In [69]:
```
import numpy as np
import pandas as pd

import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
plt.style.use('ggplot')

from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import RandomizedSearchCV
from sklearn.model_selection import cross_val_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import roc_curve, auc
from sklearn.metrics import classification_report

from sklearn.ensemble import RandomForestClassifier
```

In [70]:
```
import os
os.getcwd()
```

Out[70]:
```
'C:\\Users\\itzal\\Untitled Folder 4'
```

In [71]:
```
df = pd.read_csv('diabetic_data.csv')
```

In [72]:
```
df.head()
```

Out[72]:

	encounter_id	patient_nbr	race	gender	age	weight	admission_type_id	discharge_d
0	2278392	8222157	Caucasian	Female	[0-10)	?	6	
1	149190	55629189	Caucasian	Female	[10-20)	?	1	
2	64410	86047875	AfricanAmerican	Female	[20-30)	?	1	
3	500364	82442376	Caucasian	Male	[30-40)	?	1	
4	16680	42519267	Caucasian	Male	[40-50)	?	1	

5 rows × 50 columns

In [73]:
```
df.shape
```

Out[73]:
```
(101766, 50)
```

In [74]:

```
df.readmitted.value_counts()
```

Out[74]:

```
NO      54864
>30     35545
<30     11357
Name: readmitted, dtype: int64
```

In [75]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 101766 entries, 0 to 101765
Data columns (total 50 columns):
 #   Column                    Non-Null Count   Dtype 
---  ------                    --------------   ----- 
 0   encounter_id              101766 non-null  int64 
 1   patient_nbr               101766 non-null  int64 
 2   race                      101766 non-null  object
 3   gender                    101766 non-null  object
 4   age                       101766 non-null  object
 5   weight                    101766 non-null  object
 6   admission_type_id         101766 non-null  int64 
 7   discharge_disposition_id  101766 non-null  int64 
 8   admission_source_id       101766 non-null  int64 
 9   time_in_hospital          101766 non-null  int64 
 10  payer_code                101766 non-null  object
 11  medical_specialty         101766 non-null  object
 12  num_lab_procedures        101766 non-null  int64 
 13  num_procedures            101766 non-null  int64 
 14  num_medications           101766 non-null  int64 
 15  number_outpatient         101766 non-null  int64 
 16  number_emergency          101766 non-null  int64 
 17  number_inpatient          101766 non-null  int64 
 18  diag_1                    101766 non-null  object
 19  diag_2                    101766 non-null  object
 20  diag_3                    101766 non-null  object
 21  number_diagnoses          101766 non-null  int64 
 22  max_glu_serum             101766 non-null  object
 23  A1Cresult                 101766 non-null  object
 24  metformin                 101766 non-null  object
 25  repaglinide               101766 non-null  object
 26  nateglinide               101766 non-null  object
 27  chlorpropamide            101766 non-null  object
 28  glimepiride               101766 non-null  object
 29  acetohexamide             101766 non-null  object
 30  glipizide                 101766 non-null  object
 31  glyburide                 101766 non-null  object
 32  tolbutamide               101766 non-null  object
 33  pioglitazone              101766 non-null  object
 34  rosiglitazone             101766 non-null  object
 35  acarbose                  101766 non-null  object
 36  miglitol                  101766 non-null  object
 37  troglitazone              101766 non-null  object
 38  tolazamide                101766 non-null  object
 39  examide                   101766 non-null  object
 40  citoglipton               101766 non-null  object
 41  insulin                   101766 non-null  object
 42  glyburide-metformin       101766 non-null  object
 43  glipizide-metformin       101766 non-null  object
 44  glimepiride-pioglitazone  101766 non-null  object
 45  metformin-rosiglitazone   101766 non-null  object
 46  metformin-pioglitazone    101766 non-null  object
 47  change                    101766 non-null  object
 48  diabetesMed               101766 non-null  object
 49  readmitted                101766 non-null  object
```

```
dtypes: int64(13), object(37)
memory usage: 38.8+ MB
```

In [76]:

```
#Data Cleaning
```

In [77]:

```
df.shape
```

Out[77]:

```
(101766, 50)
```

In [78]:

```
#There are 101,766 data points in the dataset, some of them are doublicates. We w
df['patient_nbr'].value_counts()
```

Out[78]:

```
88785891     40
43140906     28
1660293      23
88227540     23
23199021     23
             ..
11005362      1
98252496      1
1019673       1
13396320      1
175429310     1
Name: patient_nbr, Length: 71518, dtype: int64
```

In [79]:

```
df = df.drop_duplicates(subset=['patient_nbr'])
```

```
In [80]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 71518 entries, 0 to 101765
Data columns (total 50 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   encounter_id              71518 non-null  int64 
 1   patient_nbr               71518 non-null  int64 
 2   race                      71518 non-null  object
 3   gender                    71518 non-null  object
 4   age                       71518 non-null  object
 5   weight                    71518 non-null  object
 6   admission_type_id         71518 non-null  int64 
 7   discharge_disposition_id  71518 non-null  int64 
 8   admission_source_id       71518 non-null  int64 
 9   time_in_hospital          71518 non-null  int64 
 10  payer_code                71518 non-null  object
 11  medical_specialty         71518 non-null  object
 12  num_lab_procedures        71518 non-null  int64 
 13  num_procedures            71518 non-null  int64 
 14  num_medications           71518 non-null  int64 
 15  number_outpatient         71518 non-null  int64 
 16  number_emergency          71518 non-null  int64 
 17  number_inpatient          71518 non-null  int64 
 18  diag_1                    71518 non-null  object
 19  diag_2                    71518 non-null  object
 20  diag_3                    71518 non-null  object
 21  number_diagnoses          71518 non-null  int64 
 22  max_glu_serum             71518 non-null  object
 23  A1Cresult                 71518 non-null  object
 24  metformin                 71518 non-null  object
 25  repaglinide               71518 non-null  object
 26  nateglinide               71518 non-null  object
 27  chlorpropamide            71518 non-null  object
 28  glimepiride               71518 non-null  object
 29  acetohexamide             71518 non-null  object
 30  glipizide                 71518 non-null  object
 31  glyburide                 71518 non-null  object
 32  tolbutamide               71518 non-null  object
 33  pioglitazone              71518 non-null  object
 34  rosiglitazone             71518 non-null  object
 35  acarbose                  71518 non-null  object
 36  miglitol                  71518 non-null  object
 37  troglitazone              71518 non-null  object
 38  tolazamide                71518 non-null  object
 39  examide                   71518 non-null  object
 40  citoglipton               71518 non-null  object
 41  insulin                   71518 non-null  object
 42  glyburide-metformin       71518 non-null  object
 43  glipizide-metformin       71518 non-null  object
 44  glimepiride-pioglitazone  71518 non-null  object
 45  metformin-rosiglitazone   71518 non-null  object
 46  metformin-pioglitazone    71518 non-null  object
 47  change                    71518 non-null  object
 48  diabetesMed               71518 non-null  object
 49  readmitted                71518 non-null  object
```

```
dtypes: int64(13), object(37)
memory usage: 27.8+ MB
```

In [81]:
```
features_drop_list = ['encounter_id', 'patient_nbr', 'weight', 'payer_code', 'mec
```

In [82]:
```
df.drop(features_drop_list, axis=1,inplace=True)
```

In [83]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 71518 entries, 0 to 101765
Data columns (total 29 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   race                      71518 non-null  object
 1   gender                    71518 non-null  object
 2   age                       71518 non-null  object
 3   admission_type_id         71518 non-null  int64 
 4   discharge_disposition_id  71518 non-null  int64 
 5   admission_source_id       71518 non-null  int64 
 6   time_in_hospital          71518 non-null  int64 
 7   num_lab_procedures        71518 non-null  int64 
 8   num_procedures            71518 non-null  int64 
 9   num_medications           71518 non-null  int64 
 10  number_outpatient         71518 non-null  int64 
 11  number_emergency          71518 non-null  int64 
 12  number_inpatient          71518 non-null  int64 
 13  diag_1                    71518 non-null  object
 14  diag_2                    71518 non-null  object
 15  diag_3                    71518 non-null  object
 16  number_diagnoses          71518 non-null  int64 
 17  max_glu_serum             71518 non-null  object
 18  A1Cresult                 71518 non-null  object
 19  metformin                 71518 non-null  object
 20  glimepiride               71518 non-null  object
 21  glipizide                 71518 non-null  object
 22  glyburide                 71518 non-null  object
 23  pioglitazone              71518 non-null  object
 24  rosiglitazone             71518 non-null  object
 25  insulin                   71518 non-null  object
 26  change                    71518 non-null  object
 27  diabetesMed               71518 non-null  object
 28  readmitted                71518 non-null  object
dtypes: int64(11), object(18)
memory usage: 16.4+ MB
```

In [84]:

```
#start by setting all values containing E or V into 0 (as one category)
df.loc[df['diag_1'].str.contains('V',na=False,case=False), 'diag_1'] = 0
df.loc[df['diag_1'].str.contains('E',na=False,case=False), 'diag_1'] = 0
df.loc[df['diag_2'].str.contains('V',na=False,case=False), 'diag_2'] = 0
df.loc[df['diag_2'].str.contains('E',na=False,case=False), 'diag_2'] = 0
df.loc[df['diag_3'].str.contains('V',na=False,case=False), 'diag_3'] = 0
df.loc[df['diag_3'].str.contains('E',na=False,case=False), 'diag_3'] = 0

#setting all missing values into -1
df['diag_1'] = df['diag_1'].replace('?', -1)
df['diag_2'] = df['diag_2'].replace('?', -1)
df['diag_3'] = df['diag_3'].replace('?', -1)

#No all diag values can be converted into numeric values
df['diag_1'] = df['diag_1'].astype(float)
df['diag_2'] = df['diag_2'].astype(float)
df['diag_3'] = df['diag_3'].astype(float)
```

In [85]:

```
#Now we will reduce the number of categories in diag features according to ICD-9
#(Missing values will be grouped as E & V values)
df['diag_1'].loc[(df['diag_1']>=1) & (df['diag_1']< 140)] = 1
df['diag_1'].loc[(df['diag_1']>=140) & (df['diag_1']< 240)] = 2
df['diag_1'].loc[(df['diag_1']>=240) & (df['diag_1']< 280)] = 3
df['diag_1'].loc[(df['diag_1']>=280) & (df['diag_1']< 290)] = 4
df['diag_1'].loc[(df['diag_1']>=290) & (df['diag_1']< 320)] = 5
df['diag_1'].loc[(df['diag_1']>=320) & (df['diag_1']< 390)] = 6
df['diag_1'].loc[(df['diag_1']>=390) & (df['diag_1']< 460)] = 7
df['diag_1'].loc[(df['diag_1']>=460) & (df['diag_1']< 520)] = 8
df['diag_1'].loc[(df['diag_1']>=520) & (df['diag_1']< 580)] = 9
df['diag_1'].loc[(df['diag_1']>=580) & (df['diag_1']< 630)] = 10
df['diag_1'].loc[(df['diag_1']>=630) & (df['diag_1']< 680)] = 11
df['diag_1'].loc[(df['diag_1']>=680) & (df['diag_1']< 710)] = 12
df['diag_1'].loc[(df['diag_1']>=710) & (df['diag_1']< 740)] = 13
df['diag_1'].loc[(df['diag_1']>=740) & (df['diag_1']< 760)] = 14
df['diag_1'].loc[(df['diag_1']>=760) & (df['diag_1']< 780)] = 15
df['diag_1'].loc[(df['diag_1']>=780) & (df['diag_1']< 800)] = 16
df['diag_1'].loc[(df['diag_1']>=800) & (df['diag_1']< 1000)] = 17
df['diag_1'].loc[(df['diag_1']==-1)] = 0


df['diag_2'].loc[(df['diag_2']>=1) & (df['diag_2']< 140)] = 1
df['diag_2'].loc[(df['diag_2']>=140) & (df['diag_2']< 240)] = 2
df['diag_2'].loc[(df['diag_2']>=240) & (df['diag_2']< 280)] = 3
df['diag_2'].loc[(df['diag_2']>=280) & (df['diag_2']< 290)] = 4
df['diag_2'].loc[(df['diag_2']>=290) & (df['diag_2']< 320)] = 5
df['diag_2'].loc[(df['diag_2']>=320) & (df['diag_2']< 390)] = 6
df['diag_2'].loc[(df['diag_2']>=390) & (df['diag_2']< 460)] = 7
df['diag_2'].loc[(df['diag_2']>=460) & (df['diag_2']< 520)] = 8
df['diag_2'].loc[(df['diag_2']>=520) & (df['diag_2']< 580)] = 9
df['diag_2'].loc[(df['diag_2']>=580) & (df['diag_2']< 630)] = 10
df['diag_2'].loc[(df['diag_2']>=630) & (df['diag_2']< 680)] = 11
df['diag_2'].loc[(df['diag_2']>=680) & (df['diag_2']< 710)] = 12
df['diag_2'].loc[(df['diag_2']>=710) & (df['diag_2']< 740)] = 13
df['diag_2'].loc[(df['diag_2']>=740) & (df['diag_2']< 760)] = 14
df['diag_2'].loc[(df['diag_2']>=760) & (df['diag_2']< 780)] = 15
df['diag_2'].loc[(df['diag_2']>=780) & (df['diag_2']< 800)] = 16
df['diag_2'].loc[(df['diag_2']>=800) & (df['diag_2']< 1000)] = 17
df['diag_2'].loc[(df['diag_2']==-1)] = 0


df['diag_3'].loc[(df['diag_3']>=1) & (df['diag_3']< 140)] = 1
df['diag_3'].loc[(df['diag_3']>=140) & (df['diag_3']< 240)] = 2
df['diag_3'].loc[(df['diag_3']>=240) & (df['diag_3']< 280)] = 3
df['diag_3'].loc[(df['diag_3']>=280) & (df['diag_3']< 290)] = 4
df['diag_3'].loc[(df['diag_3']>=290) & (df['diag_3']< 320)] = 5
df['diag_3'].loc[(df['diag_3']>=320) & (df['diag_3']< 390)] = 6
df['diag_3'].loc[(df['diag_3']>=390) & (df['diag_3']< 460)] = 7
df['diag_3'].loc[(df['diag_3']>=460) & (df['diag_3']< 520)] = 8
df['diag_3'].loc[(df['diag_3']>=520) & (df['diag_3']< 580)] = 9
df['diag_3'].loc[(df['diag_3']>=580) & (df['diag_3']< 630)] = 10
df['diag_3'].loc[(df['diag_3']>=630) & (df['diag_3']< 680)] = 11
df['diag_3'].loc[(df['diag_3']>=680) & (df['diag_3']< 710)] = 12
df['diag_3'].loc[(df['diag_3']>=710) & (df['diag_3']< 740)] = 13
df['diag_3'].loc[(df['diag_3']>=740) & (df['diag_3']< 760)] = 14
```

```
df['diag_3'].loc[(df['diag_3']>=760) & (df['diag_3']< 780)] = 15
df['diag_3'].loc[(df['diag_3']>=780) & (df['diag_3']< 800)] = 16
df['diag_3'].loc[(df['diag_3']>=800) & (df['diag_3']< 1000)] = 17
df['diag_3'].loc[(df['diag_3']==-1)] = 0
```

```
C:\Users\itzal\anaconda3\lib\site-packages\pandas\core\indexing.py:1732: Settin
gWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pyd
ata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
opy)
  self._setitem_single_block(indexer, value, name)
```

In [86]:

```
#check the results
df.diag_1.unique()
```

Out[86]:

```
array([ 3., 11.,  1.,  2.,  7.,  8., 17., 12., 13.,  9.,  0., 16., 10.,
        5.,  6.,  4., 14.])
```

In [87]:

```
df['race'] = df['race'].replace('?', 'Other')
df.race.value_counts()
```

Out[87]:

```
Caucasian          53491
AfricanAmerican    12887
Other               3126
Hispanic            1517
Asian                497
Name: race, dtype: int64
```

In [88]:

```
df.gender.value_counts()
```

Out[88]:

```
Female             38025
Male               33490
Unknown/Invalid        3
Name: gender, dtype: int64
```

In [89]:

```
df['gender'] = df['gender'].replace('Unknown/Invalid', 'Female')
df.gender.value_counts()
```

Out[89]:

```
Female    38028
Male      33490
Name: gender, dtype: int64
```

In [90]:

```
df['gender'] = df['gender'].replace('Male', 1)
df['gender'] = df['gender'].replace('Female', 0)
df.gender.value_counts()
```

Out[90]:

```
0    38028
1    33490
Name: gender, dtype: int64
```

In [91]:

```
df.age.value_counts()
```

Out[91]:

```
[70-80)     18210
[60-70)     15960
[50-60)     12466
[80-90)     11589
[40-50)      6878
[30-40)      2699
[90-100)     1900
[20-30)      1127
[10-20)       535
[0-10)        154
Name: age, dtype: int64
```

In [92]:

```
for i in range(0,10):
    df['age'] = df['age'].replace('['+str(10*i)+'-'+str(10*(i+1))+')', i*10+5)
df['age'].value_counts()
```

Out[92]:

```
75    18210
65    15960
55    12466
85    11589
45     6878
35     2699
95     1900
25     1127
15      535
5       154
Name: age, dtype: int64
```

In [93]:

```
df.max_glu_serum.value_counts()
```

Out[93]:

```
None    68062
Norm     1731
>200      969
>300      756
Name: max_glu_serum, dtype: int64
```

In [94]:

```
df['max_glu_serum']=df['max_glu_serum'].replace("None", 0)
df['max_glu_serum']=df['max_glu_serum'].replace("Norm", 1)
df['max_glu_serum']=df['max_glu_serum'].replace(">200", 2)
df['max_glu_serum']=df['max_glu_serum'].replace(">300", 3)
df.max_glu_serum.value_counts()
```

Out[94]:

```
0    68062
1     1731
2      969
3      756
Name: max_glu_serum, dtype: int64
```

In [95]:
```
df.A1Cresult.value_counts()
```

Out[95]:
```
None    58532
>8       6304
Norm     3791
>7       2891
Name: A1Cresult, dtype: int64
```

In [96]:
```
df['A1Cresult']=df['A1Cresult'].replace("None", 0)
df['A1Cresult']=df['A1Cresult'].replace("Norm", 1)
df['A1Cresult']=df['A1Cresult'].replace(">7", 2)
df['A1Cresult']=df['A1Cresult'].replace(">8", 3)
```

In [97]:
```
df.A1Cresult.value_counts()
```

Out[97]:
```
0    58532
3     6304
1     3791
2     2891
Name: A1Cresult, dtype: int64
```

In [98]:
```
drug_list = ['metformin', 'glimepiride', 'glipizide', 'glyburide', 'pioglitazone'
for i in drug_list:
    df[i] = df[i].replace('No', 0)
    df[i] = df[i].replace('Steady', 2)
    df[i] = df[i].replace('Down', 1)
    df[i] = df[i].replace('Up', 3)
```

In [99]:
```
df.insulin.value_counts()
```

Out[99]:
```
0    34921
2    22129
1     7505
3     6963
Name: insulin, dtype: int64
```

In [100]:
```
df.change.value_counts()
```

Out[100]:
```
No    39494
Ch    32024
Name: change, dtype: int64
```

In [101]:

```
df['change']=df['change'].replace('No', 0)
df['change']=df['change'].replace('Ch', 1)
df.change.value_counts()
```

Out[101]:

```
0    39494
1    32024
Name: change, dtype: int64
```

In [102]:

```
df.diabetesMed.value_counts()
```

Out[102]:

```
Yes    54319
No     17199
Name: diabetesMed, dtype: int64
```

In [103]:

```
df['diabetesMed']=df['diabetesMed'].replace('Yes', 1)
df['diabetesMed']=df['diabetesMed'].replace('No', 0)

df.diabetesMed.value_counts()
```

Out[103]:

```
1    54319
0    17199
Name: diabetesMed, dtype: int64
```

In [104]:

```
df.readmitted.value_counts()
```

Out[104]:

```
NO     42985
>30    22240
<30     6293
Name: readmitted, dtype: int64
```

In [105]:

```
df['readmitted']=df['readmitted'].replace('NO', 0)
df['readmitted']=df['readmitted'].replace('>30', 1)
df['readmitted']=df['readmitted'].replace('<30', 2)
df.readmitted.value_counts()
```

Out[105]:

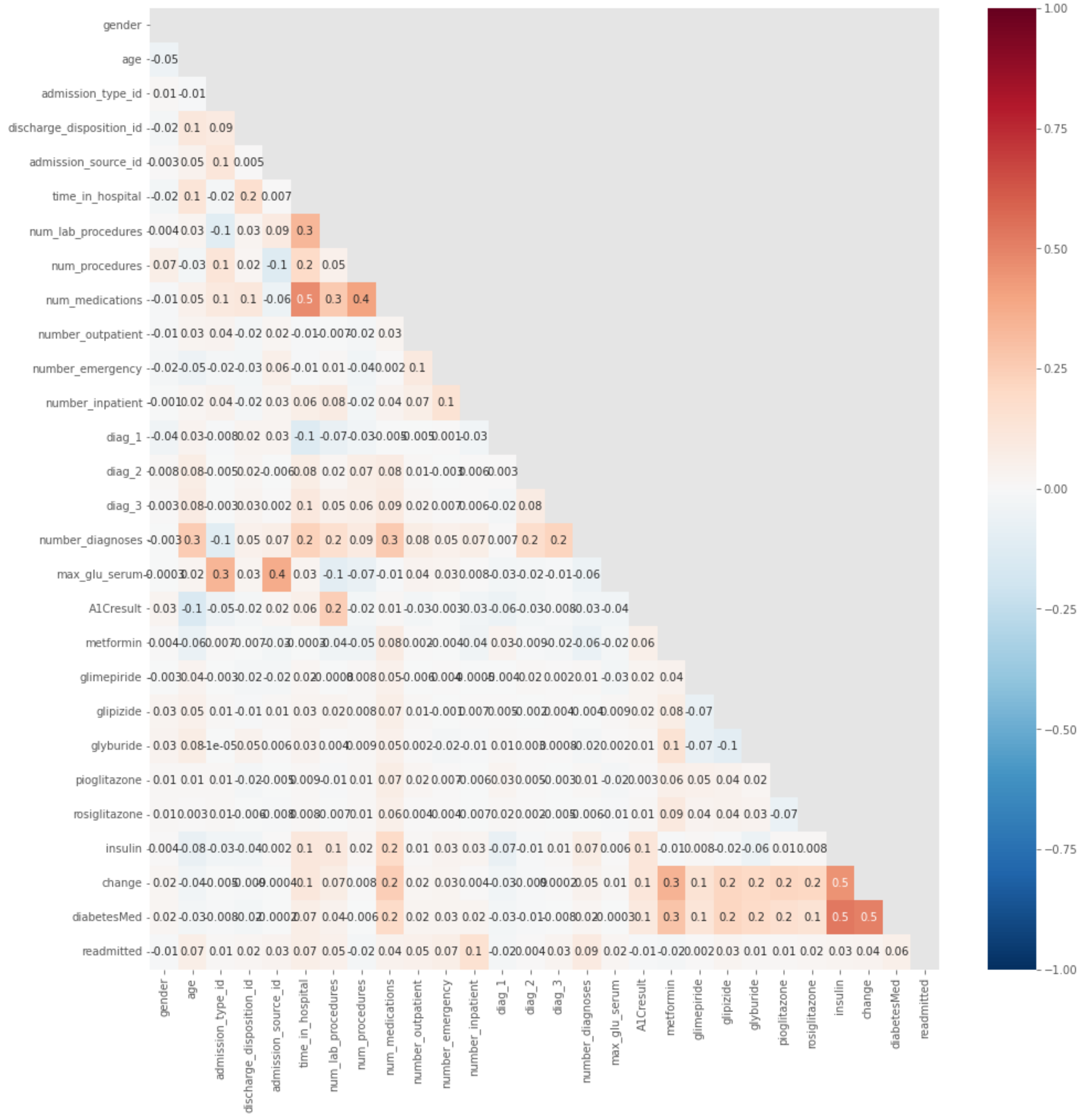
```
0    42985
1    22240
2     6293
Name: readmitted, dtype: int64
```

In [106]:

```
df.info()
```

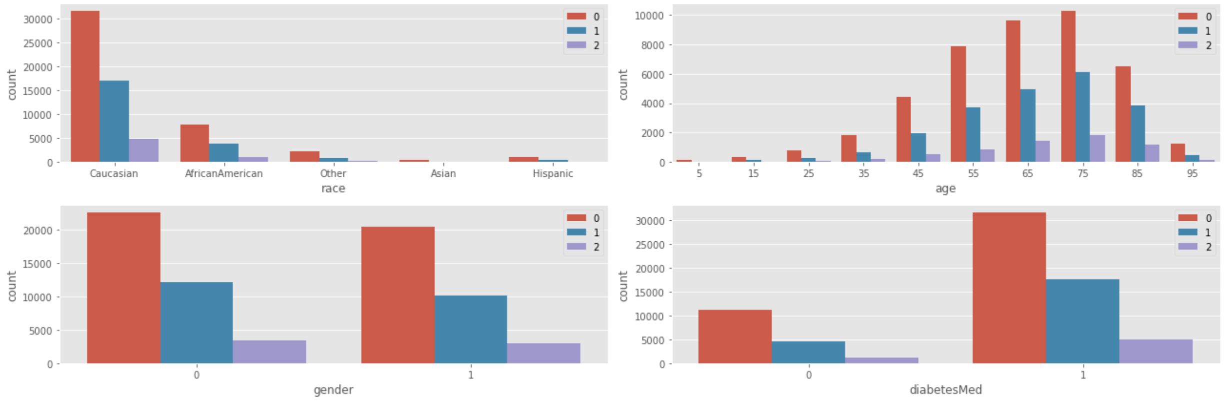
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 71518 entries, 0 to 101765
Data columns (total 29 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   race                      71518 non-null  object 
 1   gender                    71518 non-null  int64  
 2   age                       71518 non-null  int64  
 3   admission_type_id         71518 non-null  int64  
 4   discharge_disposition_id  71518 non-null  int64  
 5   admission_source_id       71518 non-null  int64  
 6   time_in_hospital          71518 non-null  int64  
 7   num_lab_procedures        71518 non-null  int64  
 8   num_procedures            71518 non-null  int64  
 9   num_medications           71518 non-null  int64  
 10  number_outpatient         71518 non-null  int64  
 11  number_emergency          71518 non-null  int64  
 12  number_inpatient          71518 non-null  int64  
 13  diag_1                    71518 non-null  float64
 14  diag_2                    71518 non-null  float64
 15  diag_3                    71518 non-null  float64
 16  number_diagnoses          71518 non-null  int64  
 17  max_glu_serum             71518 non-null  int64  
 18  A1Cresult                 71518 non-null  int64  
 19  metformin                 71518 non-null  int64  
 20  glimepiride               71518 non-null  int64  
 21  glipizide                 71518 non-null  int64  
 22  glyburide                 71518 non-null  int64  
 23  pioglitazone              71518 non-null  int64  
 24  rosiglitazone             71518 non-null  int64  
 25  insulin                   71518 non-null  int64  
 26  change                    71518 non-null  int64  
 27  diabetesMed               71518 non-null  int64  
 28  readmitted                71518 non-null  int64  
dtypes: float64(3), int64(25), object(1)
memory usage: 16.4+ MB
```

In [107]:
```
matrix = np.triu(df.corr())
fig, ax = plt.subplots(figsize=(16,16))
sns.heatmap(df.corr(), annot=True, ax=ax, fmt='.1g', vmin=-1, vmax=1, center= 0,
plt.show()
```

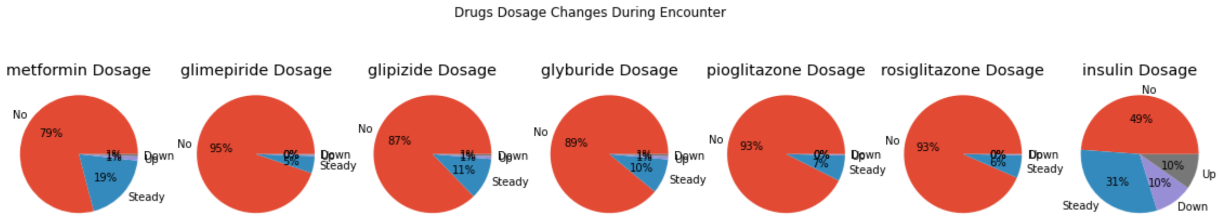


In [108]:

```
fig = plt.figure(figsize=(18, 6))

ax1 = fig.add_subplot(2, 2, 1)
ax2 = fig.add_subplot(2, 2, 2)
ax3 = fig.add_subplot(2, 2, 3)
ax4 = fig.add_subplot(2, 2, 4)

sns.countplot(data=df, x='race', hue='readmitted', ax=ax1)
sns.countplot(data=df, x='age', hue='readmitted', ax=ax2)
sns.countplot(data=df, x='gender', hue='readmitted', ax=ax3)
sns.countplot(data=df, x='diabetesMed', hue='readmitted', ax=ax4)

ax1.legend(loc='upper right')
ax2.legend(loc='upper right')
ax3.legend(loc='upper right')
ax4.legend(loc='upper right')
plt.tight_layout()
plt.show()
```

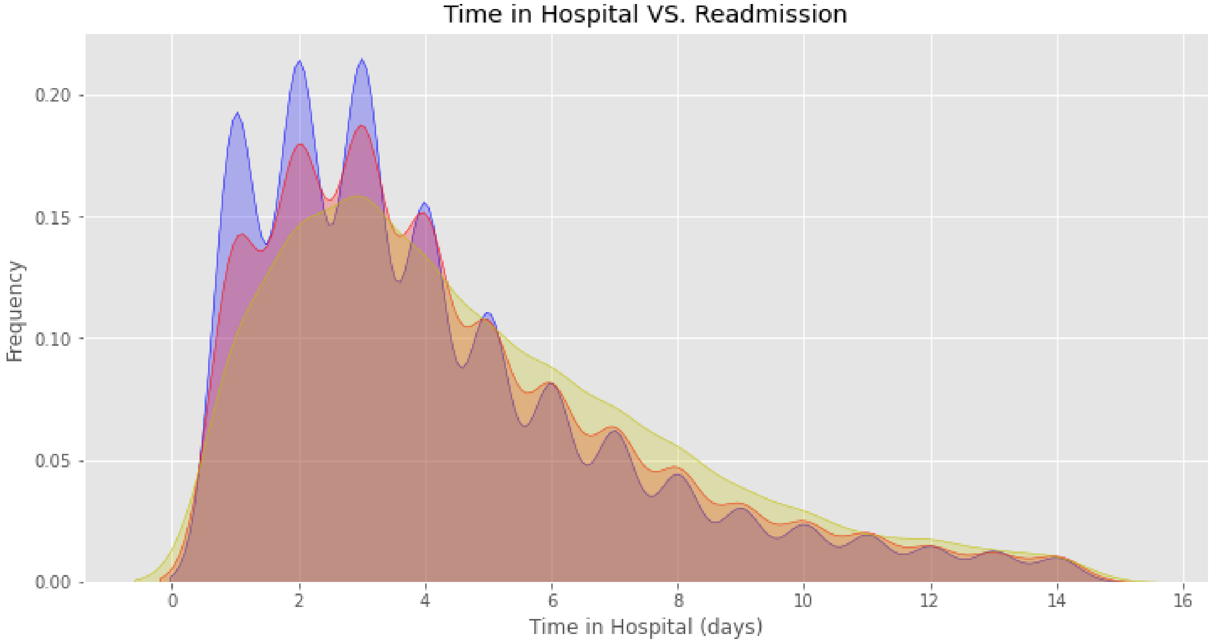


In [109]:
```
fig, ax = plt.subplots(1, 7,figsize=(20, 4), subplot_kw=dict(aspect="equal"))
ax[0].pie(df['metformin'].value_counts(), autopct='%1.0f%%', labels=['No', 'Stea
ax[0].set_title('metformin Dosage')
ax[1].pie(df['glimepiride'].value_counts(), autopct='%1.0f%%', labels=['No', 'Ste
ax[1].set_title('glimepiride Dosage')
ax[2].pie(df['glipizide'].value_counts(), autopct='%1.0f%%', labels=['No', 'Stea
ax[2].set_title('glipizide Dosage')
ax[3].pie(df['glyburide'].value_counts(), autopct='%1.0f%%', labels=['No', 'Stea
ax[3].set_title('glyburide Dosage')
ax[4].pie(df['pioglitazone'].value_counts(), autopct='%1.0f%%', labels=['No', 'St
ax[4].set_title('pioglitazone Dosage')
ax[5].pie(df['rosiglitazone'].value_counts(), autopct='%1.0f%%', labels=['No', 'S
ax[5].set_title('rosiglitazone Dosage')
ax[6].pie(df['insulin'].value_counts(), autopct='%1.0f%%', labels=['No', 'Steady'
ax[6].set_title('insulin Dosage')

fig.suptitle('Drugs Dosage Changes During Encounter')
plt.show()
```



In [110]:
```
fig = plt.figure(figsize=(12,6))
ax=sns.kdeplot(df.loc[(df['readmitted'] == 0),'time_in_hospital'] , color='b',sha
ax=sns.kdeplot(df.loc[(df['readmitted'] == 1),'time_in_hospital'] , color='r',sha
ax=sns.kdeplot(df.loc[(df['readmitted'] == 2),'time_in_hospital'] , color='y',sha
ax.set(xlabel='Time in Hospital (days)', ylabel='Frequency')
plt.title('Time in Hospital VS. Readmission')
```

Out[110]:
```
Text(0.5, 1.0, 'Time in Hospital VS. Readmission')
```



In [111]:

```
df = pd.concat([df,pd.get_dummies(df['race'], prefix='race')], axis=1).drop(['rac
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 71518 entries, 0 to 101765
Data columns (total 33 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   gender                    71518 non-null  int64  
 1   age                       71518 non-null  int64  
 2   admission_type_id         71518 non-null  int64  
 3   discharge_disposition_id  71518 non-null  int64  
 4   admission_source_id       71518 non-null  int64  
 5   time_in_hospital          71518 non-null  int64  
 6   num_lab_procedures        71518 non-null  int64  
 7   num_procedures            71518 non-null  int64  
 8   num_medications           71518 non-null  int64  
 9   number_outpatient         71518 non-null  int64  
 10  number_emergency          71518 non-null  int64  
 11  number_inpatient          71518 non-null  int64  
 12  diag_1                    71518 non-null  float64
 13  diag_2                    71518 non-null  float64
 14  diag_3                    71518 non-null  float64
 15  number_diagnoses          71518 non-null  int64  
 16  max_glu_serum             71518 non-null  int64  
 17  A1Cresult                 71518 non-null  int64  
 18  metformin                 71518 non-null  int64  
 19  glimepiride               71518 non-null  int64  
 20  glipizide                 71518 non-null  int64  
 21  glyburide                 71518 non-null  int64  
 22  pioglitazone              71518 non-null  int64  
 23  rosiglitazone             71518 non-null  int64  
 24  insulin                   71518 non-null  int64  
 25  change                    71518 non-null  int64  
 26  diabetesMed               71518 non-null  int64  
 27  readmitted                71518 non-null  int64  
 28  race_AfricanAmerican      71518 non-null  uint8  
 29  race_Asian                71518 non-null  uint8  
 30  race_Caucasian            71518 non-null  uint8  
 31  race_Hispanic             71518 non-null  uint8  
 32  race_Other                71518 non-null  uint8  
dtypes: float64(3), int64(25), uint8(5)
memory usage: 18.2 MB
```

In [112]:

```
df.shape
```

Out[112]:

```
(71518, 33)
```

In [42]:

```
y = df['readmitted']
X = df.drop(['readmitted'], axis=1)
```

In [113]:
```
y.value_counts()
```

Out[113]:
```
0    42985
1    22240
2     6293
Name: readmitted, dtype: int64
```

In [114]:
```
from imblearn.over_sampling import SMOTE
```

In [115]:
```
smote = SMOTE(sampling_strategy = 'minority')
```

In [117]:
```
x_sm, y_sm = smote.fit_resample(X,y)
```

In [118]:
```
y_sm.value_counts()
```

Out[118]:
```
0    42985
2    42985
1    22240
Name: readmitted, dtype: int64
```

In [120]:
```
#training and testing 80 20 split
```

In [121]:
```
X_train, X_test, y_train, y_test = train_test_split(x_sm,y_sm, test_size=0.2, ran
```

In [127]:
```
y_test.value_counts()
```

Out[127]:
```
0    8670
2    8586
1    4386
Name: readmitted, dtype: int64
```

In [124]:
```
y_train.shape
```

Out[124]:
```
(86568,)
```

In [126]:
```
X_test.shape
```

Out[126]:
```
(21642, 32)
```

In [128]:
```
# Model Building
```

In [129]:
```
rf = RandomForestClassifier()
rf.fit(X_train,y_train)
```

Out[129]:
```
RandomForestClassifier()
```

In [130]:
```
y_pred = rf.predict(X_test)
```

```
In [134]: rf.score(X_test,y_test) #Accuracy on test dataset

Out[134]: 0.7345439423343498
```

```
In [149]: rf.n_estimators

Out[149]: 200
```

```
In [135]: rf.score(X_train,y_train) # Accuracy on train dataset

Out[135]: 0.9999768967747897
```

In [144]:

```
n_estimators = [1,2,4,8,16,32,64,100,200]
train_results = []
test_results = []
oob_score_results = []

for estimator in n_estimators:
    rf = RandomForestClassifier(n_estimators=estimator, oob_score=True)
    rf.fit(X_train,y_train)
    oob_score_results.append(rf.oob_score_)
    train_accuracy = rf.score(X_train,y_train)
    test_accuracy = rf.score(X_test,y_test)
    train_results.append(train_accuracy)
    test_results.append(test_accuracy)
```

```
C:\Users\itzal\anaconda3\lib\site-packages\sklearn\base.py:445: UserWarning: X
does not have valid feature names, but RandomForestClassifier was fitted with f
eature names
  warnings.warn(
C:\Users\itzal\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:550: Use
rWarning: Some inputs do not have OOB scores. This probably means too few trees
were used to compute any reliable OOB estimates.
  warn(
C:\Users\itzal\anaconda3\lib\site-packages\sklearn\base.py:445: UserWarning: X
does not have valid feature names, but RandomForestClassifier was fitted with f
eature names
  warnings.warn(
C:\Users\itzal\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:550: Use
rWarning: Some inputs do not have OOB scores. This probably means too few trees
were used to compute any reliable OOB estimates.
  warn(
C:\Users\itzal\anaconda3\lib\site-packages\sklearn\base.py:445: UserWarning: X
does not have valid feature names, but RandomForestClassifier was fitted with f
eature names
  warnings.warn(
C:\Users\itzal\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:550: Use
rWarning: Some inputs do not have OOB scores. This probably means too few trees
were used to compute any reliable OOB estimates.
  warn(
C:\Users\itzal\anaconda3\lib\site-packages\sklearn\base.py:445: UserWarning: X
does not have valid feature names, but RandomForestClassifier was fitted with f
eature names
  warnings.warn(
C:\Users\itzal\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:550: Use
rWarning: Some inputs do not have OOB scores. This probably means too few trees
were used to compute any reliable OOB estimates.
  warn(
C:\Users\itzal\anaconda3\lib\site-packages\sklearn\base.py:445: UserWarning: X
does not have valid feature names, but RandomForestClassifier was fitted with f
eature names
  warnings.warn(
C:\Users\itzal\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:550: Use
rWarning: Some inputs do not have OOB scores. This probably means too few trees
were used to compute any reliable OOB estimates.
  warn(
C:\Users\itzal\anaconda3\lib\site-packages\sklearn\base.py:445: UserWarning: X
does not have valid feature names, but RandomForestClassifier was fitted with f
```

```
eature names
  warnings.warn(
C:\Users\itzal\anaconda3\lib\site-packages\sklearn\base.py:445: UserWarning: X
does not have valid feature names, but RandomForestClassifier was fitted with f
eature names
  warnings.warn(
C:\Users\itzal\anaconda3\lib\site-packages\sklearn\base.py:445: UserWarning: X
does not have valid feature names, but RandomForestClassifier was fitted with f
eature names
  warnings.warn(
C:\Users\itzal\anaconda3\lib\site-packages\sklearn\base.py:445: UserWarning: X
does not have valid feature names, but RandomForestClassifier was fitted with f
eature names
  warnings.warn(
```

In [145]:

```
test_results
```

Out[145]:

```
[0.5937066814527308,
 0.6272987709084188,
 0.6643101376952223,
 0.6936974401626467,
 0.7111634784215877,
 0.7258109232048794,
 0.7314019037057573,
 0.735329451991498,
 0.7372239164587376]
```

In [147]:

```
train_results
```

Out[147]:

```
[0.8491012845393217,
 0.8570950004620645,
 0.9410636724886794,
 0.9817715553091212,
 0.997008132335274,
 0.9996534516218464,
 0.9999653451621846,
 0.9999768967747897,
 0.9999768967747897]
```

In [148]:

```
oob_score_results
```

Out[148]:

```
[0.46814065243507996,
 0.5224678865169577,
 0.5819817946585343,
 0.6353848997320026,
 0.678483966361704,
 0.7044057850475927,
 0.7206011459199704,
 0.7274859070326217,
 0.7350753165141853]
```

In [136]:

```
# Confusion matrix
```

In [151]:

```
y_predicted2 = rf.predict(X_train) #confusion matrix on the train set
```

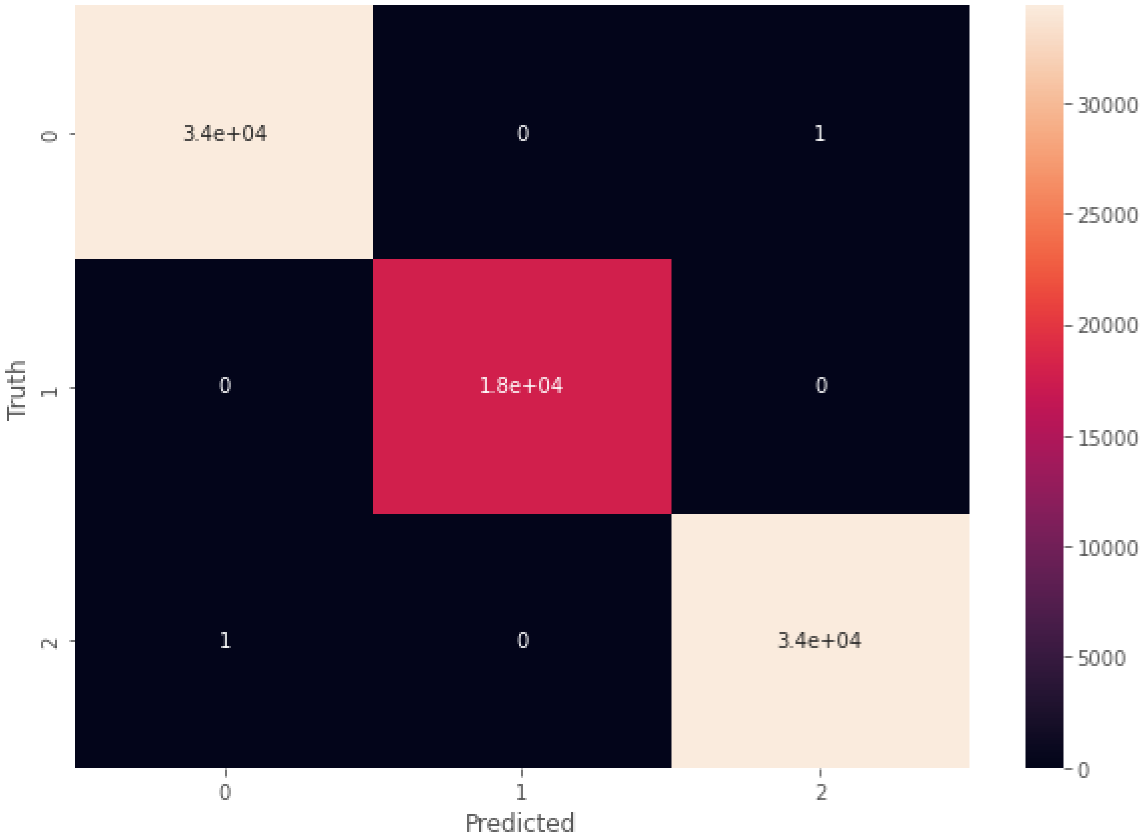
```
In [152]: cm2 = confusion_matrix(y_train,y_predicted2) #confusion matrix on the train set
```

```
In [153]: cm2
```

```
Out[153]: array([[34314,     0,     1],
                 [    0, 17854,     0],
                 [    1,     0, 34398]], dtype=int64)
```

```
In [154]: plt.figure(figsize=(10,7))
          sns.heatmap(cm2,annot=True)
          plt.xlabel('Predicted')
          plt.ylabel('Truth')
```

```
Out[154]: Text(69.0, 0.5, 'Truth')
```



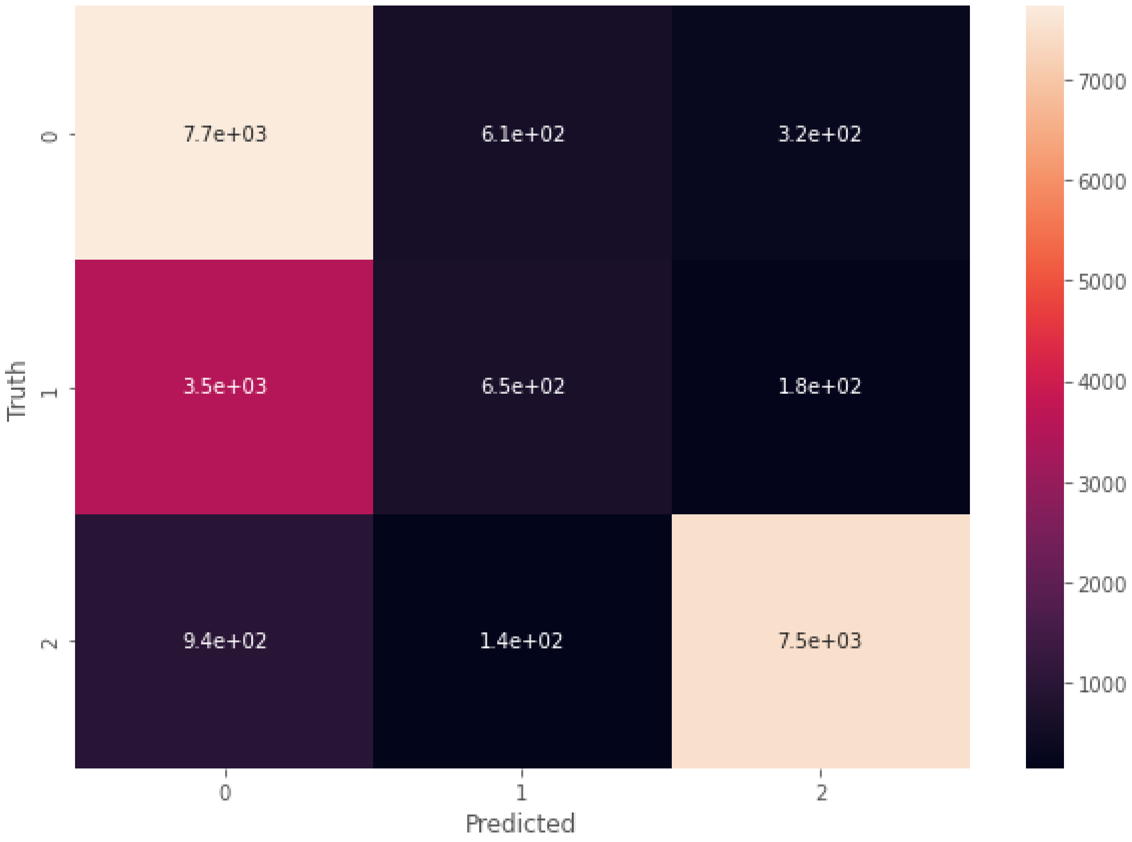
```
In [137]: y_predicted = rf.predict(X_test) # confusion matrix on the test set
```

```
In [138]: cm = confusion_matrix(y_test, y_predicted) # confusion matrix on the test set
          cm
```

```
Out[138]: array([[7740,  612,  318],
                 [3549,  654,  183],
                 [ 943,  140, 7503]], dtype=int64)
```

```
In [141]: plt.figure(figsize=(10,7))
          sns.heatmap(cm,annot=True)
          plt.xlabel('Predicted')
          plt.ylabel('Truth')
```

```
Out[141]: Text(69.0, 0.5, 'Truth')
```

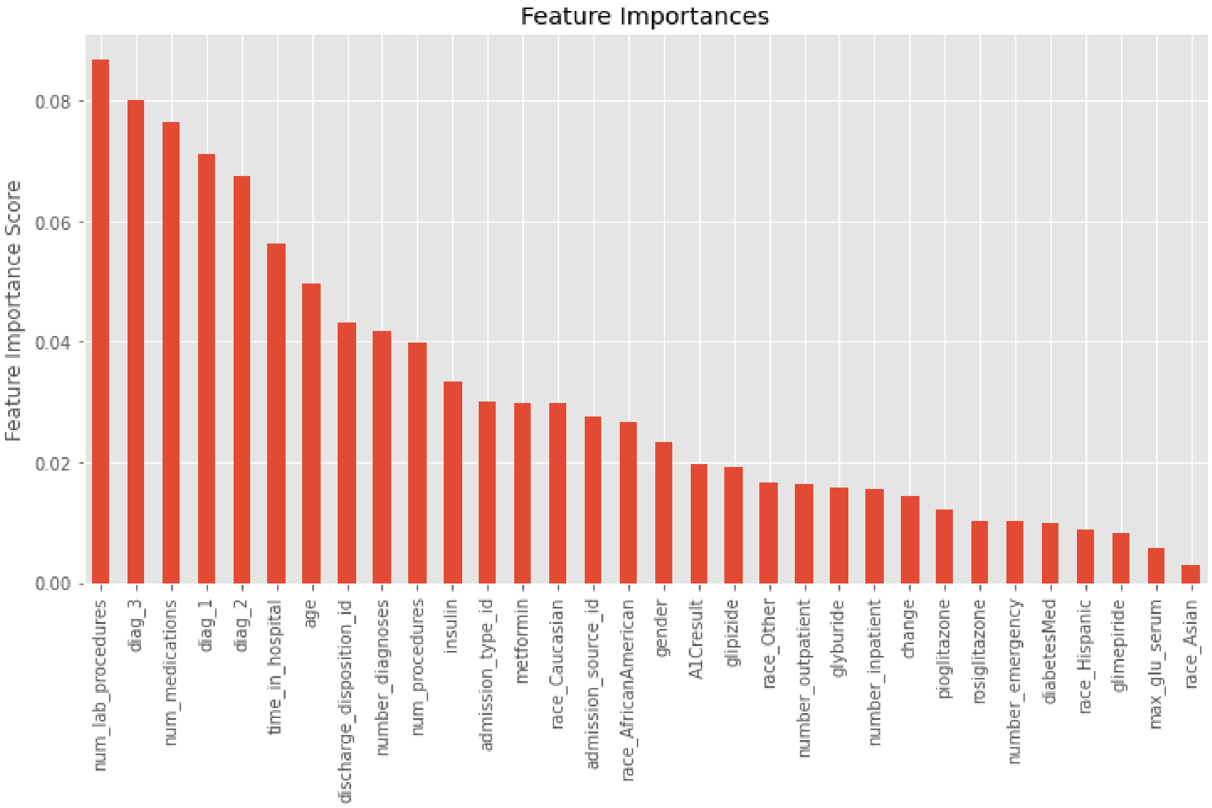


In [155]:

```
#define a list that has all feature names
predictors = [x for x in X_train.columns]

feat_imp = pd.Series(rf.feature_importances_, predictors).sort_values(ascending=F
fig = plt.figure(figsize=(12, 6))
feat_imp.plot(kind='bar', title='Feature Importances')
plt.ylabel('Feature Importance Score')
```

Out[155]:

```
Text(0, 0.5, 'Feature Importance Score')
```



```
In [156]: rf.feature_importances_
```

```
Out[156]: array([0.02338156, 0.04949679, 0.03003841, 0.04312932, 0.02744736,
                 0.05646569, 0.08683474, 0.03985971, 0.07660576, 0.01641532,
                 0.01008906, 0.01558161, 0.07127608, 0.06763529, 0.08009236,
                 0.04170548, 0.00573432, 0.01959074, 0.02988467, 0.00821005,
                 0.01925168, 0.0157632 , 0.01211103, 0.01035439, 0.03342479,
                 0.01439375, 0.01002   , 0.02681102, 0.00308674, 0.02965384,
                 0.00890467, 0.01675056])
```

```
In [ ]:
```