MIMICIII Dataset Extraction

Step1: Gain Access to the MIMIC III Dataset

To start the research project on Analyzing Clinical Narratives/Doctor Notes/Discharge Summaries, the extraction of the base dataset MIMIC III from PHYSIONET was the primary task of this sprint. The completion of the CITI "Data or Specimens Only Research" course was mandatory to gain permission to access the dataset [1]. After completion, of course, we got a credentialed user allowance to access the required dataset.

Step2: Accessing and understand the required Tables

While the MIMIC III dataset contains multiple tables, the research on the required topics led us to use two crucial health-related tables on PHYSIONET – ADMISSIONS table and NOTEVENTS table. We downloaded the tables and stored on google drive so they can be accessed using the Google Collaboratory. A detailed understanding of the tables was developed using [2] [3].

Step3: Merge Table

With the aim to work with a singe csv, we merged the files ADMISSIONS and NOTEVENTS into one dataset using the "pandas" library in python. The primary key used for combining the dataset is "HADM_ID," and "SUBJECT_ID." After merging the dataset, the final dimensions of the dataset are (1851344, 29). This merged dataset is then converted into a CSV file for further analytics.

There are two main topics to work on after this initial step: Summarization and Re-Admission Prediction.

Dataset Description

Both the datasets - ADMISSIONS and NOTEVENTS are related to the hospital database. The ADMISSIONS dataset is used to define the patient's hospital admission.

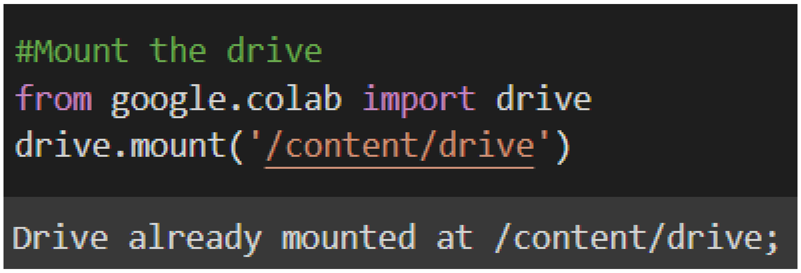
The NOTEVENTS dataset consists of all the notes for patients, and it can be linked to the ADMISSIONS dataset using primary keys, "HADM_ID" and "SUBJECT_ID."

For future research purposes, only important columns such as SUBJECT_ID, HADM_ID, CATEGORY, TEXT, ADMITTIME, DISCHTIME, DEATHTIME, and ADMISSION_TYPE are considered in the merged output dataset.

Code Explanation

The screenshots are attached in this section.

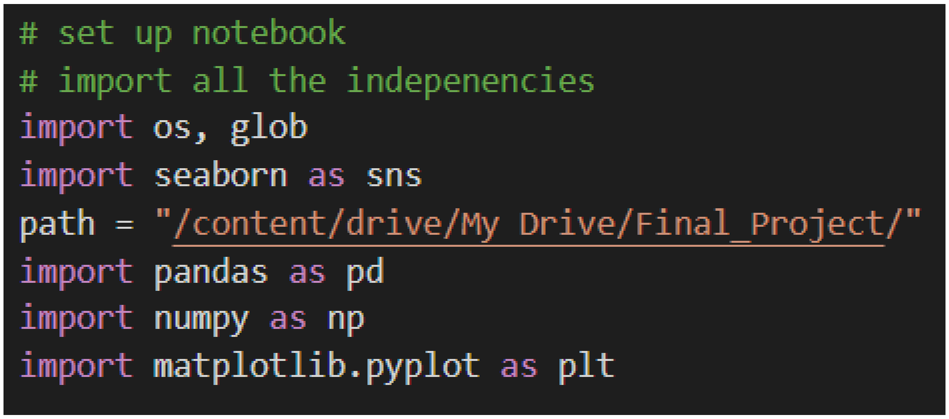
- Mount the drive



```
#Mount the drive
from google.colab import drive
drive.mount('/content/drive')
```

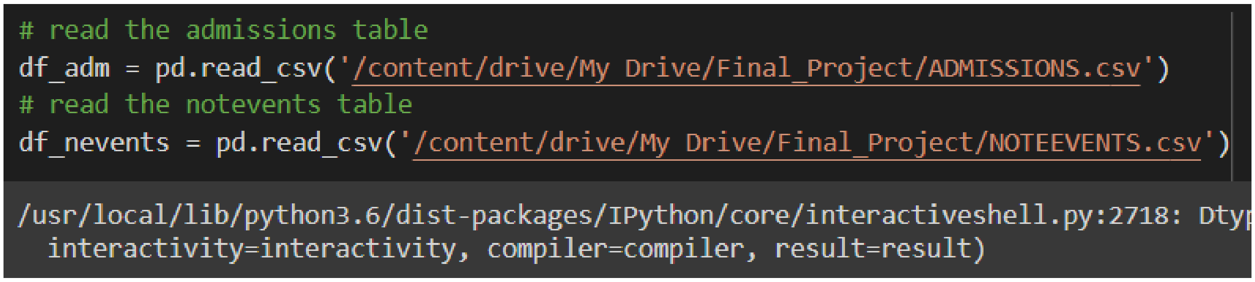
```
Drive already mounted at /content/drive;
```

- Import packages



```
# set up notebook
# import all the indepenencies
import os, glob
import seaborn as sns
path = "/content/drive/My Drive/Final_Project/"
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

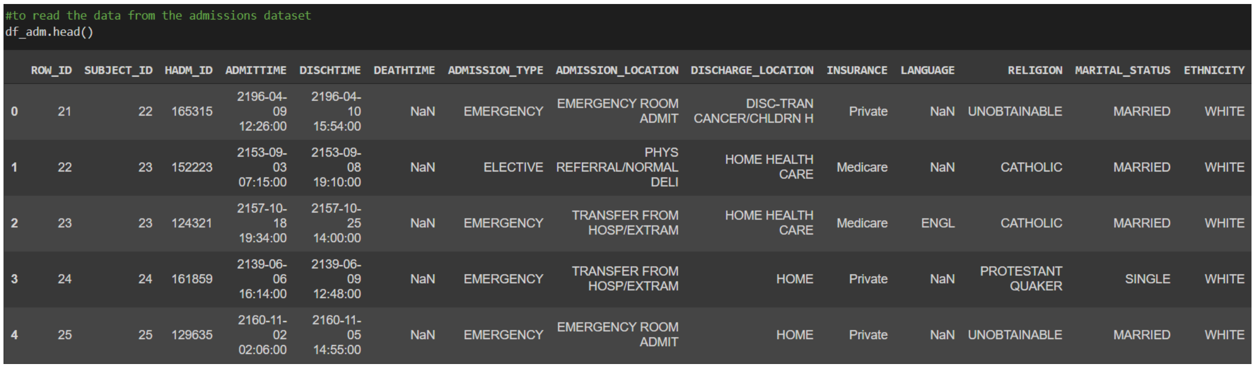
- Read the CSV



```
# read the admissions table
df_adm = pd.read_csv('/content/drive/My Drive/Final_Project/ADMISSIONS.csv')
# read the notevents table
df_nevents = pd.read_csv('/content/drive/My Drive/Final_Project/NOTEEVENTS.csv')
```

```
/usr/local/lib/python3.6/dist-packages/IPython/core/interactiveshell.py:2718: Dtyp
  interactivity=interactivity, compiler=compiler, result=result)
```

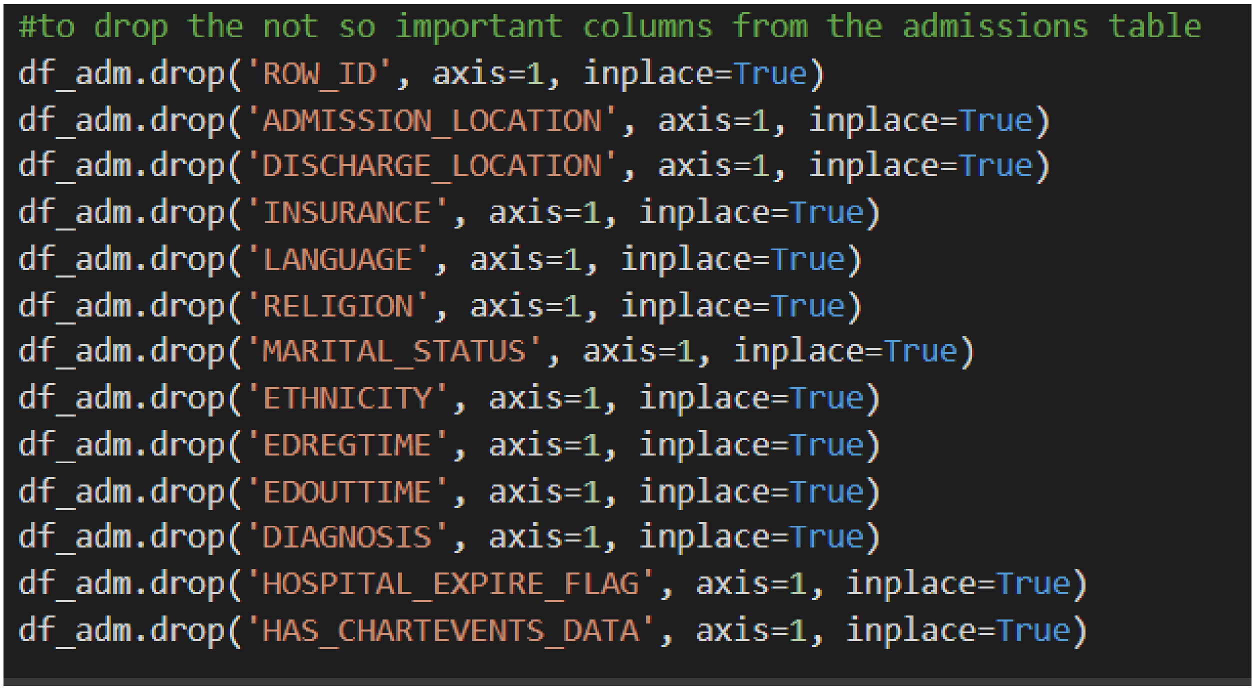
- Admission Tables Top 5 rows



```
#to read the data from the admissions dataset
df_adm.head()
```

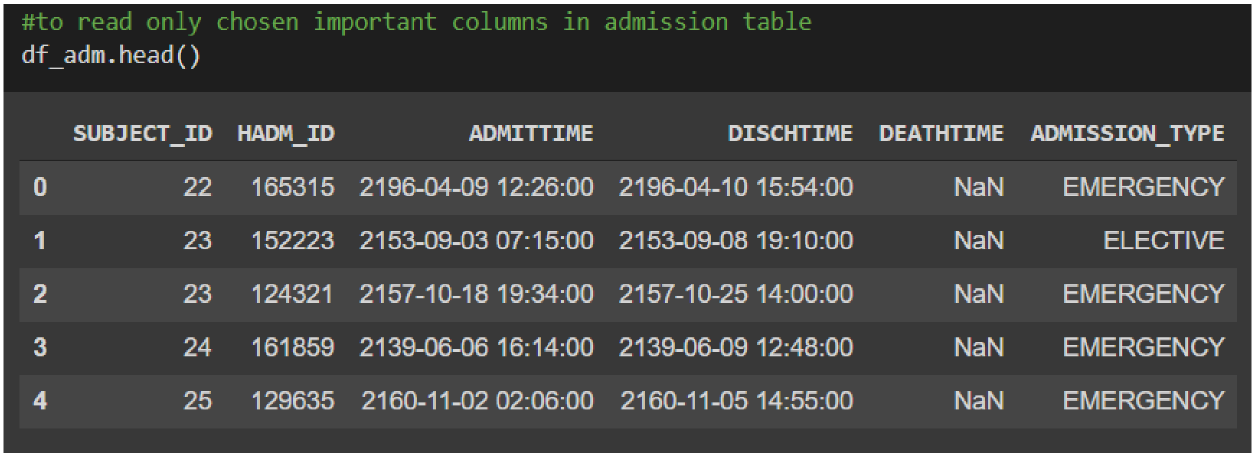
	ROW_ID	SUBJECT_ID	HADM_ID	ADMITTIME	DISCHTIME	DEATHTIME	ADMISSION_TYPE	ADMISSION_LOCATION	DISCHARGE_LOCATION	INSURANCE	LANGUAGE	RELIGION	MARITAL_STATUS	ETHNICITY
0	21	22	165315	2196-04-09 12:26:00	2196-04-10 15:54:00	NaN	EMERGENCY	EMERGENCY ROOM ADMIT	DISC-TRAN CANCER/CHLDRN H	Private	NaN	UNOBTAINABLE	MARRIED	WHITE
1	22	23	152223	2153-09-03 07:15:00	2153-09-08 19:10:00	NaN	ELECTIVE	PHYS REFERRAL/NORMAL DELI	HOME HEALTH CARE	Medicare	NaN	CATHOLIC	MARRIED	WHITE
2	23	23	124321	2157-10-18 19:34:00	2157-10-25 14:00:00	NaN	EMERGENCY	TRANSFER FROM HOSP/EXTRAM	HOME HEALTH CARE	Medicare	ENGL	CATHOLIC	MARRIED	WHITE
3	24	24	161859	2139-06-06 16:14:00	2139-06-09 12:48:00	NaN	EMERGENCY	TRANSFER FROM HOSP/EXTRAM	HOME	Private	NaN	PROTESTANT QUAKER	SINGLE	WHITE
4	25	25	129635	2160-11-02 02:06:00	2160-11-05 14:55:00	NaN	EMERGENCY	EMERGENCY ROOM ADMIT	HOME	Private	NaN	UNOBTAINABLE	MARRIED	WHITE

- Drop Non essential columns from Admission Table



```
#to drop the not so important columns from the admissions table
df_adm.drop('ROW_ID', axis=1, inplace=True)
df_adm.drop('ADMISSION_LOCATION', axis=1, inplace=True)
df_adm.drop('DISCHARGE_LOCATION', axis=1, inplace=True)
df_adm.drop('INSURANCE', axis=1, inplace=True)
df_adm.drop('LANGUAGE', axis=1, inplace=True)
df_adm.drop('RELIGION', axis=1, inplace=True)
df_adm.drop('MARITAL_STATUS', axis=1, inplace=True)
df_adm.drop('ETHNICITY', axis=1, inplace=True)
df_adm.drop('EDREGTIME', axis=1, inplace=True)
df_adm.drop('EDOUTTIME', axis=1, inplace=True)
df_adm.drop('DIAGNOSIS', axis=1, inplace=True)
df_adm.drop('HOSPITAL_EXPIRE_FLAG', axis=1, inplace=True)
df_adm.drop('HAS_CHARTEVENTS_DATA', axis=1, inplace=True)
```

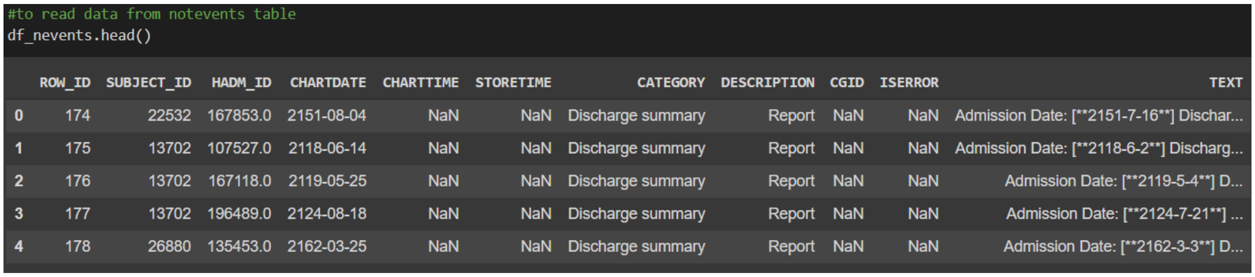
- Admission Tables top 5 row after drop



```
#to read only chosen important columns in admission table
df_adm.head()
```

	SUBJECT_ID	HADM_ID	ADMITTIME	DISCHTIME	DEATHTIME	ADMISSION_TYPE
0	22	165315	2196-04-09 12:26:00	2196-04-10 15:54:00	NaN	EMERGENCY
1	23	152223	2153-09-03 07:15:00	2153-09-08 19:10:00	NaN	ELECTIVE
2	23	124321	2157-10-18 19:34:00	2157-10-25 14:00:00	NaN	EMERGENCY
3	24	161859	2139-06-06 16:14:00	2139-06-09 12:48:00	NaN	EMERGENCY
4	25	129635	2160-11-02 02:06:00	2160-11-05 14:55:00	NaN	EMERGENCY

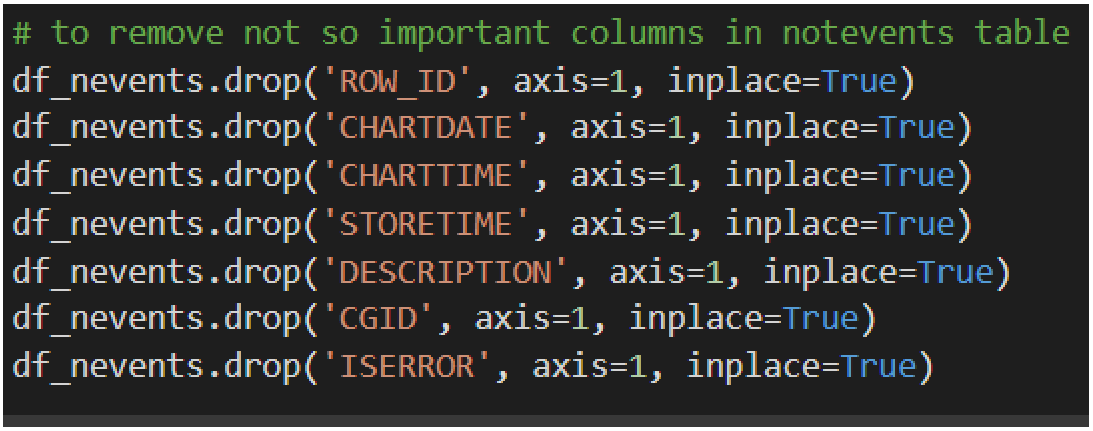
- NoteEvents Table Top 5 row



```
#to read data from notevents table
df_nevents.head()
```

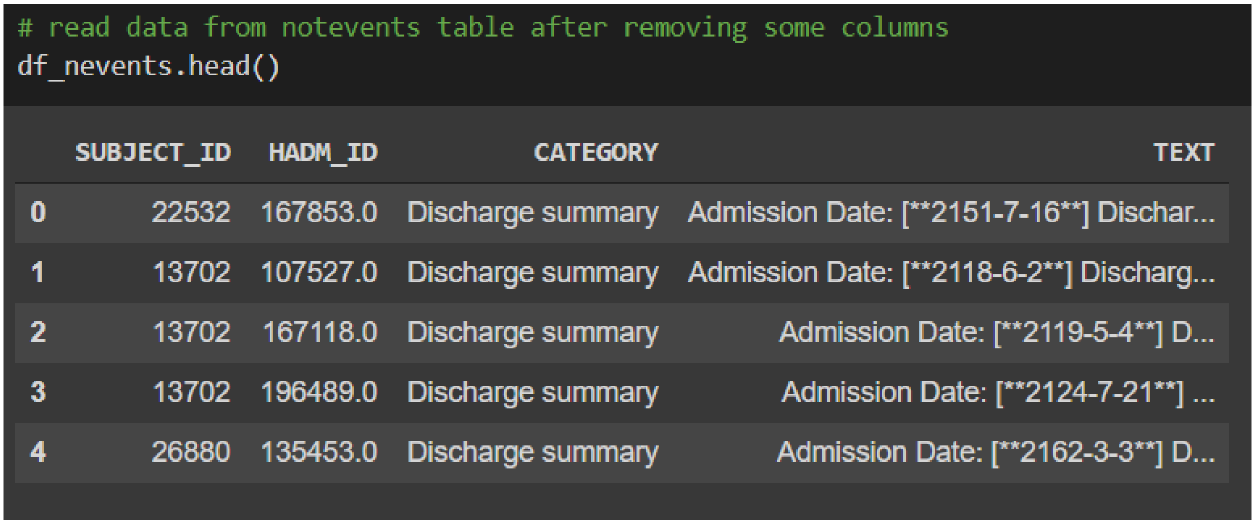
	ROW_ID	SUBJECT_ID	HADM_ID	CHARTDATE	CHARTTIME	STORETIME	CATEGORY	DESCRIPTION	CGID	ISERROR	TEXT
0	174	22532	167853.0	2151-08-04	NaN	NaN	Discharge summary	Report	NaN	NaN	Admission Date: [**2151-7-16**] Dischar...
1	175	13702	107527.0	2118-06-14	NaN	NaN	Discharge summary	Report	NaN	NaN	Admission Date: [**2118-6-2**] Discharg...
2	176	13702	167118.0	2119-05-25	NaN	NaN	Discharge summary	Report	NaN	NaN	Admission Date: [**2119-5-4**] D...
3	177	13702	196489.0	2124-08-18	NaN	NaN	Discharge summary	Report	NaN	NaN	Admission Date: [**2124-7-21**] ...
4	178	26880	135453.0	2162-03-25	NaN	NaN	Discharge summary	Report	NaN	NaN	Admission Date: [**2162-3-3**] D...

- Drop Non-essential columns from NoteEvents Table



```
# to remove not so important columns in notevents table
df_nevents.drop('ROW_ID', axis=1, inplace=True)
df_nevents.drop('CHARTDATE', axis=1, inplace=True)
df_nevents.drop('CHARTTIME', axis=1, inplace=True)
df_nevents.drop('STORETIME', axis=1, inplace=True)
df_nevents.drop('DESCRIPTION', axis=1, inplace=True)
df_nevents.drop('CGID', axis=1, inplace=True)
df_nevents.drop('ISERROR', axis=1, inplace=True)
```

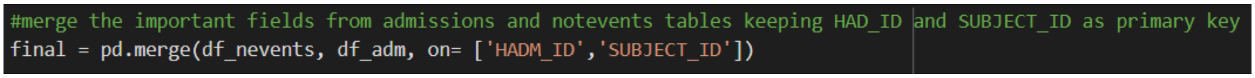
- NoteEvents Tables top 5 row after drop



```
# read data from notevents table after removing some columns
df_nevents.head()
```

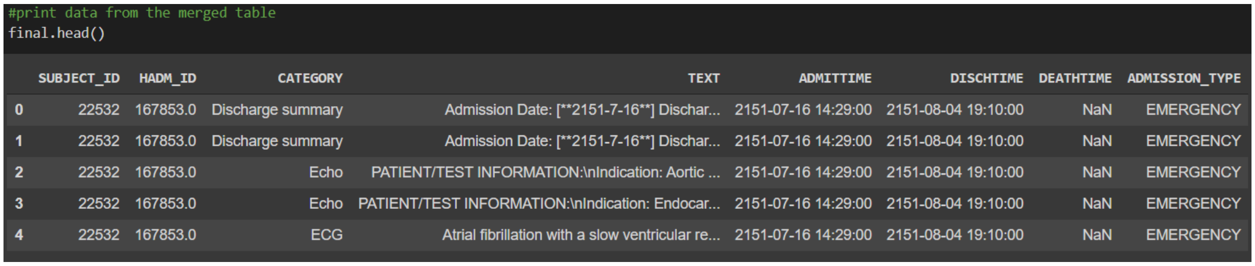
	SUBJECT_ID	HADM_ID	CATEGORY	TEXT
0	22532	167853.0	Discharge summary	Admission Date: [**2151-7-16**] Dischar...
1	13702	107527.0	Discharge summary	Admission Date: [**2118-6-2**] Discharg...
2	13702	167118.0	Discharge summary	Admission Date: [**2119-5-4**] D...
3	13702	196489.0	Discharge summary	Admission Date: [**2124-7-21**] ...
4	26880	135453.0	Discharge summary	Admission Date: [**2162-3-3**] D...

- Merge the Admission and NoteEvent table



```
#merge the important fields from admissions and notevents tables keeping HAD_ID and SUBJECT_ID as primary key
final = pd.merge(df_nevents, df_adm, on= ['HADM_ID','SUBJECT_ID'])
```

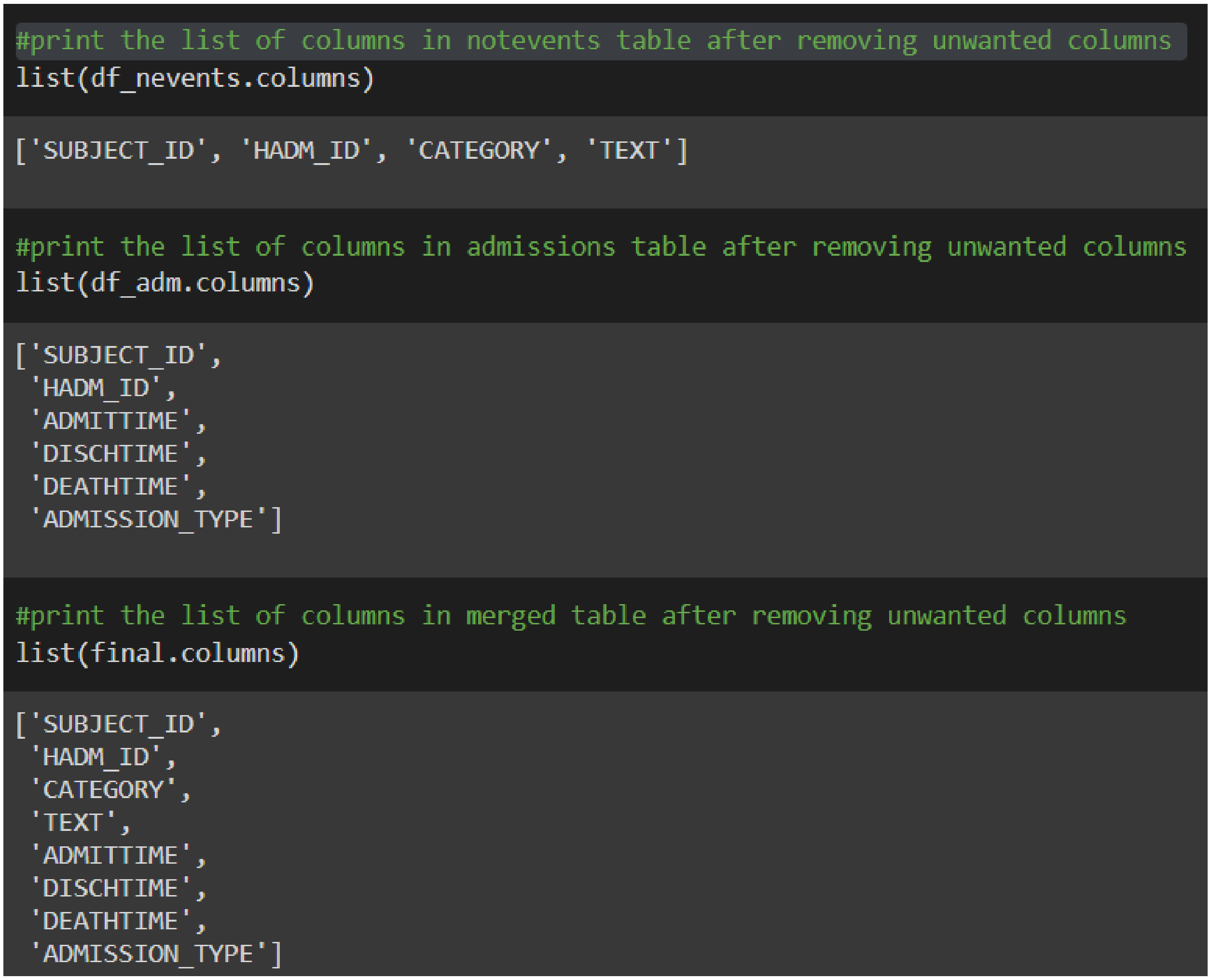
- Merged Table top 5 rows



```
#print data from the merged table
final.head()
```

	SUBJECT_ID	HADM_ID	CATEGORY	TEXT	ADMITTIME	DISCHTIME	DEATHTIME	ADMISSION_TYPE
0	22532	167853.0	Discharge summary	Admission Date: [**2151-7-16**] Dischar...	2151-07-16 14:29:00	2151-08-04 19:10:00	NaN	EMERGENCY
1	22532	167853.0	Discharge summary	Admission Date: [**2151-7-16**] Dischar...	2151-07-16 14:29:00	2151-08-04 19:10:00	NaN	EMERGENCY
2	22532	167853.0	Echo	PATIENT/TEST INFORMATION:\nIndication: Aortic ...	2151-07-16 14:29:00	2151-08-04 19:10:00	NaN	EMERGENCY
3	22532	167853.0	Echo	PATIENT/TEST INFORMATION:\nIndication: Endocar...	2151-07-16 14:29:00	2151-08-04 19:10:00	NaN	EMERGENCY
4	22532	167853.0	ECG	Atrial fibrillation with a slow ventricular re...	2151-07-16 14:29:00	2151-08-04 19:10:00	NaN	EMERGENCY

- List if columns in the Admission, NoteEvents and the Merged Table



```
#print the list of columns in notevents table after removing unwanted columns
list(df_nevents.columns)
```

```
['SUBJECT_ID', 'HADM_ID', 'CATEGORY', 'TEXT']
```

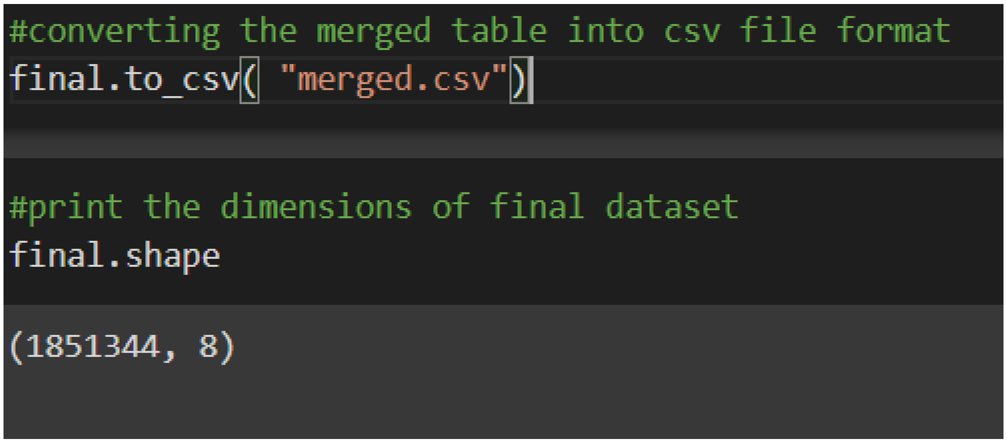
```
#print the list of columns in admissions table after removing unwanted columns
list(df_adm.columns)
```

```
['SUBJECT_ID',
 'HADM_ID',
 'ADMITTIME',
 'DISCHTIME',
 'DEATHTIME',
 'ADMISSION_TYPE']
```

```
#print the list of columns in merged table after removing unwanted columns
list(final.columns)
```

```
['SUBJECT_ID',
 'HADM_ID',
 'CATEGORY',
 'TEXT',
 'ADMITTIME',
 'DISCHTIME',
 'DEATHTIME',
 'ADMISSION_TYPE']
```

- Converting the final merge table to csv and displaying its dimensions



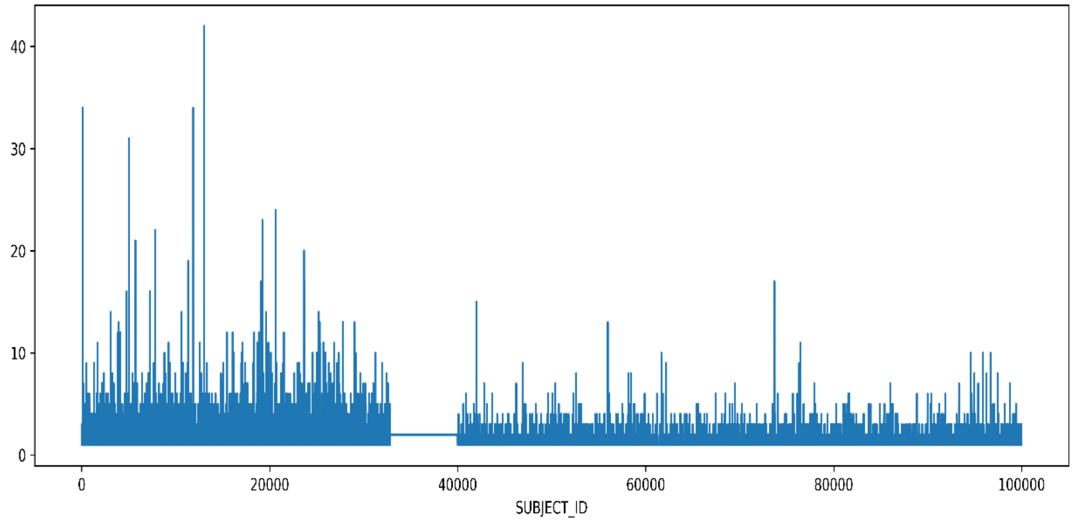
```
#converting the merged table into csv file format
final.to_csv( "merged.csv")

#print the dimensions of final dataset
final.shape
```

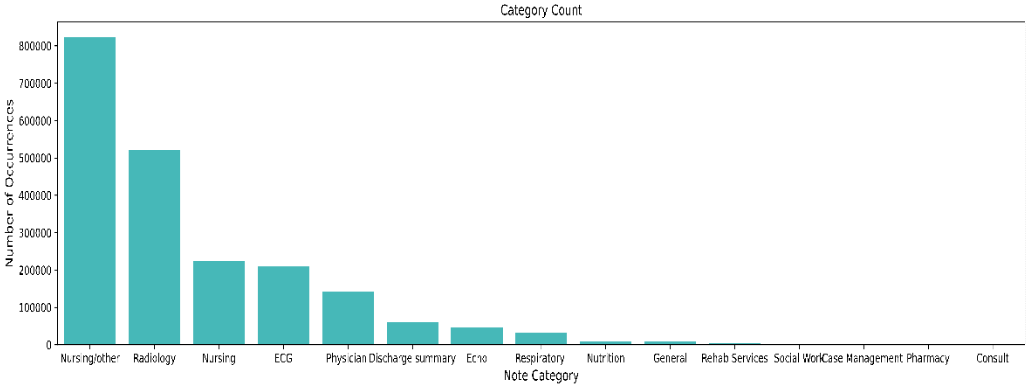
```
(1851344, 8)
```

Visualization

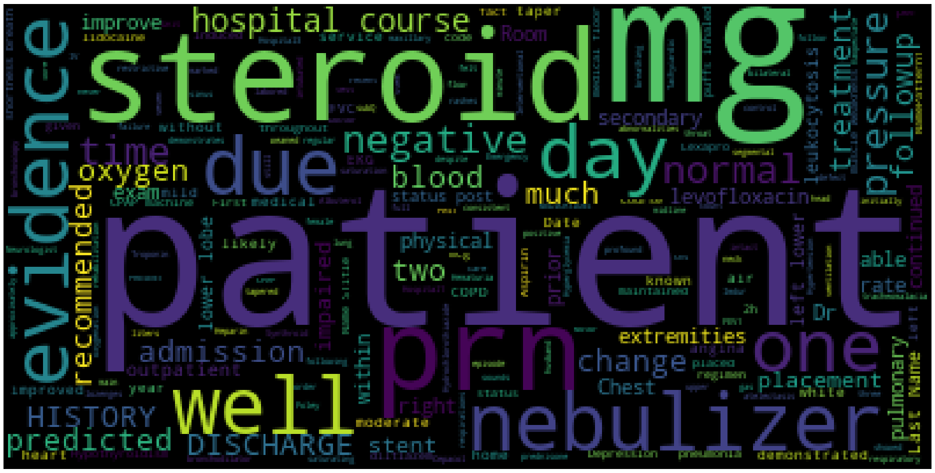
In the Admission table Count the number of Hospital admission for each Subject_ID



In the NoteEvents table the number of notes in each category



Word Cloud for a single note text highlighting the frequent words



References:

[1] https://mimic.physionet.org/gettingstarted/access/

[2] https://mimic.physionet.org/mimictables/admissions/

[3] https://mimic.physionet.org/mimictables/noteevents/