# Technical Approach Review on Classification of Patients' Readmissions with Diagnosis of Diabetes

Prepared by: Michael Akinosho
Prepared on: January 4th, 2022

# Background

1) Source of data is UCI Machine Learning Repository, URL is: https://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008

2) The data is provided by the Virigina Commonwealth University, on Diabetes related admission from 130-US hospitals for years 1999 through 2008.

3) Target is predicting classifying admissions as re-admit; less than 30, more than 30 or none.

4) This is a classification problem.

5) Data has 50 columns and 6) 100,000 rows of data

7) Dataset has a lot of columns, it is going to test my healthcare background. One of the reasons why I choose this dataset, I have done a lot of work on clinical outcomes.

# Background, contd.

Problem Statement: Classify Encounters into readmitted <30 (<=30), >30 and NO. This project is going to model the classification of patient readmissions into the following three classes:

- Within 30 days (30th day inclusive) "<30" or "(<=30)"
- From 31 days ">30"
- Not readmitted "NO"

All encounters have a diagnosis of diabetes which is based on the clinical analysis of the patient's HbA1c or A1C test.

The American Diabetes Association (ADA) considers this relatively simple blood test a POWERHOUSE!! Some key bullets from their website: ADA/A1C

- The test provides a picture of a person's blood sugar level over two to three months.
- The higher the levels, the greater a person's risk of developing diabetes complications.
- It can identify prediabetes, which raises the risk of diabetes.
- It can be used to diagnose diabetes.
- And, it's used to monitor how well the diabetes treatment is working over time.

The A1C feature in our dataset is very important, but it not enough by itself to help classify encounters into readmitted classes.

Throughout this project, I will continuously revisit this Problem Statement to ensure the model developed is built to help solve it.

# Incorporating CPU and Memory Options

```python
# Mounting Google Drive
# Adding error handling when running Colab notebook locally
# Adding cell notebook when ran locally on a machine with 32 GB of RAM and i9 Processor
# If running, run as standard Colab file please provide path to file in the try block
try:
    from google.colab import drive
    drive.mount('/content/drive')
    filepath = "/content/drive/Othercomputers/My Laptop/diabetes_readmission/"
    msg = "Using Google Colab runtime, connection and resources"
except ModuleNotFoundError:
    filepath = 'C:\\Users\\micha\\Documents\\GitHub\\diabetes_readmission\\'
    msg = "Using local machine runtime, connection and resources"

print(msg)
filename = 'diabetes.csv'
filepathname = filepath + filename
```

```
Using local machine runtime, connection and resources
```

# Initial Libraries, Base Style and Index Defined

[2]

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
import os
```

[3]

```
plt.style.use('seaborn-deep')
```

[5]

```
df = pd.read_csv(filepathname, header=0, index_col = 'encounter_id')
df.head(1)
```

| | patient_nbr | race | gender | age | weight | admission_type_id | discharge_disposition_id | admission_source_id |
|---|---|---|---|---|---|---|---|---|
| **encounter_id** | | | | | | | | |
| **2278392** | 8222157 | Caucasian | Female | [0-10) | ? | 6 | 25 | 1 |

1 rows × 49 columns

# Topical Inspection of Hospitalization Data

```
[6]  1  df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 101766 entries, 2278392 to 443867222
Data columns (total 49 columns):
 #   Column                    Non-Null Count   Dtype
---  ------                    --------------   -----
 0   patient_nbr               101766 non-null  int64
 1   race                      101766 non-null  object
 2   gender                    101766 non-null  object
 3   age                       101766 non-null  object
 4   weight                    101766 non-null  object
 5   admission_type_id         101766 non-null  int64
 6   discharge_disposition_id  101766 non-null  int64
 7   admission_source_id       101766 non-null  int64
 8   time_in_hospital          101766 non-null  int64
 9   payer_code                101766 non-null  object
 10  medical_specialty         101766 non-null  object
 11  num_lab_procedures        101766 non-null  int64
 12  num_procedures            101766 non-null  int64
 13  num_medications           101766 non-null  int64
 14  number_outpatient         101766 non-null  int64
 15  number_emergency          101766 non-null  int64
 16  number_inpatient          101766 non-null  int64
 17  diag_1                    101766 non-null  object
 18  diag_2                    101766 non-null  object
 19  diag_3                    101766 non-null  object
 20  number_diagnoses          101766 non-null  int64
 21  max_glu_serum             101766 non-null  object
 22  A1Cresult                 101766 non-null  object
```

```
 23  metformin                 101766 non-null  object
 24  repaglinide               101766 non-null  object
 25  nateglinide               101766 non-null  object
 26  chlorpropamide            101766 non-null  object
 27  glimepiride               101766 non-null  object
 28  acetohexamide             101766 non-null  object
 29  glipizide                 101766 non-null  object
 30  glyburide                 101766 non-null  object
 31  tolbutamide               101766 non-null  object
 32  pioglitazone              101766 non-null  object
 33  rosiglitazone             101766 non-null  object
 34  acarbose                  101766 non-null  object
 35  miglitol                  101766 non-null  object
 36  troglitazone              101766 non-null  object
 37  tolazamide                101766 non-null  object
 38  examide                   101766 non-null  object
 39  citoglipton               101766 non-null  object
 40  insulin                   101766 non-null  object
 41  glyburide-metformin       101766 non-null  object
 42  glipizide-metformin       101766 non-null  object
 43  glimepiride-pioglitazone  101766 non-null  object
 44  metformin-rosiglitazone   101766 non-null  object
 45  metformin-pioglitazone    101766 non-null  object
 46  change                    101766 non-null  object
 47  diabetesMed               101766 non-null  object
 48  readmitted                101766 non-null  object
dtypes: int64(12), object(37)
memory usage: 38.8+ MB
```

# Helper Functions - Used on Repetitive Tasks

```python
# Helper function that will show normalized value counts for features
def show_column_values():
    for feature in df.columns:
        print("Name of Feature:", feature)
        print(df[feature].value_counts(normalize=True))
        print("\n")
```

```python
# Helper function to generate specific univariate plots
def create_plots(myXaxis,myYaxis,myXlabel,myYlabel,myTitle,myPlot,fsize=7):

    fig, notch_ax = plt.subplots(1, 1, figsize = (fsize,fsize))
    if myPlot == "Boxplot":
        sns.boxplot(ax = notch_ax,x = myXaxis, y = myYaxis, data = df,notch=True)
    elif myPlot == "Barplot":
        df_sub = df.groupby([myXaxis],as_index=False)[[myYaxis]].count()
        sns.barplot(ax = notch_ax, data = df_sub, x = myXaxis, y = myYaxis,edgecolor='black')
    elif myPlot == "Heatmap":
        sns.heatmap(ax = notch_ax, data=df.corr(),annot=True)
    elif myPlot == "Histplot":
        sns.histplot(data=df, x = myXaxis,bins=10)

    plt.title(myTitle)
    plt.xlabel(myXlabel)
    plt.ylabel(myYlabel)

    # Heatmap does not need to have y axis label formatted
    if myPlot != "Heatmap":
        plt.gca().yaxis.set_major_formatter(plt.matplotlib.ticker.StrMethodFormatter('{x:,.0f}'))
    plt.show()
```

# Future Key Decisions Looming; **Impractical Number of Unique Values Across Columns**

```
show_column_values()

Name of Feature: metformin
No        0.803589
Steady    0.180276
Up        0.010485
Down      0.005650
Name: metformin, dtype: float64


Name of Feature: repaglinide
No        0.984877
Steady    0.013600
Up        0.001081
Down      0.000442
Name: repaglinide, dtype: float64


Name of Feature: nateglinide
No        0.993092
Steady    0.006564
Up        0.000236
Down      0.000108
Name: nateglinide, dtype: float64


Name of Feature: chlorpropamide
No        0.999155
```

# Rationale & Methodology on Dropped Columns

This section attempts to explain the reasons for dropping some of the columns from the initial dataframe. The reasons are:

After exploring the dataframe and reading the [Beata et al article], and these four columns (weight, payer_code, medical_specialty and patient_nbr) are dropped:

1. weight: Over 97% of the records are missing the weight value
2. payer_code: Over 40% of the records are missing the payor code, payers don't make the decision on when to admit
3. medical_specialty: Over 49% of the records are missing the specialty value, specialty is not making the decision to admit. It is based on the clinical findings which are determined from the assessment of the health team
4. patient_nbr: not reliable enough to use and since it is an integer, the model might give higher patient_nbr a higher weight/value

# Rationale & Methodology on Dropped Columns, contd.

5. The following columns are also dropped because they don't support variance explanation:
    - admission_type_id; values such as Emergency, Urgent, Elective, Newborn and Trauma Center won't help in classifying the readmitted target
    - admission_source_id; the 26 unique values won't help in classifying the readmitted target, grouping these values into clusters won't improve the model
6. The following columns are dropped because the numeric values assigned within each feature will not improve the classification model's performance, these features are more related to other diagnoses, which we account for in the number_diagnoses (co-morbidity) feature:
    - num_lab_procedures
    - num_procedures
    - num_medications

# Rationale & Methodology on Dropped Columns, contd.

7. The following columns would make the model unnecessarily complex, the values reported are ICD9 (International Classification of Diseases and Procedures) codes used by hospitals to code diagnoses and procedures, including these columns as features would require significant pre-work to evaluate the ICD9 codes and convert these codes into values which represent some weight in terms of high-risk for readmission, a separate project on its own:
    - diag_1
    - diag_2
    - diag_3
8. All the medications listed as columns are also dropped except insulin.

# Rationale & Methodology on Dropped Columns, contd.

```python
features_included = ['race','gender','age','discharge_disposition_id',
                     'time_in_hospital','number_outpatient',
                     'number_emergency','number_inpatient',
                     'number_diagnoses','max_glu_serum','A1Cresult',
                     'insulin','change','diabetesMed','readmitted']
```

```python
# Resetting dataframe df to columns to be used as features and target
# Columns not in the list features_included will be dropped
df = df[features_included]
```

```python
df.shape # Dropped 44 columns
```

```
(101766, 15)
```

# Rationale & Methodology on Dropped Rows

This section attempts to explain the reasons and approaches to dropping rows with invalid values.

- Based on the Beata et al article, excluding records where the discharge disposition is expired or discharged to hospice, will eliminate bias.
- The approach taken by the authors is correct, it would be nonsensical to build the model with records of patients that have expired.
- One might ask why not have the model predict that these patients should be classified in the "NO" class.
- The problem with the statement above is that some in the "NO" class may neither be expired nor transferred to hospice.
- The model needs to learn well how to clarify the "NO" class without bias from records we know will not be readmitted at all.
- Within discharge disposition NULL (NaN), Not Mapped, Unknown/Invalid are also dropped.

# Rationale & Methodology on Clustering Values

Creating clusters for the following five features, all have integer values with over 20 unique values in each feature.

Clustering and grouping will minimize the impact of outliers.

Changing the values to categorical values would make the model inefficient.

For the discharge_disposition, clusters have been identified based on the description of the value, see discharge_df above.

# Rationale & Methodology on Clustering Values, contd.

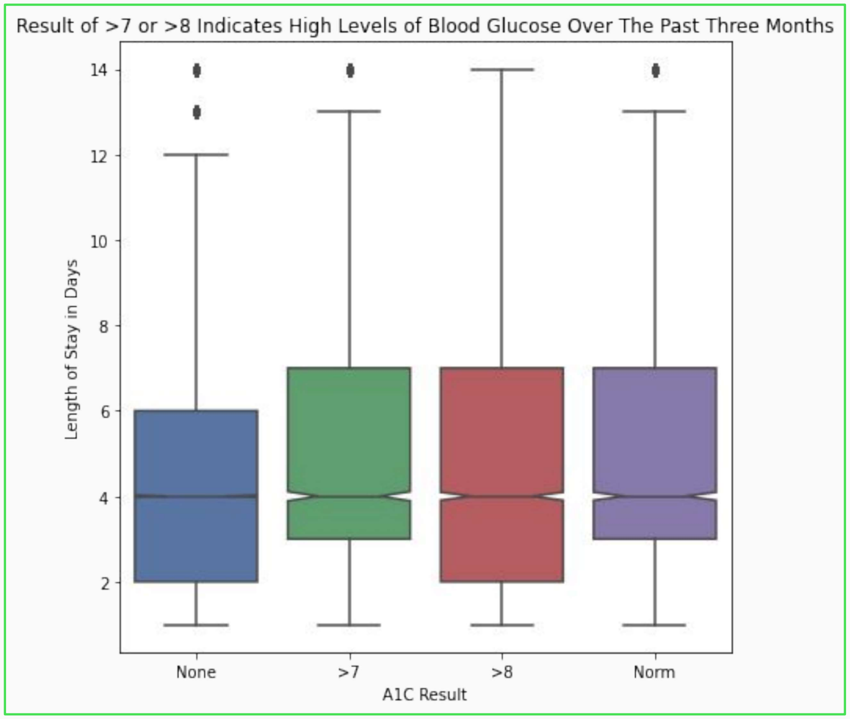
For these four features below:

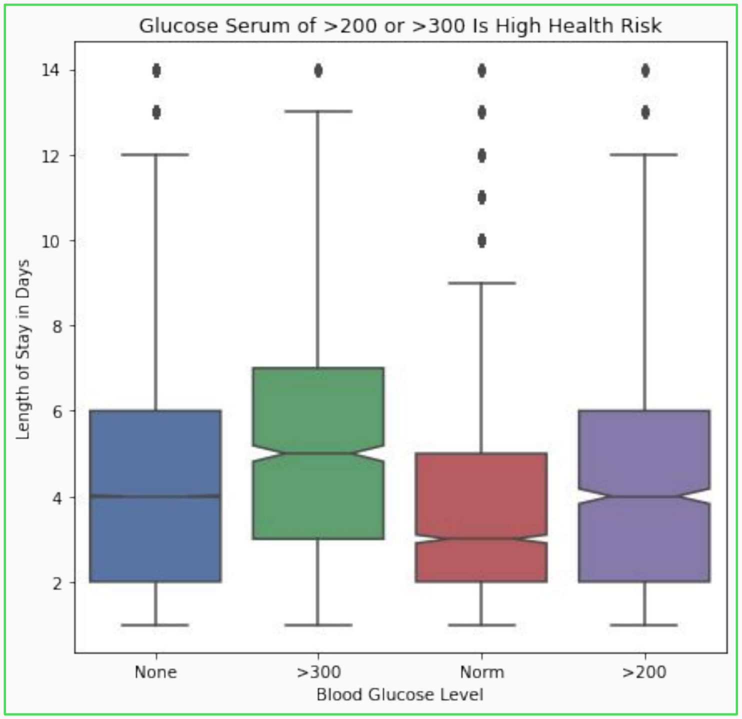
```
* number_outpatient - number of outpatient visits of patient in the year preceding the encounter
* number_emergency - number of emergency visits of patient in the year preceding the encounter
* number_inpatient - number of inpatient visits(admissions) in the year preceding the encounter
* number_diagnoses - number of diagnoses entered into the system
```

Numeric values of 10 or greater than will be grouped under 10, more diagnoses or visits will not necessarily improve the classification performance of the model. These values are already high values for these kinds of features.

# Testing Patients’ A1C and Blood Serum Glucose

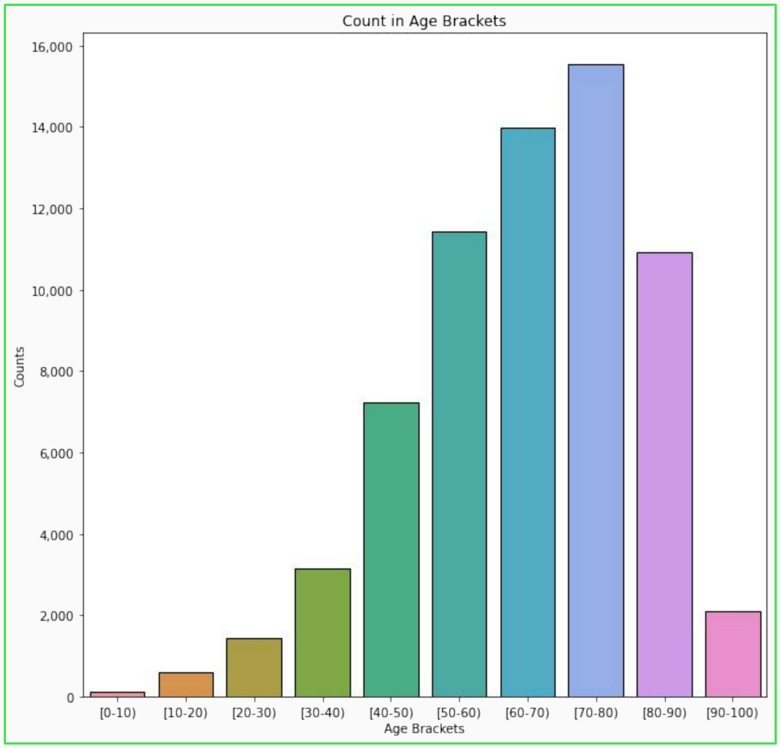
Patients with A1C greater than 7 and serum glucose greater than 200 are showing higher hospital length of stay



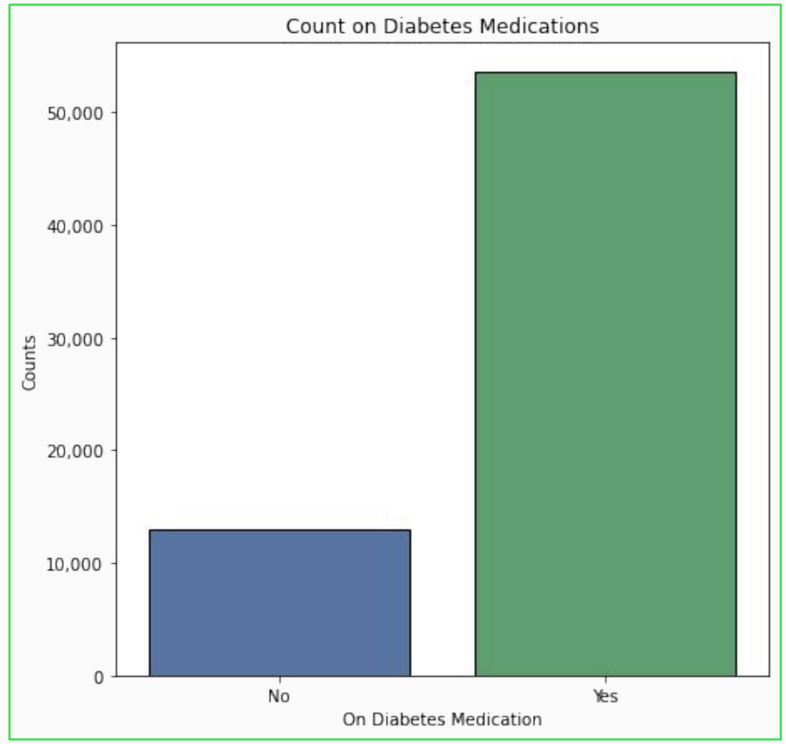



# Age Demographics and Medication Usage; What is the intersection readmits and med usage?

Over 50% of the sample are over the age of 60




About 20% are not taking any diabetes medications



# References and Citations


1. Source of data is UCI Machine Learning Repository, URL is: https://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008
2. List of features and descriptions: https://www.hindawi.com/journals/bmri/2014/781670/tab1/
3. Open Access article: https://www.hindawi.com/journals/bmri/2014/781670/
4. Beata et al article: Beata Strack, Jonathan P. DeShazo, Chris Gennings, Juan L. Olmo, Sebastian Ventura, Krzysztof J. Cios, John N. Clore, "Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records", BioMed Research International, vol. 2014, Article ID 781670, 11 pages, 2014. https://doi.org/10.1155/2014/781670
5. American Diabetes Association: ADA/A1C

# Thank You

Questions and Answers