DATA CLEANING, EXPLORATION AND MACHINE LEARNING OF MIMIC DATASET

In [1]:
```
# Importing the libraries required
from pyspark.sql import Row
from pyspark.sql.types import *
from pyspark.sql.functions import sum
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from pyspark.sql.functions import rank, col, unix_timestamp, from_unixtime, t
from pyspark.sql import functions as F
import seaborn as sns
timeFmt = "yyyy-MM-dd"
from pyspark.sql.functions import *
```

In [2]:
```
#Reading MIMIC data
df_mimic_raw = spark.read.csv("mimic_master.csv", header='true', inferSchema=
```

DATA EXPLORATION AND CALCULATION

In [3]:
```
df_mimic_raw.show(2)
```

```
+---------+---+-------+-----+-----+------+-----+------+-------------------+
----------+----------+---------+---------+-------------------+----------+--
---------+
|person_id|age|paramCT|gpiCT|ndcCT|ahfsCT|medCT|gender|         birth_date|
F_T2D_Diag|F_T1D_Diag|F_LD_Diag|F_KD_Diag|         F_CVD_Diag|F_ALZ_Diag|F_
ALZD_Diag|
+---------+---+-------+-----+-----+------+-----+------+-------------------+
----------+----------+---------+---------+-------------------+----------+--
---------+
|      148| 78|   1815|  224|  224|   224|  224|     F|2029-07-11 00:00:00|
null|      null|     null|     null|2107-09-05 14:58:00|      null|       n
ull|
|      463| 62|    121|   13|   13|    13|   13|     F|2136-09-25 00:00:00|
null|      null|     null|     null|               null|      null|       n
ull|
+---------+---+-------+-----+-----+------+-----+------+-------------------+
----------+----------+---------+---------+-------------------+----------+--
---------+
only showing top 2 rows
```

In [4]:
```
df_mimic_raw.cache()
df_mimic_raw.printSchema()
```

```
root
 |-- person_id: integer (nullable = true)
 |-- age: integer (nullable = true)
 |-- paramCT: integer (nullable = true)
 |-- gpiCT: integer (nullable = true)
 |-- ndcCT: integer (nullable = true)
 |-- ahfsCT: integer (nullable = true)
 |-- medCT: integer (nullable = true)
 |-- gender: string (nullable = true)
 |-- birth_date: timestamp (nullable = true)
 |-- F_T2D_Diag: timestamp (nullable = true)
 |-- F_T1D_Diag: timestamp (nullable = true)
 |-- F_LD_Diag: timestamp (nullable = true)
 |-- F_KD_Diag: timestamp (nullable = true)
 |-- F_CVD_Diag: timestamp (nullable = true)
 |-- F_ALZ_Diag: timestamp (nullable = true)
 |-- F_ALZD_Diag: timestamp (nullable = true)
```

In [5]:
```
print('Total population: ', df_mimic_raw.count())
print('-------------------------------------------')
print('Population count by gender')
df_mimic_raw.groupBy('gender').count().show()
```

```
Total population:  52643
-------------------------------------------
Population count by gender
+------+-----+
|gender|count|
+------+-----+
|     F|23184|
|     M|29459|
+------+-----+
```

In [6]:
```
#Certain selected columns as required
df_mimic=df_mimic_raw.select('person_id','age','gender','birth_date','F_T2D_D
df_mimic.show(5)
```

```
+---------+---+------+-------------------+----------+----------+
|person_id|age|gender|         birth_date|F_T2D_Diag|F_ALZ_Diag|
+---------+---+------+-------------------+----------+----------+
|      148| 78|     F|2029-07-11 00:00:00|      null|      null|
|      463| 62|     F|2136-09-25 00:00:00|      null|      null|
|      471| 75|     F|2046-08-30 00:00:00|      null|      null|
|      833|  0|     M|2137-05-23 00:00:00|      null|      null|
|     1088| 68|     M|2102-03-05 00:00:00|      null|      null|
+---------+---+------+-------------------+----------+----------+
only showing top 5 rows
```

In [7]:
```
print('Total population: ', df_mimic.count())
print('-------------------------------------------')
print('Population count by gender')
df_mimic.groupBy('gender').count().show()
```

```
Total population:  52643
-------------------------------------------
Population count by gender
+------+-----+
|gender|count|
+------+-----+
|     F|23184|
|     M|29459|
+------+-----+
```

In [8]:
```
print('Average age of total population')
df_mimic.select(mean("age")).show()
```

```
Average age of total population
+-----------------+
|         avg(age)|
+-----------------+
|63.00890906673252|
+-----------------+
```

Which Disease is diagnosed first?

In [9]:
```
# Order of Disease Diagnosis
T2D_OR_AD_FIRST =F.round((F.col("F_T2D_Diag").cast("long") - F.col("F_ALZ_Dia
df_mimic=df_mimic.withColumn("T2D_OR_AD_FIRST",T2D_OR_AD_FIRST)
```

People who have only T2D meaning AD not at all(control data)

In [10]:
```
#people who have only T2D but not AD-----------------control data
df_mimic_control=df_mimic.filter(df_mimic.F_T2D_Diag.isNotNull() & df_mimic.F
df_mimic_control.show(5)
```

```
+---------+---+------+-------------------+-------------------+----------+--
-------------+
|person_id|age|gender|         birth_date|         F_T2D_Diag|F_ALZ_Diag|T2
D_OR_AD_FIRST|
+---------+---+------+-------------------+-------------------+----------+--
-------------+
|     1829| 53|     M|2133-06-23 00:00:00|2187-04-17 14:00:00|      null|
null|
|     4101| 63|     M|2042-10-12 00:00:00|2103-01-22 18:01:00|      null|
null|
|     4101| 60|     M|2042-10-12 00:00:00|2103-01-22 18:01:00|      null|
null|
|     4900| 70|     M|2133-02-23 00:00:00|2193-04-17 08:00:00|      null|
null|
|     4900| 60|     M|2133-02-23 00:00:00|2193-04-17 08:00:00|      null|
null|
+---------+---+------+-------------------+-------------------+----------+--
-------------+
only showing top 5 rows
```

In [11]:
```
print('Total number of people with only T2D diagnosed: ')
df_mimic_control.count()
```

```
Total number of people with only T2D diagnosed:
```

Out[11]: 12016

In [12]:
```
print('Total number of people with only T2D diagnosed by gender')
df_mimic_control.groupby(["gender"]).count().show()
```

```
Total number of people with only T2D diagnosed by gender
+------+-----+
|gender|count|
+------+-----+
|     F| 5125|
|     M| 6891|
+------+-----+
```

People diagonosed with both or either one disease

In [13]:
```
# Disease not null(people diagnosed either one or both diseases)
df_mimic_disease=df_mimic.filter(df_mimic.F_T2D_Diag.isNotNull() & df_mimic.F
df_mimic_disease.show(5)
```

```
+---------+---+------+-------------------+-------------------+-------------
------+---------------+
|person_id|age|gender|         birth_date|         F_T2D_Diag|         F_AL
Z_Diag|T2D_OR_AD_FIRST|
+---------+---+------+-------------------+-------------------+-------------
------+---------------+
|    23706| 65|     F|2134-01-20 00:00:00|2199-12-31 12:25:00|2200-06-30 2
3:45:00|         -0.497|
|    23706| 66|     F|2134-01-20 00:00:00|2199-12-31 12:25:00|2200-06-30 2
3:45:00|         -0.497|
|     6597| 72|     M|2028-07-08 00:00:00|2100-09-13 16:17:00|2100-09-13 1
6:17:00|            0.0|
|    64798| 88|     F|2083-09-26 00:00:00|2172-08-05 08:18:00|2172-08-05 0
8:18:00|            0.0|
|    79229| 78|     M|2074-06-03 00:00:00|2153-02-11 22:20:00|2153-02-11 2
2:20:00|            0.0|
+---------+---+------+-------------------+-------------------+-------------
------+---------------+
only showing top 5 rows
```

In [14]:
```
print('Total number of people with both diseases: ')
df_mimic_disease.count()
```

```
Total number of people with both diseases:
```

Out[14]: 168

In [15]:
```
print('Total number of people with both diseases by gender')
df_mimic_disease.groupby(["gender"]).count().show()
```

```
Total number of people with both diseases by gender
+------+-----+
|gender|count|
+------+-----+
|     F|  103|
|     M|   65|
+------+-----+
```

People diagonosed with AD only meaning T2D not at all

In [16]:
```
#people who have only AD but not T2D
df_mimic_AD_only=df_mimic.filter(df_mimic.F_T2D_Diag.isNull() & df_mimic.F_AL
df_mimic_AD_only.show(5)
```

```
+---------+---+------+-------------------+----------+-------------------+--
-------------+
|person_id|age|gender|         birth_date|F_T2D_Diag|         F_ALZ_Diag|T2
D_OR_AD_FIRST|
+---------+---+------+-------------------+----------+-------------------+--
-------------+
|    32592| 72|     M|2064-12-27 00:00:00|      null|2137-08-18 01:06:00|
null|
|    94950|300|     F|1855-12-06 23:15:22|      null|2155-12-07 22:37:00|
null|
|    99454| 79|     F|2100-01-11 00:00:00|      null|2179-11-18 04:34:00|
null|
|    27471|300|     M|1822-08-09 23:15:22|      null|2122-08-10 23:13:00|
null|
|    32622| 78|     F|2076-01-26 00:00:00|      null|2154-05-15 19:14:00|
null|
+---------+---+------+-------------------+----------+-------------------+--
-------------+
only showing top 5 rows
```

In [17]:
```
print('Total number of people with only AD diagnosed: ')
df_mimic_AD_only.count()
```

```
Total number of people with only AD diagnosed:
```

Out[17]: 432

In [18]:
```
print('Total number of people with only AD diagnosed by gender')
df_mimic_AD_only.groupby(["gender"]).count().show()
```

```
Total number of people with only AD diagnosed by gender
+------+-----+
|gender|count|
+------+-----+
|     F|  245|
|     M|  187|
+------+-----+
```

Population calculation based on order of diagnosis

In [19]:
```
df_mimic_disease.show(5)
```

```
+---------+---+------+-------------------+-------------------+-------------
------+---------------+
|person_id|age|gender|         birth_date|         F_T2D_Diag|         F_AL
Z_Diag|T2D_OR_AD_FIRST|
+---------+---+------+-------------------+-------------------+-------------
------+---------------+
|    23706| 65|     F|2134-01-20 00:00:00|2199-12-31 12:25:00|2200-06-30 2
3:45:00|         -0.497|
|    23706| 66|     F|2134-01-20 00:00:00|2199-12-31 12:25:00|2200-06-30 2
3:45:00|         -0.497|
|     6597| 72|     M|2028-07-08 00:00:00|2100-09-13 16:17:00|2100-09-13 1
6:17:00|            0.0|
|    64798| 88|     F|2083-09-26 00:00:00|2172-08-05 08:18:00|2172-08-05 0
8:18:00|            0.0|
|    79229| 78|     M|2074-06-03 00:00:00|2153-02-11 22:20:00|2153-02-11 2
2:20:00|            0.0|
+---------+---+------+-------------------+-------------------+-------------
------+---------------+
only showing top 5 rows
```

People with T2D diagnosed first

In [20]:
```
print('People with T2D diagnosed first')
T2D_First=df_mimic_disease.filter(df_mimic_disease.T2D_OR_AD_FIRST<=0)
T2D_First.count()
```

```
People with T2D diagnosed first
```

Out[20]:
```
164
```

In [21]:
```
print('People with T2D diagnosed first by gender')
T2D_First.groupby('gender').count().show()
```

```
People with T2D diagnosed first by gender
+------+-----+
|gender|count|
+------+-----+
|     F|  103|
|     M|   61|
+------+-----+
```

People with both T2D and AD diagnosed at the same time

In [22]:
```
print('People with both T2D and AD diagnosed at the same time')
T2D_AD=df_mimic_disease.filter(df_mimic_disease.T2D_OR_AD_FIRST==0)
T2D_AD.count()
```

```
People with both T2D and AD diagnosed at the same time
```

Out[22]:
```
128
```

In [23]:
```
print('People with both T2D and AD diagnosed at the same time by gender')
T2D_AD.groupby('gender').count().show()
```

```
People with both T2D and AD diagnosed at the same time by gender
+------+-----+
|gender|count|
+------+-----+
|     F|   77|
|     M|   51|
+------+-----+
```

People with AD diagnosed first

In [24]:
```
print('People with AD diagnosed first')
AD_First=df_mimic_disease.filter(df_mimic_disease.T2D_OR_AD_FIRST>0)
AD_First.count()
```

```
People with AD diagnosed first
```

Out[24]:
```
4
```

In [25]:
```
print('People with AD diagnosed first by gender')
AD_First.groupby('gender').count().show()
```

```
People with AD diagnosed first by gender
+------+-----+
|gender|count|
+------+-----+
|     M|    4|
+------+-----+
```

In [26]:
```
df_mimic.show(5)
```

```
+---------+---+------+-------------------+----------+----------+-----------
----+
|person_id|age|gender|         birth_date|F_T2D_Diag|F_ALZ_Diag|T2D_OR_AD_F
IRST|
+---------+---+------+-------------------+----------+----------+-----------
----+
|      148| 78|     F|2029-07-11 00:00:00|      null|      null|
null|
|      463| 62|     F|2136-09-25 00:00:00|      null|      null|
null|
|      471| 75|     F|2046-08-30 00:00:00|      null|      null|
null|
|      833|  0|     M|2137-05-23 00:00:00|      null|      null|
null|
|     1088| 68|     M|2102-03-05 00:00:00|      null|      null|
null|
+---------+---+------+-------------------+----------+----------+-----------
----+
only showing top 5 rows
```

Calculation of age of diagnosis

In [27]:
```
#Age at which diseases diagnosed
Age_T2D_First =F.round((F.col("F_T2D_Diag").cast("long") - F.col("birth_date"
Age_AD_First =F.round((F.col("F_ALZ_Diag").cast("long") - F.col("birth_date")
```

In [28]:
```
df_mimic=df_mimic.withColumn("Age_T2D_First",Age_T2D_First).withColumn("Age_A
```

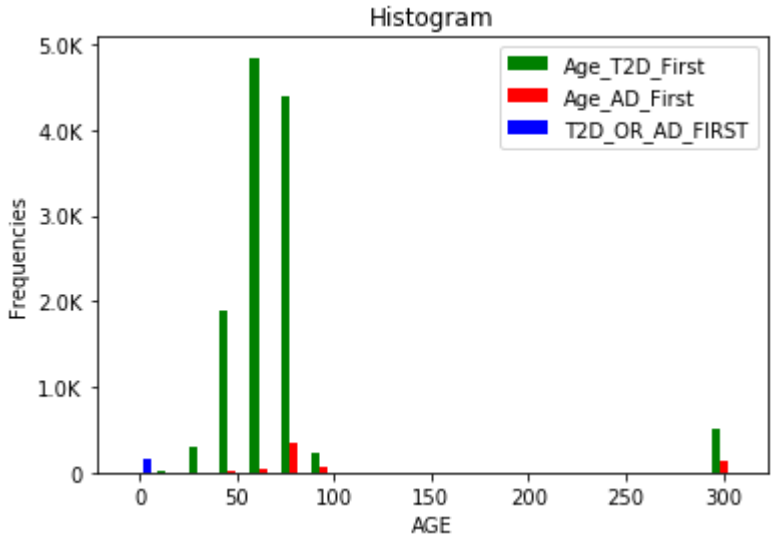
In [29]:
```
df_mimic.show(5)
```

```
+---------+---+------+-------------------+----------+----------+-----------
----+-------------+------------+
|person_id|age|gender|         birth_date|F_T2D_Diag|F_ALZ_Diag|T2D_OR_AD_F
IRST|Age_T2D_First|Age_AD_First|
+---------+---+------+-------------------+----------+----------+-----------
----+-------------+------------+
|      148| 78|     F|2029-07-11 00:00:00|      null|      null|
null|         null|        null|
|      463| 62|     F|2136-09-25 00:00:00|      null|      null|
null|         null|        null|
|      471| 75|     F|2046-08-30 00:00:00|      null|      null|
null|         null|        null|
|      833|  0|     M|2137-05-23 00:00:00|      null|      null|
null|         null|        null|
|     1088| 68|     M|2102-03-05 00:00:00|      null|      null|
null|         null|        null|
+---------+---+------+-------------------+----------+----------+-----------
----+-------------+------------+
only showing top 5 rows
```

In [30]:
```
#histogram  disease data frame plot age T2D first and gender
df_mimic_hist=df_mimic.select('Age_T2D_First','Age_AD_First','T2D_OR_AD_FIRST
```

In [31]:
```
#histogram of age of T2D and age of AD in one graph
from pyspark_dist_explore import hist
fig, ax = plt.subplots()
hist(ax, df_mimic_hist, bins = 20, color=['green','red','blue'])
ax.set_ylabel('Frequencies')
ax.set_xlabel('AGE')
ax.set_title('Histogram')
ax.legend(prop={'size': 10})
```

Out[31]: <matplotlib.legend.Legend at 0x1bee46ffdc8>



In [36]:
```
df_mimic=df_mimic.select('person_id','age','gender','Age_T2D_First','Age_AD_F
```

In [37]:
```
df_mimic.columns
```

Out[37]:
```
['person_id',
 'age',
 'gender',
 'Age_T2D_First',
 'Age_AD_First',
 'T2D_OR_AD_FIRST']
```

In [38]:
```
pd_mimic=df_mimic.toPandas()
```

In [40]:
```
pd_mimic.head(5)
```

Out[40]:

	person_id	age	gender	Age_T2D_First	Age_AD_First	T2D_OR_AD_FIRST
0	148	78	F	NaN	NaN	NaN
1	463	62	F	NaN	NaN	NaN
2	471	75	F	NaN	NaN	NaN
3	833	0	M	NaN	NaN	NaN
4	1088	68	M	NaN	NaN	NaN

Saving MIMIC as mimic_final CSV file

In [41]:
```
pd_mimic.to_csv('final_mimic.csv')
```

In [ ]:
```

```