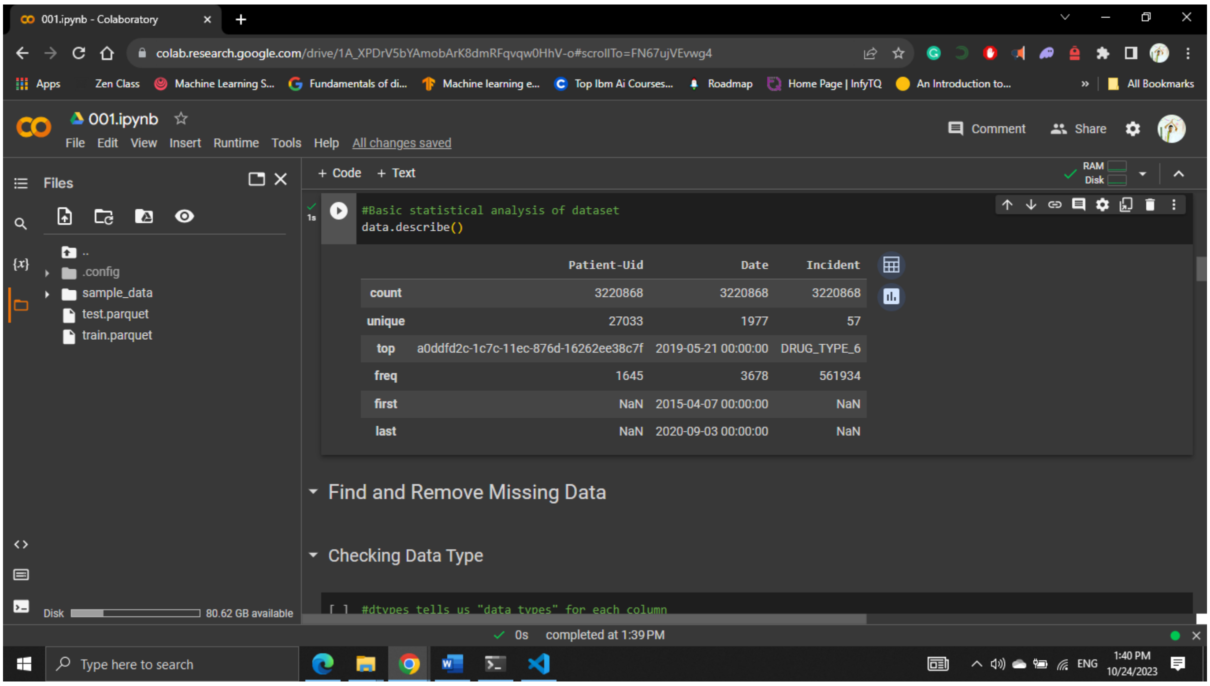


Dataset Analysis

Unique Values in Incident column







Data Summary

Total Rows: 3,220,868

Columns

Patient-Uid: Unique alphanumeric identifier for patients

Date: Date when the event occurred

Incident: Describes the type of event that occurred (e.g., primary diagnosis, symptom type, drug type)

Unique Incidents

There are a total of 57 unique types of incidents.

Most Frequent Incidents

The most frequent incident is **DRUG_TYPE_6** with 549,616 occurrences, followed by **DRUG_TYPE_1** with 484,565 occurrences, and **PRIMARY_DIAGNOSIS** with 424,879 occurrences.

Occurrences of "Target Drug"

There are 67,218 occurrences of the "TARGET DRUG" incident.

Data Range

The data spans from April 7, 2015, to September 3, 2020.

Number of Duplicates

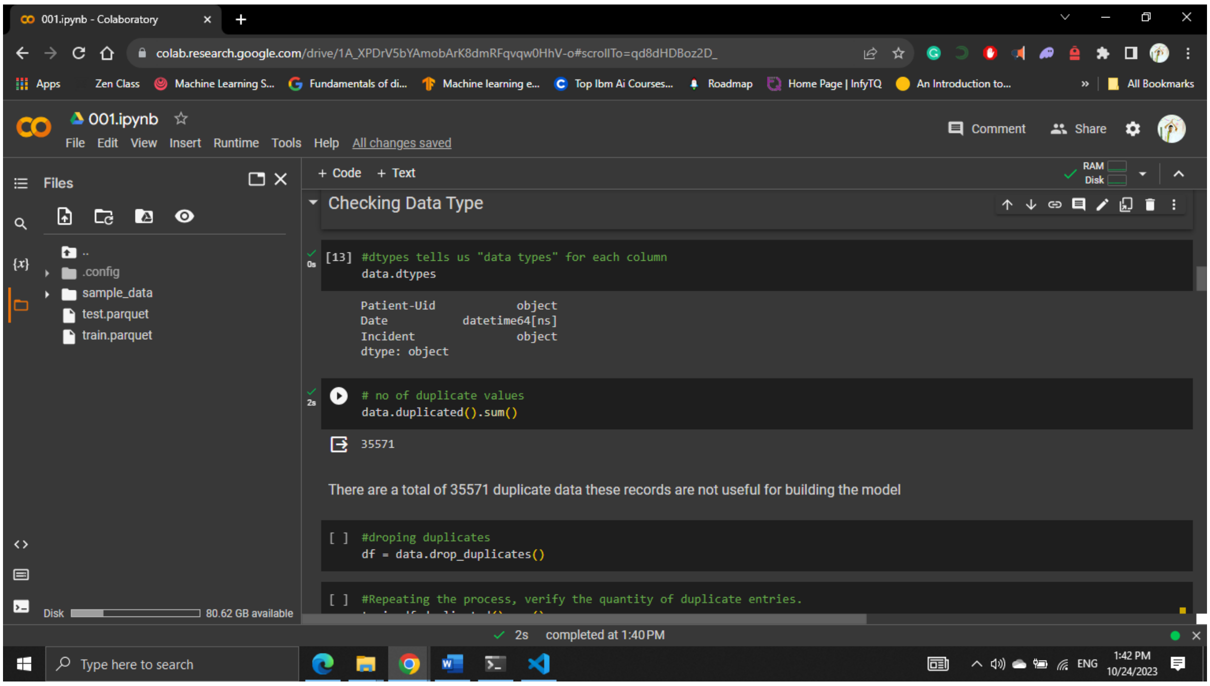
There are 35,571 duplicate rows in the dataset.

Value Counts for Each Incident Type

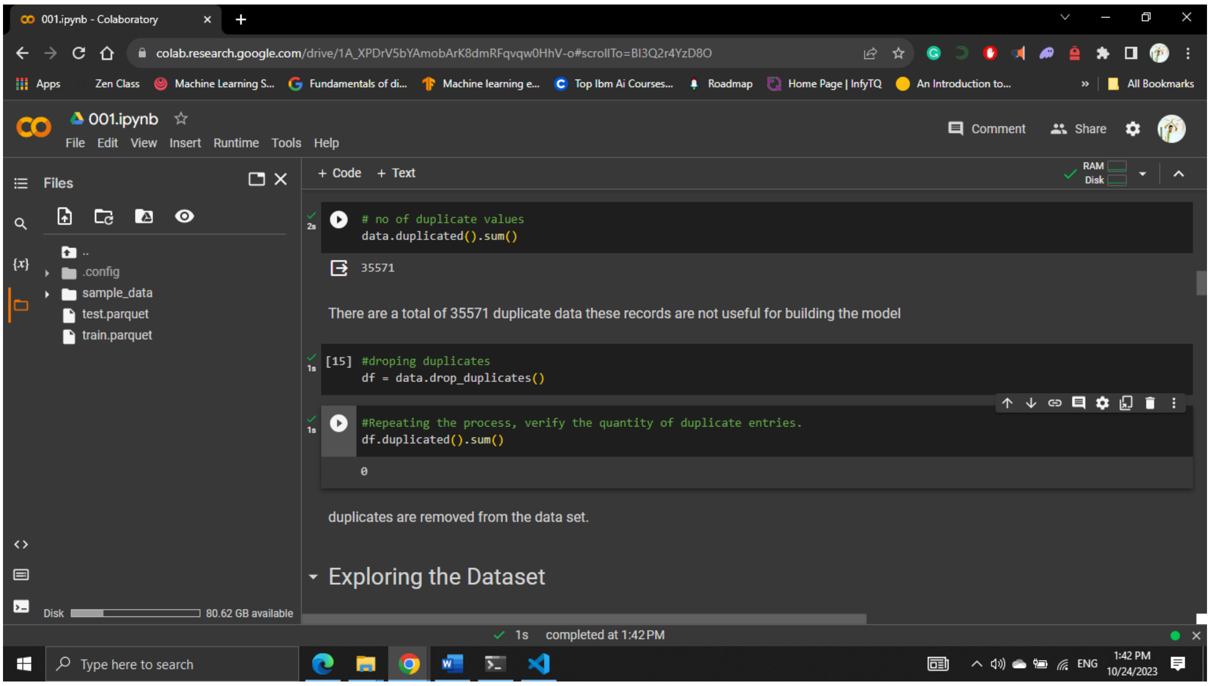
DRUG_TYPE_6 occurs most frequently with 549,616 instances followed by **DRUG_TYPE_1** with 484,565 instances, and **PRIMARY_DIAGNOSIS** with 424,879 instances.

There's a wide range of incident types including various drug types symptom types and test types.

Find and Remove Missing Data



Duplicates Are Removed from The Data Set.

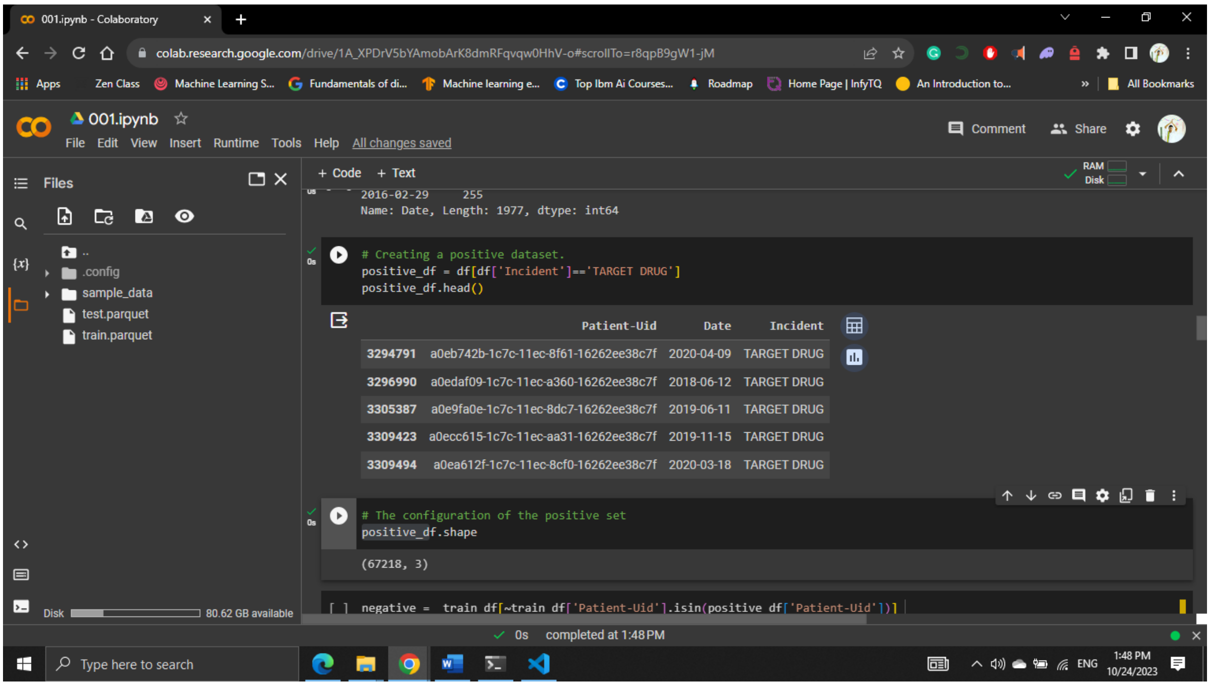


Positive Dataset.

positive_df: This is a new DataFrame that will store the positive examples. These examples represent instances where the incident is 'TARGET DRUG' meaning that the patient has taken the "Target Drug".

df[df['Incident']=='TARGET DRUG']: This line of code filters the df DataFrame to select only the rows where the value in the 'Incident' column is 'TARGET DRUG'. In other words it selects all the records where the incident is the "Target Drug".

.head(): This function is used to display the first few rows of the DataFrame. By default it displays the first 5 rows. This is useful for quickly inspecting the data. show the first 5 rows.



Negative Dataset

negative = df[~df['Patient-Uid'].isin(positive_df['Patient-Uid'])]:

df['Patient-Uid'].isin(positive_df['Patient-Uid']) checks if the **'Patient-Uid'** values in **df** are present in the **'Patient-Uid'** values of **positive_df**.

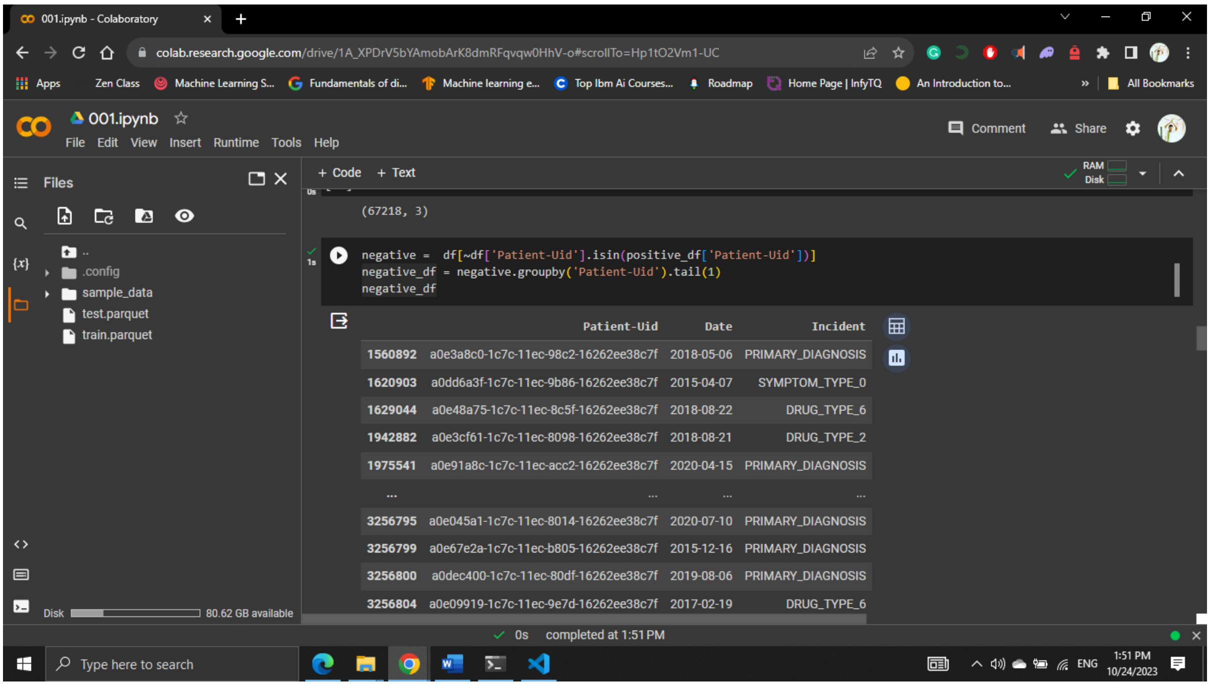
~ is a logical negation operator so **~df['Patient-Uid'].isin(positive_df['Patient-Uid'])** will return **True** for rows where **'Patient-Uid'** is not in **positive_df**.

This filters the DataFrame **df** to select only the rows where the **'Patient-Uid'** is not in the positive set. These are the negative examples (Starmer, 2022).

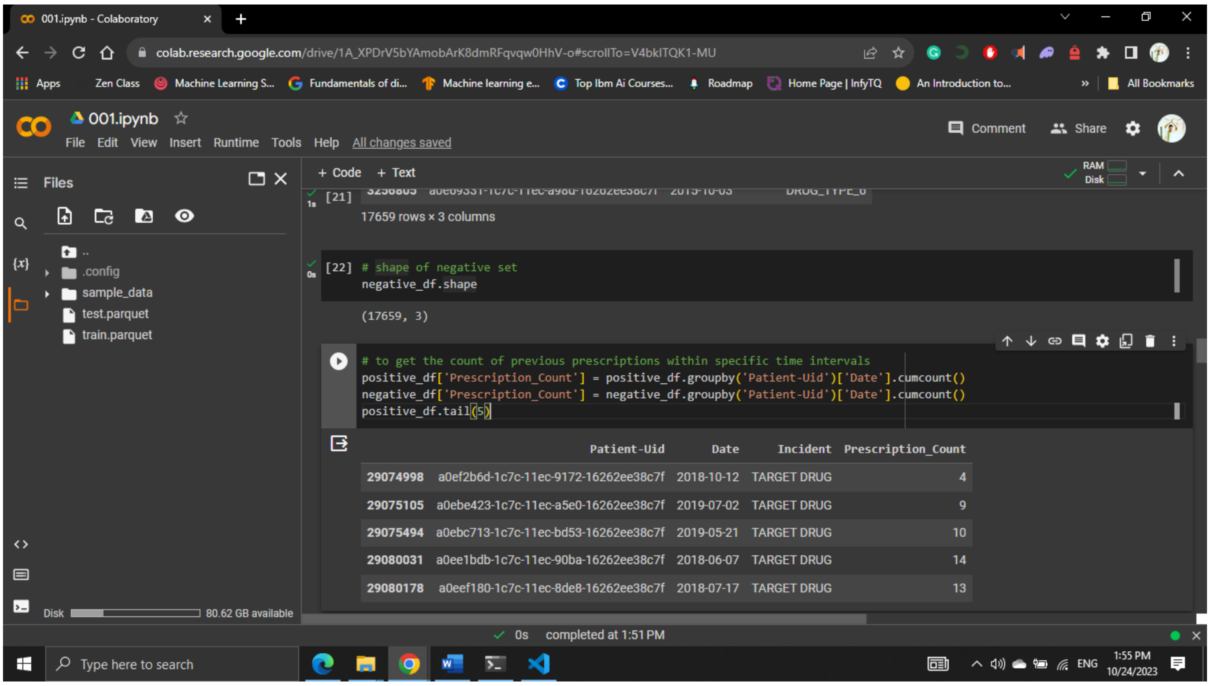
negative_df = negative.groupby('Patient-Uid').tail(1)

.groupby('Patient-Uid') groups the Data Frame by the unique **'Patient-Uid'** values.

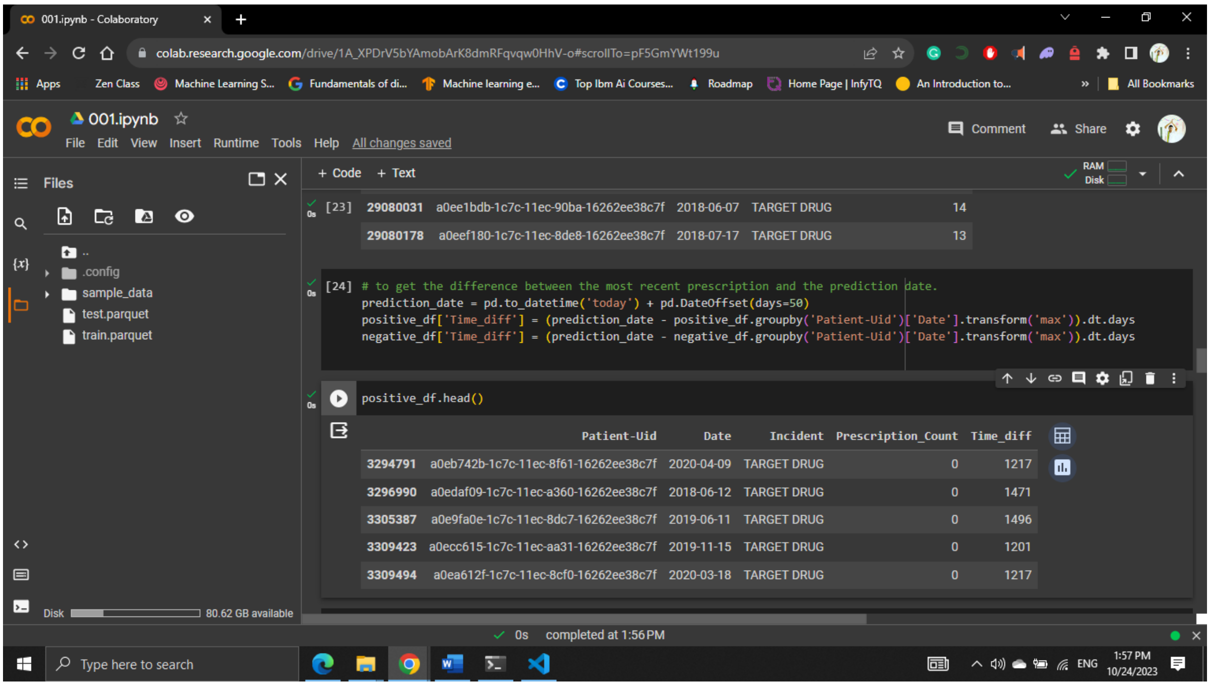
.tail(1) selects the last row from each group. This is assuming that the data is sorted by time so the last entry for each patient is the most recent.

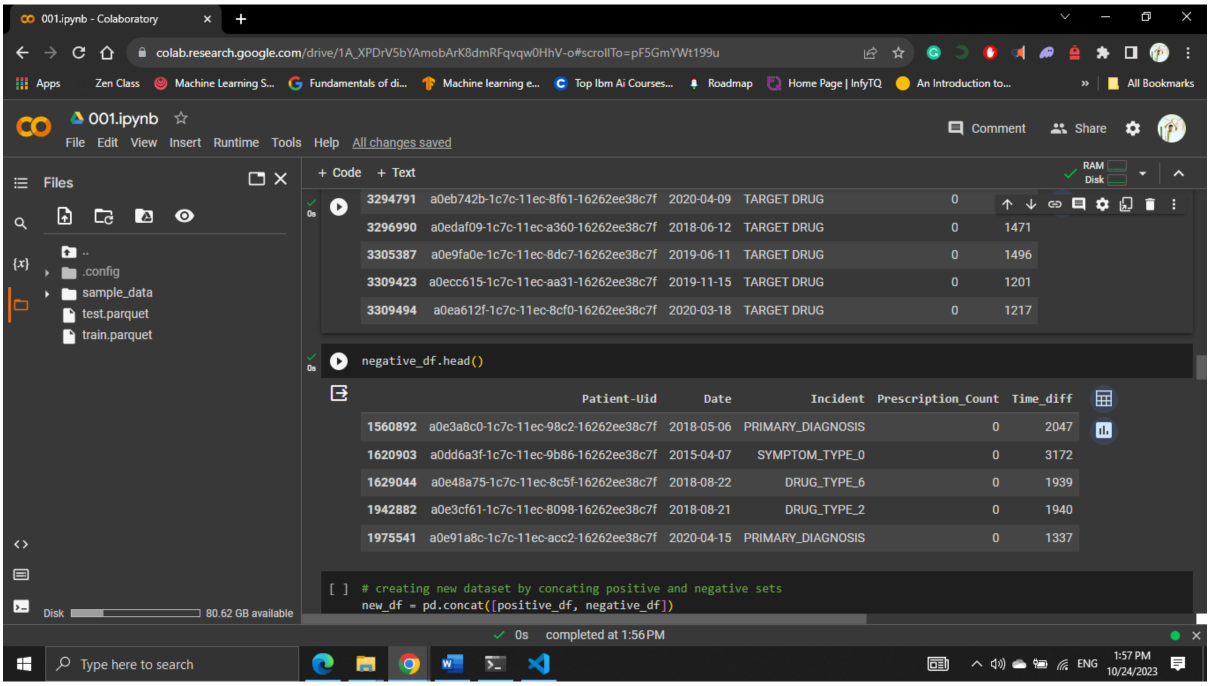


To get the count of previous prescriptions within specific time intervals

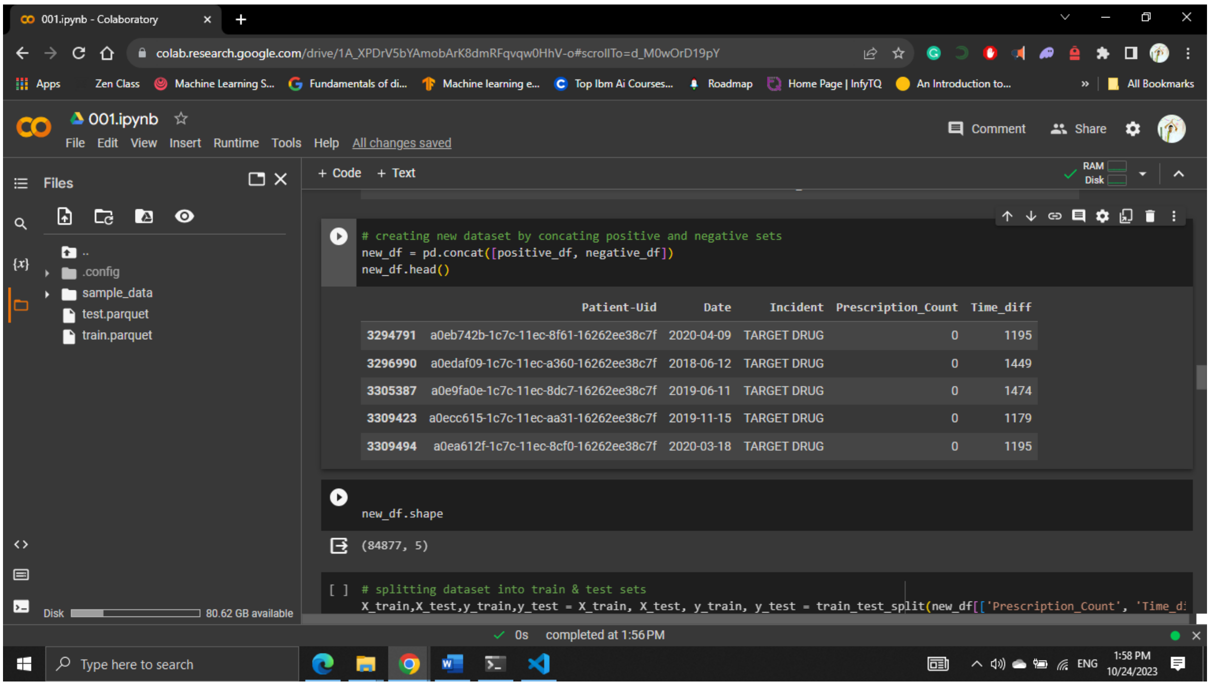


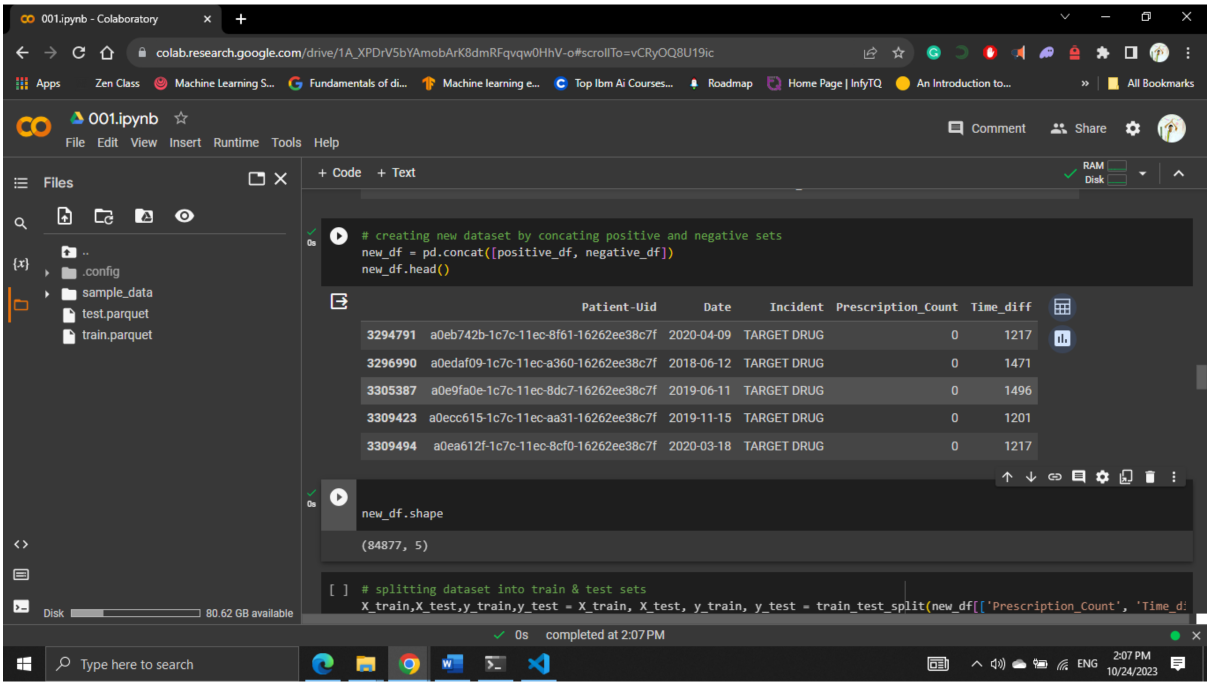
To get the difference between the most recent prescription and the prediction date.



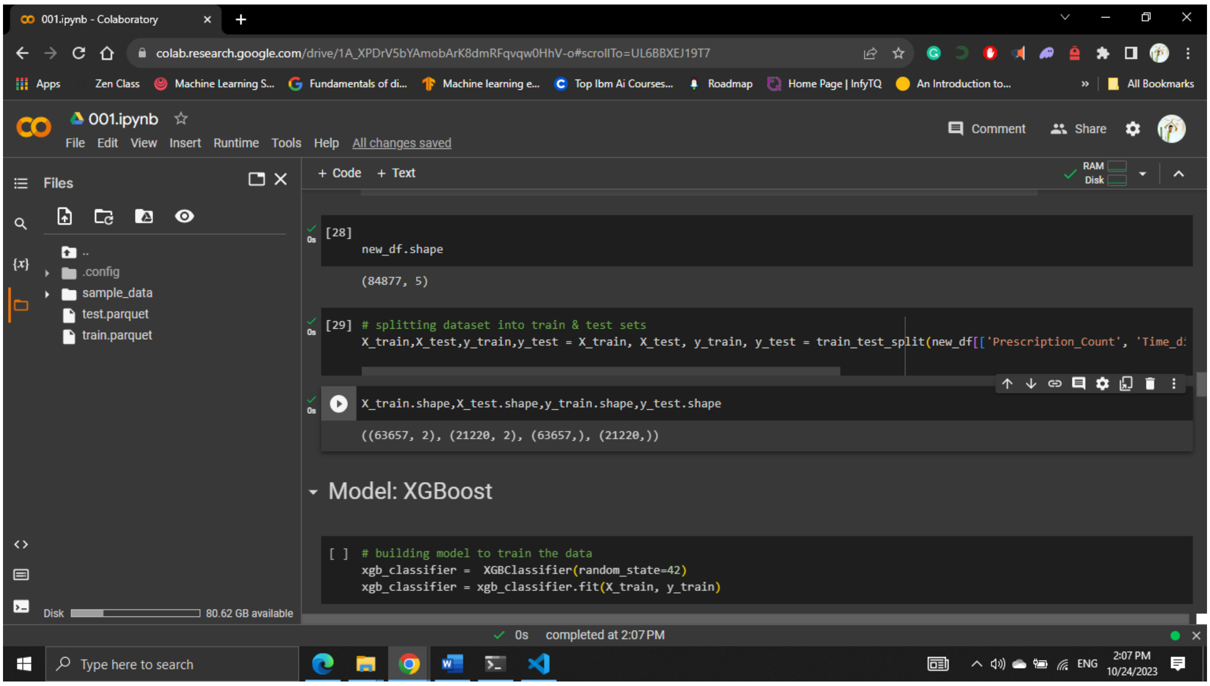


creating new dataset by concating positive and negative sets

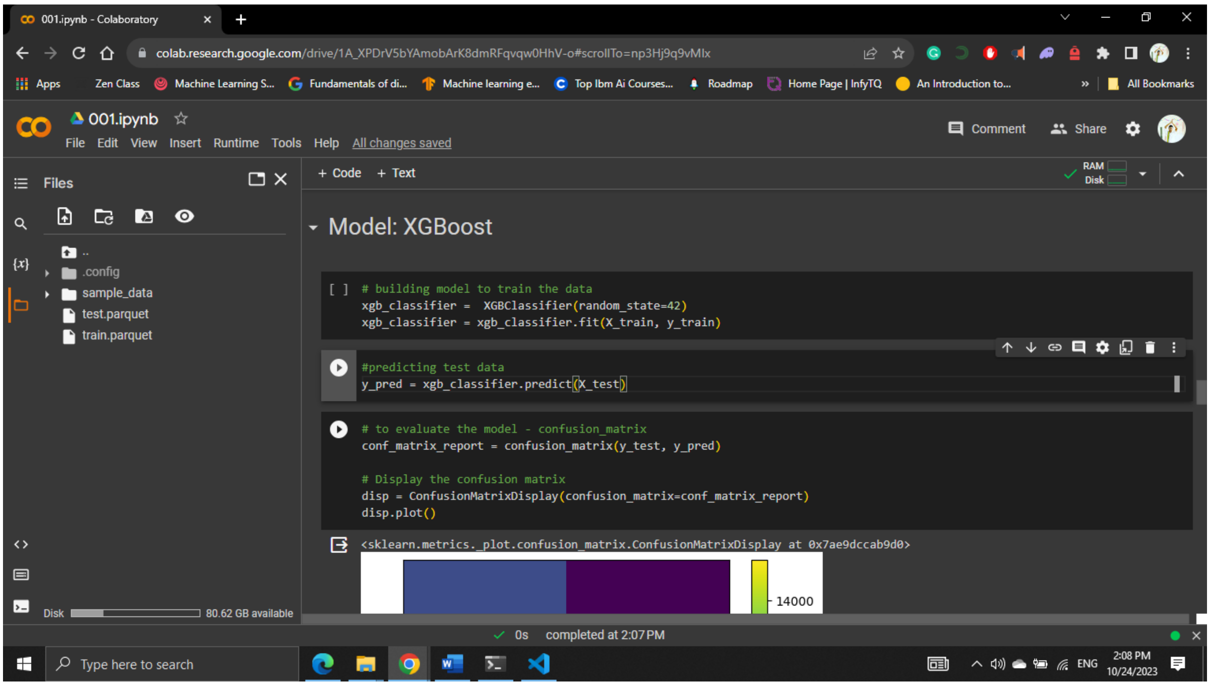


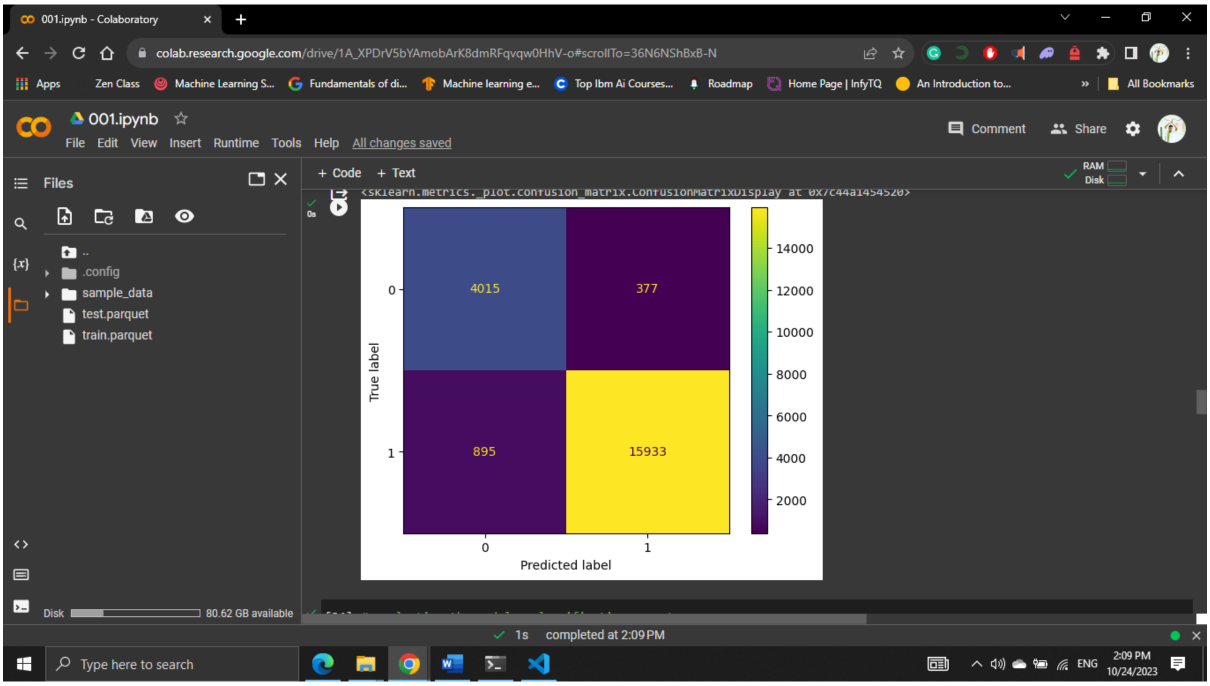


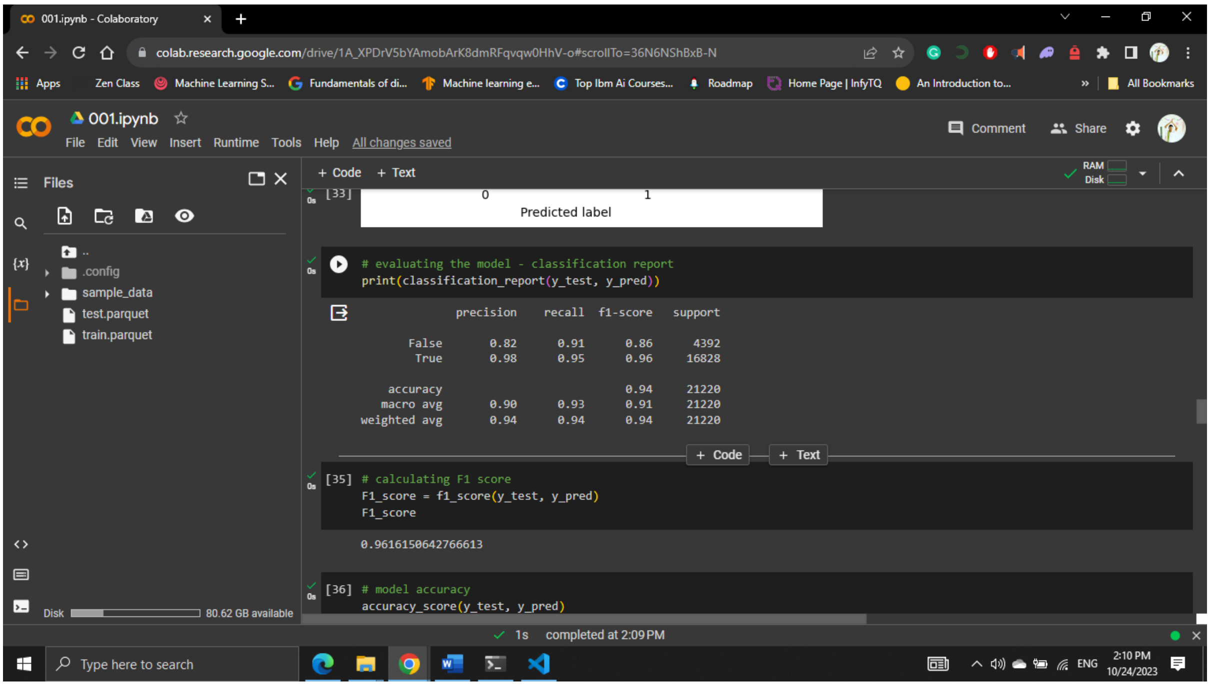
splitting dataset into train & test sets

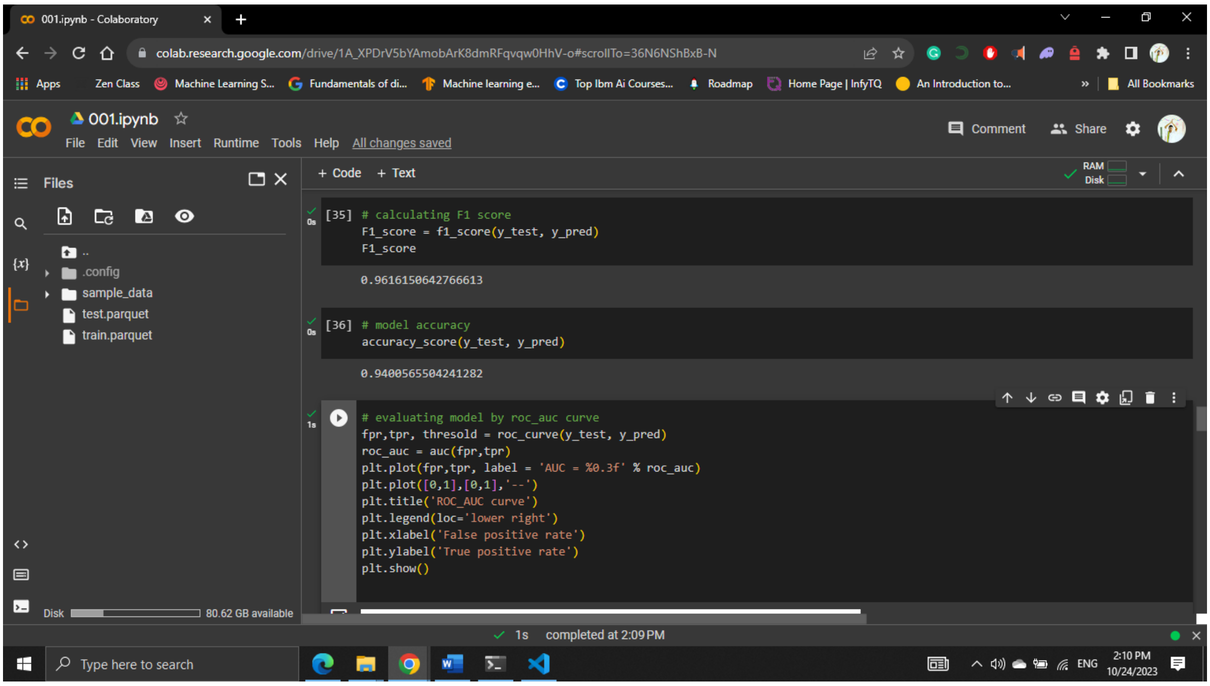


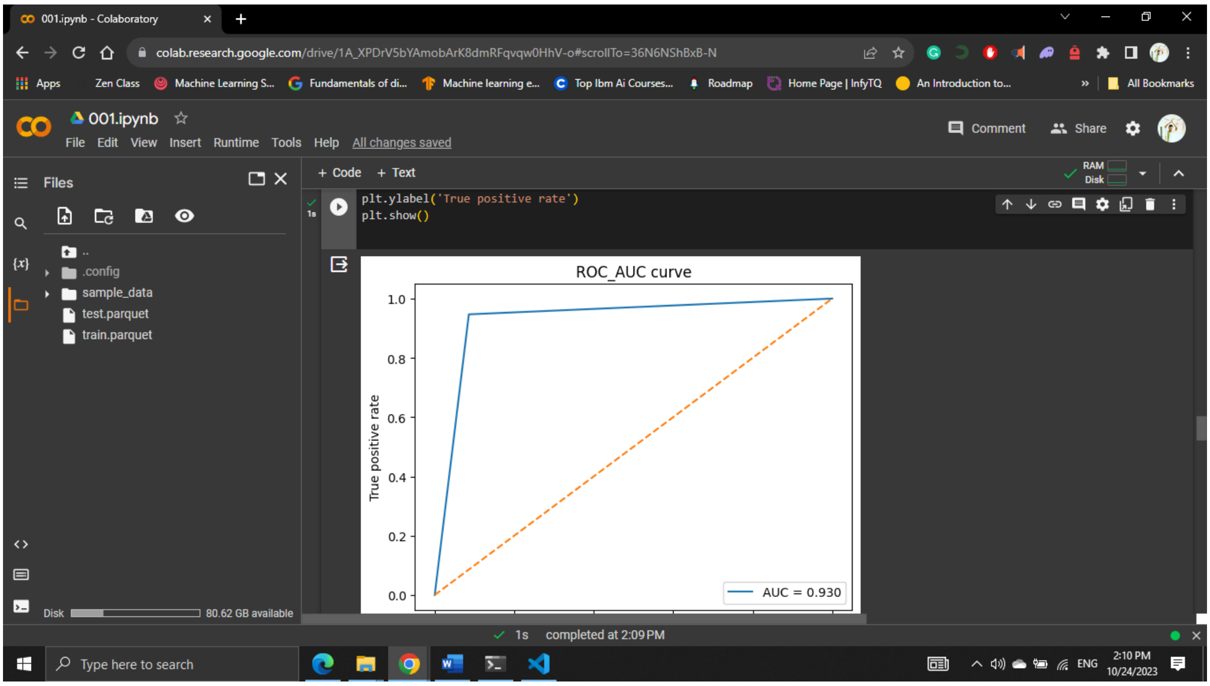
Model: XGBoost






001.ipynb - Colaboratory
colab.research.google.com/drive/1A_XPDrV5bYAmobArK8dmRFqvqw0HhV-o#scrollTo=36N6NShBxB-N
001.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Files
.config
sample_data
test.parquet
train.parquet
Predicted label
# evaluating the model - classification report
print(classification_report(y_test, y_pred))
precision recall f1-score support
False 0.82 0.91 0.86 4392
True 0.98 0.95 0.96 16828
accuracy 0.94 21220
macro avg 0.90 0.93 0.91 21220
weighted avg 0.94 0.94 0.94 21220
[35] # calculating F1 score
F1_score = f1_score(y_test, y_pred)
F1_score
0.9616150642766613
[36] # model accuracy
accuracy_score(y_test, y_pred)
Disk 80.62 GB available
completed at 2:09 PM


001.ipynb - Colaboratory
colab.research.google.com/drive/1A_XPDrV5bYAmobArK8dmRFqvqw0HhV-o#scrollTo=36N6NShBxB-N
001.ipynb
[35] # calculating F1 score
F1_score = f1_score(y_test, y_pred)
F1_score
0.9616150642766613
[36] # model accuracy
accuracy_score(y_test, y_pred)
0.9400565504241282
evaluating model by roc_auc curve
fpr,tpr, thresold = roc_curve(y_test, y_pred)
roc_auc = auc(fpr,tpr)
plt.plot(fpr,tpr, label = 'AUC = %0.3f' % roc_auc)
plt.plot([0,1],[0,1],'--')
plt.title('ROC_AUC curve')
plt.legend(loc='lower right')
plt.xlabel('False positive rate')
plt.ylabel('True positive rate')
plt.show()
Disk 80.62 GB available
completed at 2:09 PM



Building and Training the Model

The XGBoost Classifier is utilized to build a predictive model. This classifier is known for its effectiveness in various machine learning tasks.

xgb_classifier = XGBClassifier(random_state=42) XGBoost Classifier with a specific random seed.

xgb_classifier = xgb_classifier.fit(X_train, y_train) trains the model on the training data (**X_train** and **y_train**).

The model is using the XGBoost Classifier a algorithm known for its effectiveness in a wide range of machine learning tasks.

Confusion Matrix Evaluation

The resulting confusion matrix is displayed as:

[[4015, 377], [895, 15933]]

The confusion matrix provides model's predictions. It consists of four values: true negatives (4015) false positives (377) false negatives (895) and true positives (15933).

Classification Report

The classification report provides a concise overview of the performance metrics of the model, encompassing precision recall and F1-score for both the True and False classes. Additionally it offers assistance by providing the support which refers to the frequency of each class in the exam set.

The classification report provides a comprehensive overview of the model's performance by presenting metrics like as precision recall and F1-score for each individual class. The precision recall and F1-score for the 'False' class are 0.82 0.91 and 0.86 respectively. In the 'True' class the precision is 0.98 the recall is 0.95 and the F1-score is 0.96.

Calculating F1 Score

The F1 score is a composite measure that combines precision and recall. The metric gives a singular numerical representation denoting the efficacy of the model. The F1 score denoted as F1_score is computed by applying the f1_score function on the true labels (y_test) and predicted labels (y_pred). The F1 score as determined through calculations is roughly 0.962.

The F1 score a metric that combines precision and recall is estimated to be around 0.962. This observation suggests that the model exhibits excellent performance in achieving a balance between precision and recall.

The Area Under the Curve

In this particular instance the area under the receiver operating characteristic curve (AUC) is calculated to be 0.930. This value indicates that the model has a robust capacity to distinguish between patients who meet the criteria for receiving the "Target Drug" and those who do not (Starmer, 2022).

The computed value for the Area Under the Receiver Operating Characteristic Curve (AUC) is 0.930. The obtained score demonstrates that the model has a significant degree of discriminative capability efficiently discerning between patients who meet the criteria for receiving the 'Target Drug' and those who do not.

Model Accuracy

The measure of model accuracy quantifies the proportion of right predictions relative to the total number of predictions made. The function accuracy_score(y_test, y_pred) computes the accuracy by comparing the genuine labels (y_test) with the predicted labels (y_pred).

The accuracy of the model was determined to be roughly 0.940 indicating that it accurately predicts outcomes in approximately 94% of cases. The model has a notable level of accuracy estimated at around 94% indicating its ability to accurately forecast patient eligibility for the 'Target Drug' in approximately 94% of instances.

References

Starmer, J. (2022). The Statquest illustrated guide to machine learning!!!: master the concepts, one full-color picture at a time, from the basics all the way to neural networks. BAM!.