Canadian Hospital Re-admittance Challenge

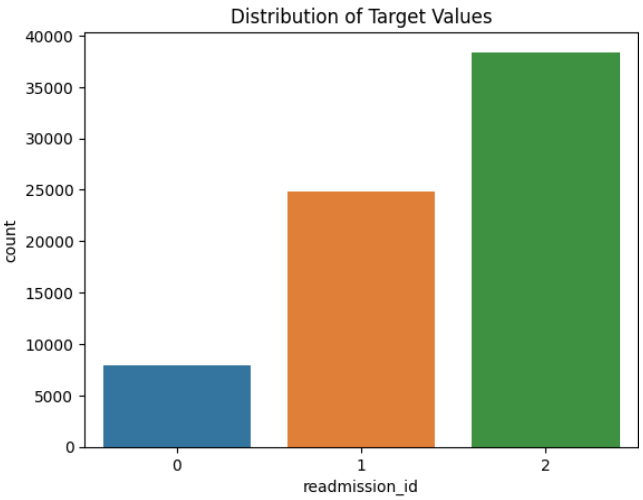
Readmission of a patient generally indicates incorrect diagnosis or prescription. Predicting the readmission can help us prevent this.

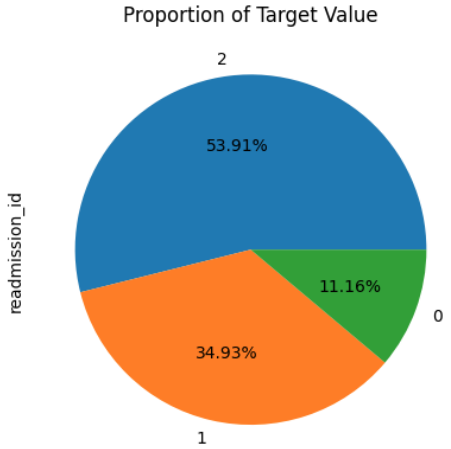
Authors:

Madhav Sood IMT2021009
Vineet Priyedarshi IMT2021018
Shlok Agrawal IMT2021103

EDA and Pre-processing:

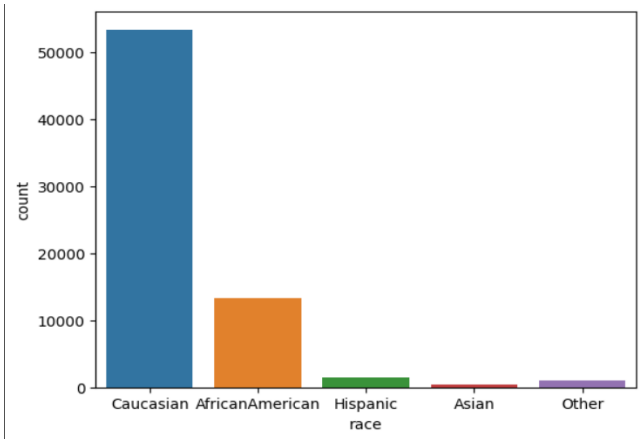
Distribution and Proportion of Target Values:

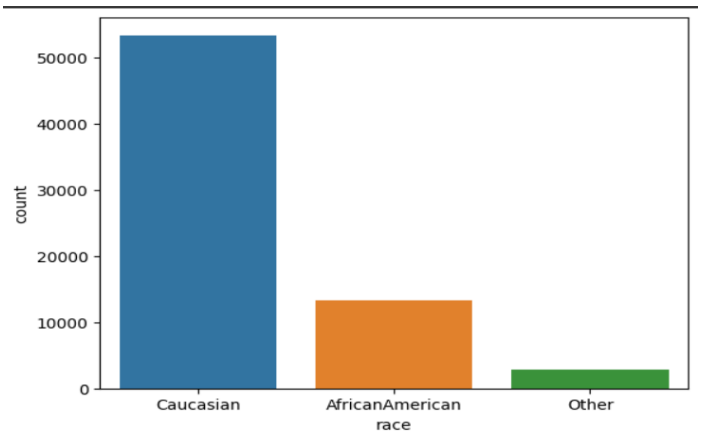




Race:

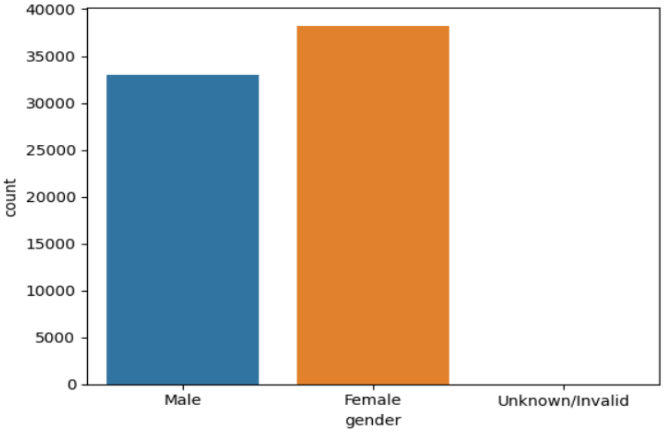
Since Asian and Hispanic have negligible data, we group them up with ‘Other’.

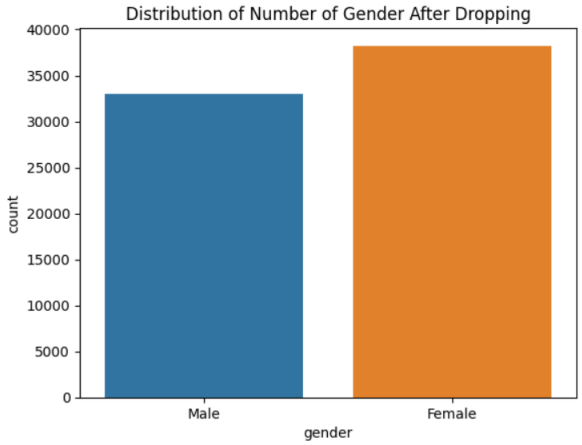




Gender:

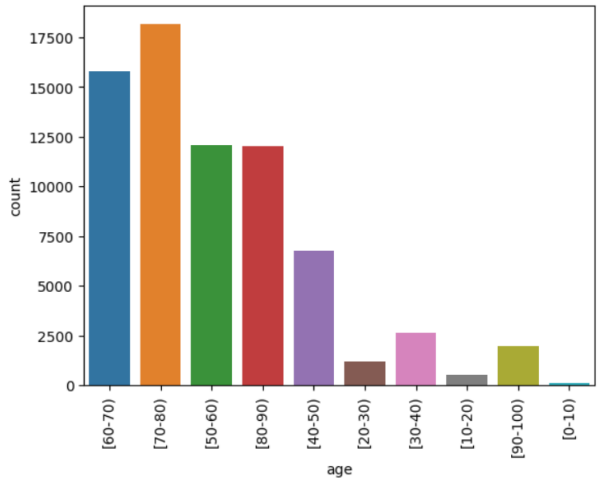
We drop all rows with Unknown/Invalid value as gender.

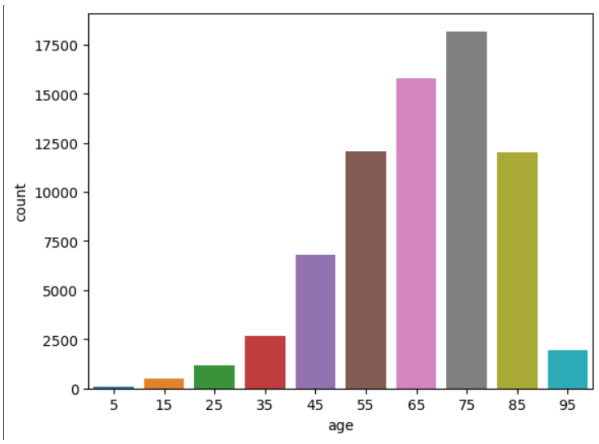




Age:

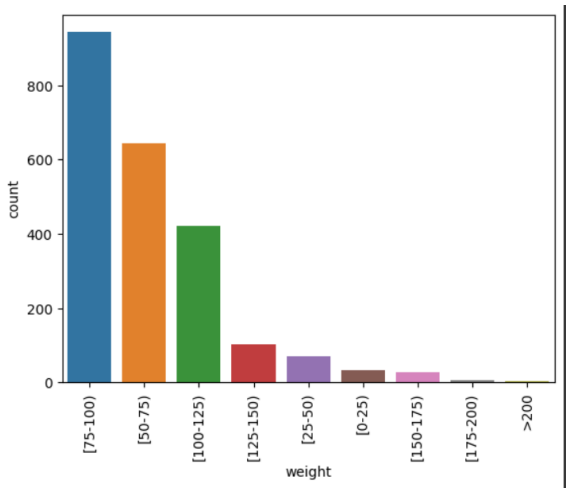
We transform the ages of patients within the range of (a, b) into the average age, which is (a + b) / 2.

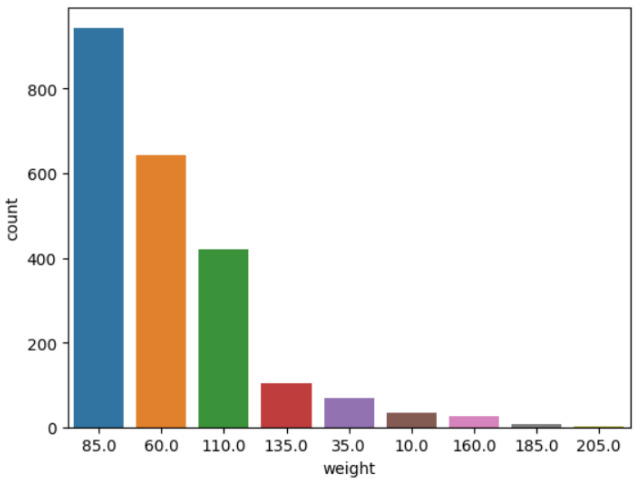




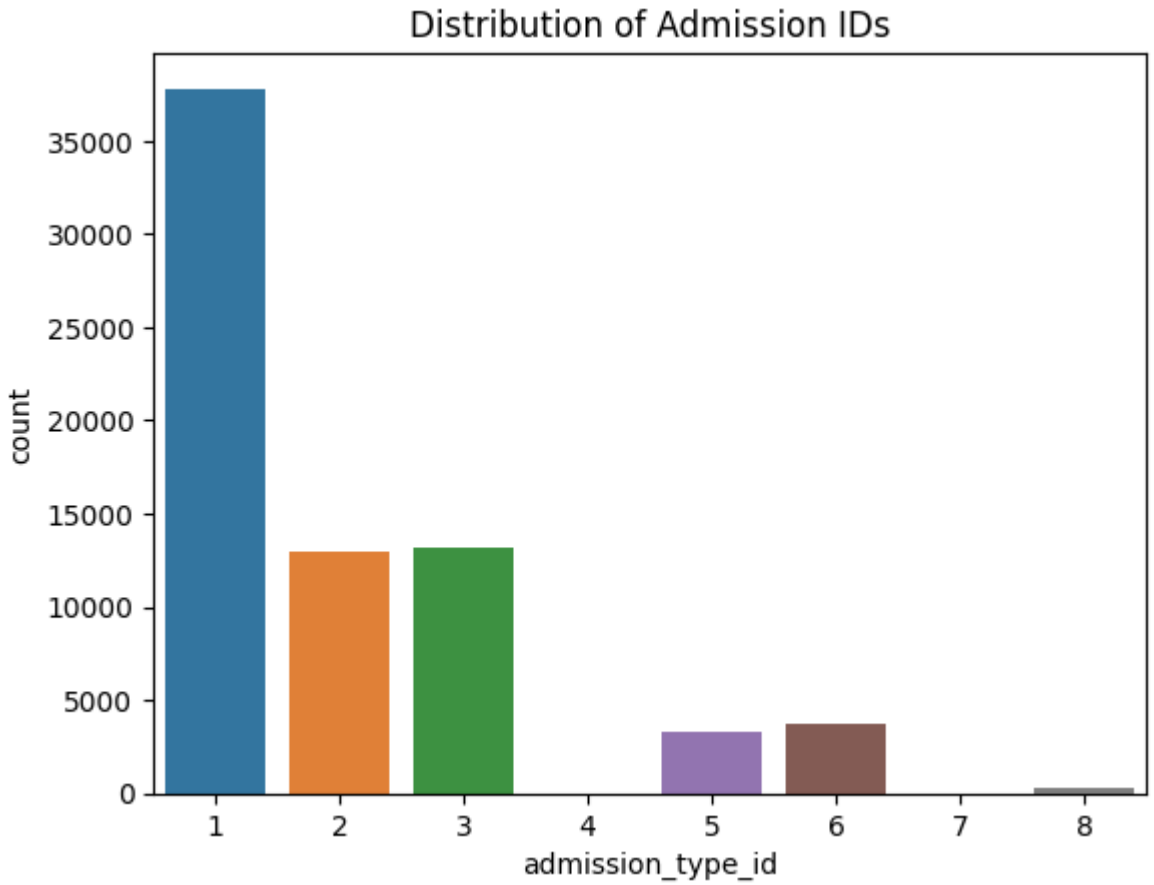
Weight:

We transform the weights of patients within the range of (a, b) into the average weight, which is (a + b) / 2.

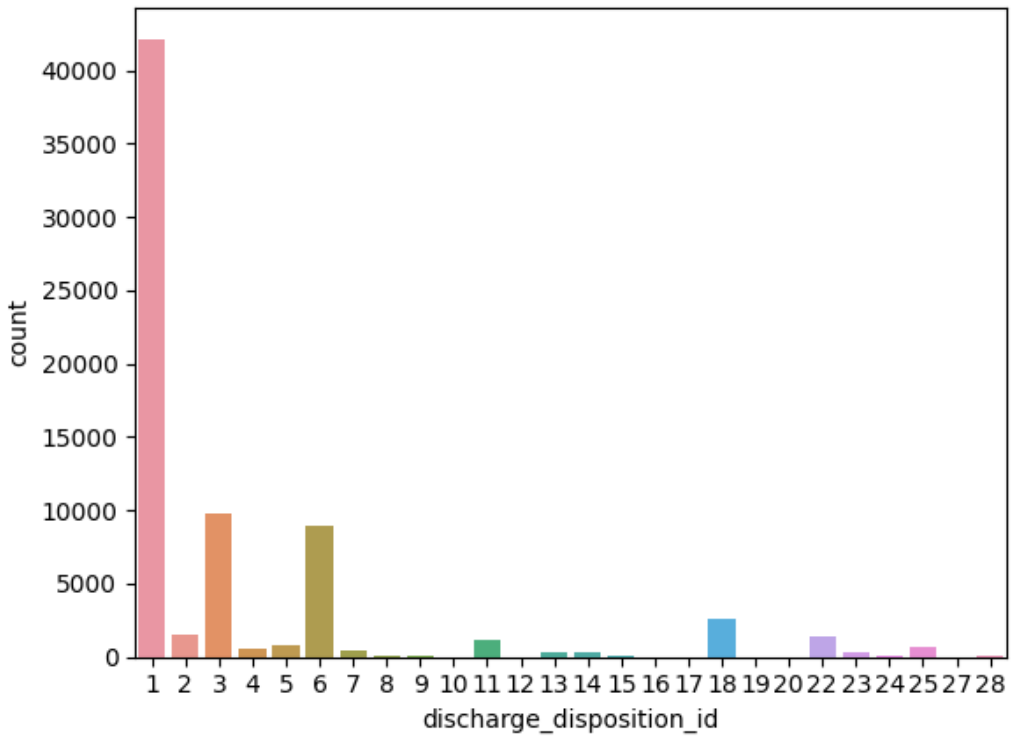




Admission_type_id:

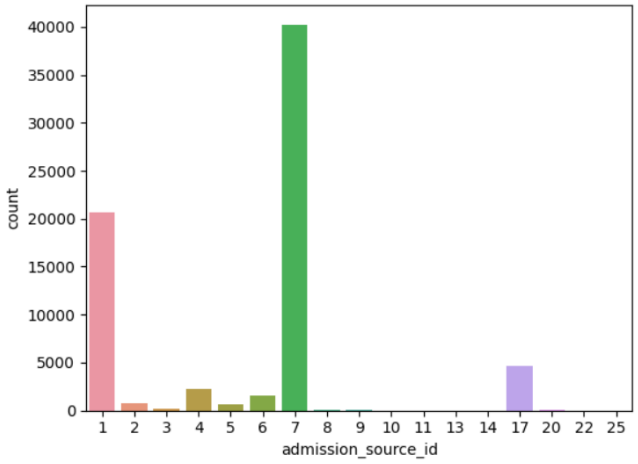


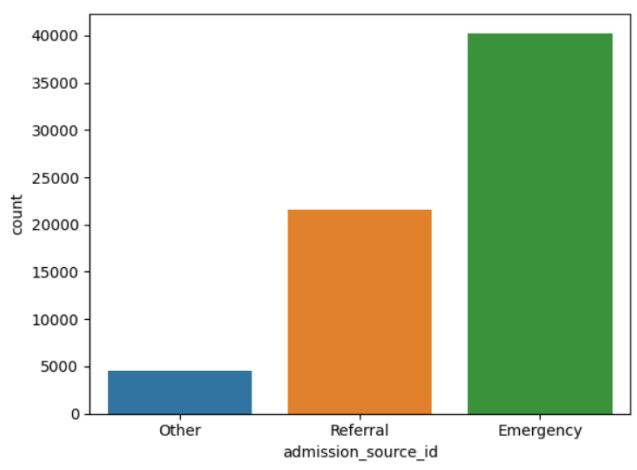
Discharge_disposition_id:



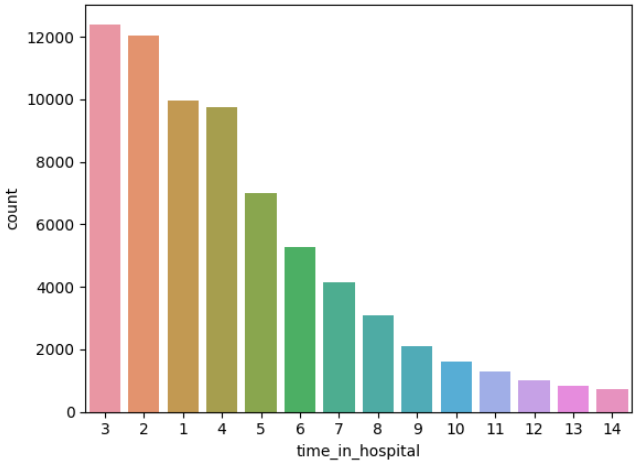
Admission_source_id:

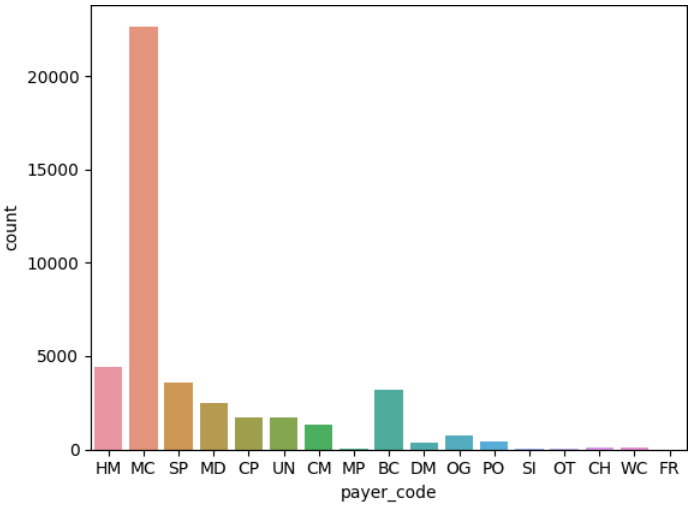
We classify admission_source into three classes - referral, emergency, other.





Time_in_hospital and Payer_code:



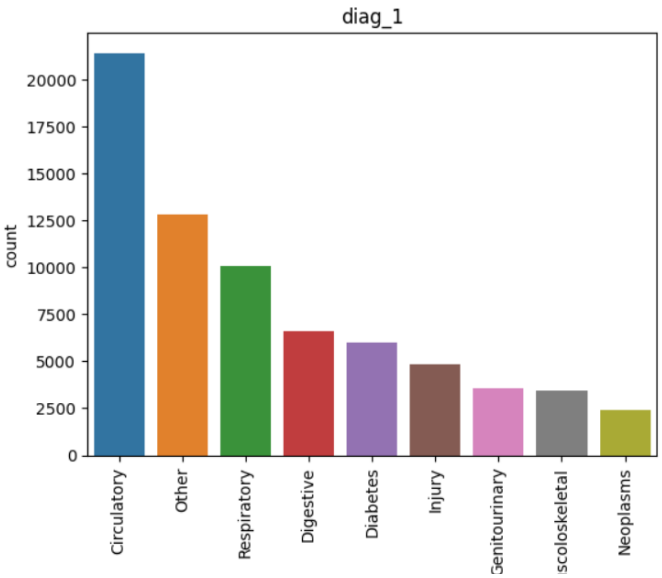


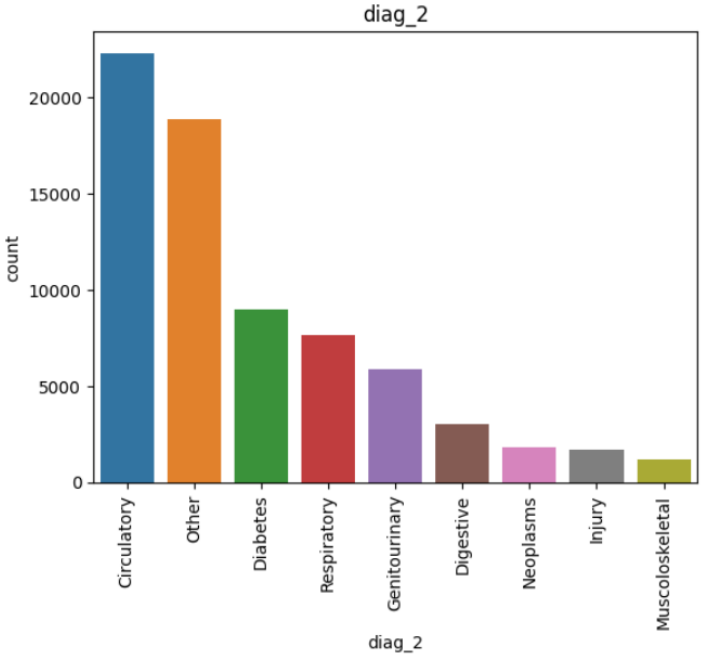
Diag1, Diag2, Diag3:

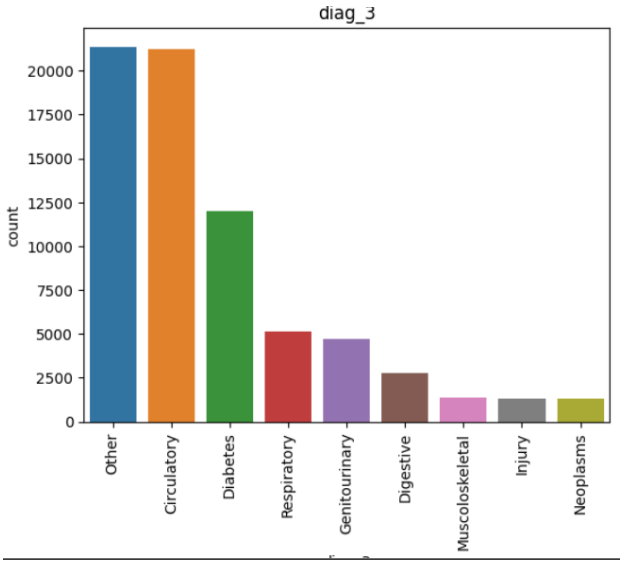
- diag_1: The primary diagnosis (coded as first three digits of ICD9); 848 distinct values
- diag_2: Secondary diagnosis (coded as first three digits of ICD9); 923 distinct values
- diag_3: Additional secondary diagnosis (coded as first three digits of ICD9); 954 distinct values

We instead classify them as categorical values:

- Circulatory - between 390 - 450 or 785
- Respiratory - between 460 - 520 or 786
- Digestive - between 520 - 580 or 787
- Diabetes - between 250 - 251
- Injury - between 800 - 999
- Muscoloskeletal - between 710 - 740
- Genitourinary - between 580 - 630 or 788
- Neoplasm - between 140 - 240
- Other -

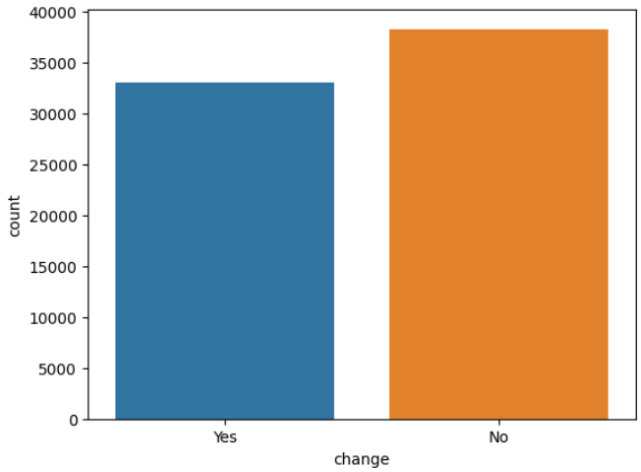






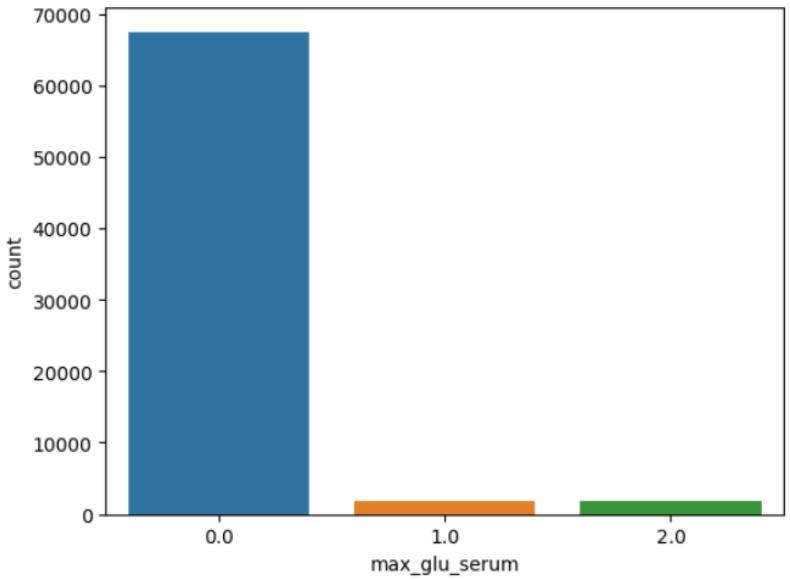
Change: If there was a change in medication

The change column has 2 values, Yes or No. We have changed “ch” to “Yes”.



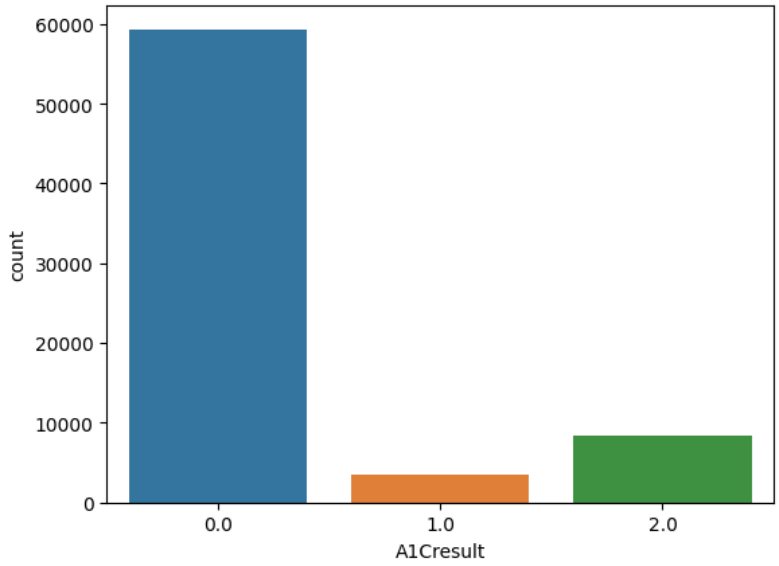
max_glu_serum:

The column has 3 values,
NULL → 0
Norm → 1
>200 and >300 → 2



A1c test result:

NULL → 0
Norm → 1
>7 and >8 → 2



Feature Generation: Patient_id_frequency

We used the frequency of patient_id to create create a new feature, patient_id_frequency. We have considered the test and train data as dependent for this feature, and have added the frequency of the train data to the test data for the particular patient_id. This was the key insight to get the score of 0.7+.

Handling NULL Values:

Percentage of NULL Values in Columns:

```
race                   2.272823
weight                96.841352
admission_source_id    6.846546
payer_code            39.556105
medical_specialty     49.033454
dtype: float64
```

- ‘weight', 'payer_code', 'medical_specialty’ were dropped.
- ‘Race’ and ‘admission_source_id’ were filled with mode values.

Dropping columns:

Dropped columns of some drugs as they were imbalanced.
repaglinide', 'nateglinide', 'chlorpropamide', 'acetohexamide', 'tolbutamide', 'acarbose', 'miglitol', 'troglitazone', 'tolazamide', 'examide', 'citoglipton', 'glyburide-metformin', 'glipizide-metformin', 'metformin-rosiglitazone', 'metformin-pioglitazone', 'glimepiride-pioglitazone' were dropped.

Encoding:

Dummy encoded the categorical columns 'race', 'gender', 'admission_source_id', 'diag_1', 'diag_2', 'diag_3', 'metformin', 'glimepiride', 'glipizide', 'glyburide', 'pioglitazone', 'rosiglitazone', 'insulin', 'change', 'diabetesMed'.

Models Tried:

We first try **Non neural ensembles:**

Test-Train split is 20-80.

Random Forest:

Prediction on the training dataset:

```
Accuracy: 0.6901101986383098
0.48452001638832104
Classification Report:
              precision    recall  f1-score   support

           0       0.45      0.02      0.04      1554
           1       0.61      0.61      0.61      4990
           2       0.74      0.88      0.80      7703

    accuracy                           0.69     14247
   macro avg       0.60      0.50      0.48     14247
weighted avg       0.66      0.69      0.65     14247
```

On the test dataset: 0.708

Feature Importance by the random forest classifier:

- frequency: 0.1120307343885143
- enc_id: 0.07872950864227435
- patient_id: 0.07744797173713146
- num_lab_procedures: 0.06349947714367646

- num_medications: 0.05661497915647388
- time_in_hospital: 0.042745179263562756
- age: 0.035882631344346626
- number_inpatient: 0.03441714319795621
- discharge_disposition_id: 0.02953866182686755
- num_procedures: 0.02933302844702797
- number_diagnoses: 0.02896792148525359
- admission_type_id: 0.020931401887635287
- number_outpatient: 0.0141691206911626
- diag_2_Circulatory: 0.01151537238728194
- diag_3_Other: 0.011490357656518783
- diag_3_Circulatory: 0.01147980906486234
- number_emergency: 0.011287277968881834
- gender_Female: 0.011193961712651678
- diag_1_Circulatory: 0.01117442522108308
- gender_Male: 0.0110639885176475
- diag_2_Other: 0.010936823783497591
- A1Cresult: 0.01088053771243707
- diag_1_Other: 0.00943089459124816
- diag_3_Diabetes: 0.00896875287996946
- insulin_Steady: 0.008899971565210116
- race_Caucasian: 0.008858597997484282
- race_AfricanAmerican: 0.008232879612574753
- insulin_No: 0.00797248029288315
- diag_1_Respiratory: 0.007712344116043658 5
- diag_2_Diabetes: 0.007515202177789529
- admission_source_id_Emergency: 0.007498621191639368
- diag_2_Respiratory: 0.007418958901423863
- admission_source_id_Referral: 0.00727921866395266
- change_No: 0.0072030506599767685
- change_Yes: 0.00720084962676118
- metformin_No: 0.006812130760256965
- diag_1_Digestive: 0.00673525307728447
- diag_2_Genitourinary: 0.006672656931518185
- metformin_Steady: 0.006630506310605617
- diag_3_Respiratory: 0.006389720974986572
- glipizide_No: 0.006350754362968675
- diag_1_Diabetes: 0.005917147333625065
- insulin_Down: 0.005901108424588778
- glipizide_Steady: 0.005895859996528847
- diag_1_Injury: 0.005796439088684599
- insulin_Up: 0.005731660194740521
- diag_3_Genitourinary: 0.00568575267915593
- glyburide_No: 0.005640622048135199
- glyburide_Steady: 0.005334835964549822
- diabetesMed_Yes: 0.004679267738957148
- diag_1_Genitourinary: 0.004638364720728753
- diabetesMed_No: 0.004526294167789058

- pioglitazone_No: 0.00446233737227704
- pioglitazone_Steady: 0.004376128702185501
- max_glu_serum: 0.004205576969335931
- diag_2_Digestive: 0.004150165455050339
- diag_3_Digestive: 0.0041224348034043234
- rosiglitazone_No: 0.003935490001721713
- rosiglitazone_Steady: 0.003877527234807635
- glimepiride_No: 0.0036439649038062425
- diag_1_Muscoloskeletal: 0.0036348744292445034
- admission_source_id_Other: 0.0035163467182060455
- race_Other: 0.0034755931428602114
- glimepiride_Steady: 0.0033595902298634134
- diag_1_Neoplasms: 0.003050487577261583
- diag_2_Neoplasms: 0.0028837717872590824
- diag_2_Injury: 0.0025734511144516084
- diag_3_Neoplasms: 0.0023705315714694216
- diag_3_Muscoloskeletal: 0.002322168966629193
- diag_3_Injury: 0.0023070206903689277
- diag_2_Muscoloskeletal: 0.0019287527248025807
- metformin_Up: 0.0010483374882502353
- glipizide_Up: 0.001047405353112099
- glyburide_Up: 0.0009507636066328211
- glipizide_Down: 0.0008645475754104656
- glyburide_Down: 0.0007641883161276305
- metformin_Down: 0.0006274946179293416
- pioglitazone_Up: 0.00042217662933528464
- glimepiride_Up: 0.0004119074836461238
- glimepiride_Down: 0.00030063625249450665
- pioglitazone_Down: 0.00019178417898271178
- rosiglitazone_Up: 0.00018645191414185146
- rosiglitazone_Down: 0.0001275819001258089

Random Forest with Random OverSampling:

Prediction on the training dataset:

```
Accuracy: 0.6908822910086334
F1 Macro: 0.5104639139993035
Classification Report:
              precision    recall  f1-score   support

           0       0.41      0.07      0.11      1554
           1       0.60      0.63      0.62      4990
           2       0.75      0.85      0.80      7703

    accuracy                           0.69     14247
   macro avg       0.59      0.52      0.51     14247
weighted avg       0.66      0.69      0.66     14247
```

On the test dataset: 0.704

Random Forest with Boosting:

```
Accuracy: 0.6912332420860532
F1 Macro: 0.48593510705841503
Classification Report:
              precision    recall  f1-score   support

           0       0.44      0.02      0.04      1554
           1       0.61      0.62      0.61      4990
           2       0.74      0.87      0.80      7703

    accuracy                           0.69     14247
   macro avg       0.60      0.50      0.49     14247
weighted avg       0.66      0.69      0.65     14247
```

On the test data: 0.708

Random Forest with bagging:

```
Accuracy: 0.6929178072576683
F1 Macro: 0.4791334928025015
Classification Report:
              precision    recall  f1-score   support

           0       0.59      0.01      0.02      1554
           1       0.61      0.61      0.61      4990
           2       0.74      0.88      0.80      7703

    accuracy                           0.69     14247
   macro avg       0.65      0.50      0.48     14247
weighted avg       0.68      0.69      0.65     14247
```

On the test data: **0.709**

lightGBM (without hyperparameter optimization):

Prediction on the training dataset:

```
Accuracy: 0.7047799536744578
Classification Report:
              precision    recall  f1-score   support

           0       0.57      0.07      0.13      1554
           1       0.64      0.61      0.62      4990
           2       0.74      0.89      0.81      7703

    accuracy                           0.70     14247
   macro avg       0.65      0.53      0.52     14247
weighted avg       0.69      0.70      0.67     14247
```

On the test dataset: 0.725

lightGBM (with hyperparameter optimization):

Prediction on the training dataset:

```
Accuracy: 0.7054818558292973
Classification Report:
              precision    recall  f1-score   support

           0       0.59      0.07      0.12      1554
           1       0.63      0.62      0.62      4990
           2       0.74      0.89      0.81      7703

    accuracy                           0.71     14247
   macro avg       0.66      0.53      0.52     14247
weighted avg       0.69      0.71      0.67     14247

Confusion Matrix:
[[ 106  957  491]
 [  57 3070 1863]
 [  17  811 6875]]
```

On the test dataset: 0.728

lightGBM (with hyperparameter optimization) and oversampling:

Prediction on the training dataset:

```
Accuracy (with oversampling): 0.6399241945672773
Classification Report (with oversampling):
              precision    recall  f1-score   support

           0       0.26      0.44      0.32      1554
           1       0.61      0.45      0.52      4990
           2       0.78      0.80      0.79      7703

    accuracy                           0.64     14247
   macro avg       0.55      0.57      0.55     14247
weighted avg       0.67      0.64      0.65     14247
```

On the test dataset: 0.644

lightGBM (with hyperparameter optimization) and bagging:

Prediction on the training dataset:

```
Accuracy (with bagging): 0.7054818558292973
Classification Report (with bagging):
              precision    recall  f1-score   support

           0       0.59      0.07      0.12      1554
           1       0.63      0.62      0.62      4990
           2       0.74      0.89      0.81      7703

    accuracy                           0.71     14247
   macro avg       0.66      0.53      0.52     14247
weighted avg       0.69      0.71      0.67     14247

Confusion Matrix (with bagging):
[[ 106  957  491]
 [  57 3070 1863]
 [  17  811 6875]]
```

On the test dataset: 0.728 same as before

lightGBM (with hyperparameter optimization) and bagging and boosting:

Prediction on the training dataset:

```
Accuracy (with boosting): 0.7053414753983295
Classification Report (with boosting):
              precision    recall  f1-score   support

           0       0.56      0.08      0.14      1554
           1       0.64      0.61      0.63      4990
           2       0.74      0.89      0.81      7703

    accuracy                           0.71     14247
   macro avg       0.65      0.53      0.52     14247
weighted avg       0.69      0.71      0.67     14247

Confusion Matrix (with boosting):
[[ 122  932  500]
 [  68 3063 1859]
 [  27  812 6864]]
```

On the test dataset: 0.728 same as before

We then tried different pre processing. For that we again tried a few ensembles:

Random Forest with boosting:

Prediction on training dataset:

```
Accuracy: 0.698694553621561
F1 Macro: 0.49049040296773244
Classification Report:
              precision    recall  f1-score   support

           0       0.59      0.01      0.03      1710
           1       0.60      0.67      0.63      4989
           2       0.76      0.88      0.81      7549

    accuracy                           0.70     14248
   macro avg       0.65      0.52      0.49     14248
weighted avg       0.68      0.70      0.66     14248
```

On the test dataset:**0.702** which is a bit less compared to 0.708 by our previous pre processing.

Neural Networks

Neural Networks Experiments with old Preprocessing:

- First run with 3 layers and scaling: 0.552 (hidden layer1: 64, layer2: 32, layer3: 16 neurons, 64 batch size)
- Dropping enc_id with 3 layers and scaling: 0.549
- Weighted loss function inversely proportional to frequency of target variables: 0.555
- Weighted loss function: [10, 6, 2] as weights: 0.566 (not dropped enc_id)
- Weighted loss function: [12, 7, 1] as weights: 0.575 (not dropped enc_id)
- Same thing with dropped enc_id: 0.572
- Implementing xavier_initialisation: 0.568
- Adding four layers: 0.565
- 20 epochs (for 3 layers with base accuracy 0.569): 0.571
- 2 Layers, 15 epochs: 0.567
- 4 layers, 5 epochs, 512 batch size: 0.563
- Removed weights from last: 0.571
- Removed normalization: 0.544
- 15 epochs: 0.555

New Preprocessing

Model- NN with 4 layers, 5 epochs to measure accuracy

Dropping columns:

- ‘Enc_id’, ‘patient_id’: 0.71
- 'patient_id', 'enc_id', 'readmission_id', 'repaglinide', 'nateglinide', 'chlorpropamide', 'acetohexamide', 'tolbutamide', 'acarbose', 'miglitol', 'troglitazone', 'tolazamide', 'examide', 'citoglipton', 'glyburide-metformin', 'glipizide-metformin', ‘metformin-rosiglitazone', 'metformin-pioglitazone', 'glimepiride-pioglitazone': 0.717
- 'patient_id', 'enc_id', 'readmission_id', 'repaglinide', 'nateglinide', 'chlorpropamide', 'acetohexamide', 'tolbutamide', 'acarbose', 'miglitol', 'troglitazone', 'tolazamide', 'examide', 'citoglipton', 'glyburide-metformin', 'glipizide-metformin', ‘metformin-rosiglitazone', 'metformin-pioglitazone', 'glimepiride-pioglitazone', 'weight', 'payer_code', 'medical_specialty': 0.7
- Keeping ‘medical_specialty’: 0.704
- Keeping ‘payer_code’: 0.71
- Deciding to not drop any columns with high null values.
- Keeping ‘end_id’ : 0.535
- Keeping ‘patient_id’: 0.54

Final Dropped: 'patient_id', 'enc_id', 'readmission_id', 'repaglinide', 'nateglinide', 'chlorpropamide', 'acetohexamide', 'tolbutamide', 'acarbose', 'miglitol', 'troglitazone', 'tolazamide', 'examide', 'citoglipton', 'glyburide-metformin', 'glipizide-metformin', ‘metformin-rosiglitazone', 'metformin-pioglitazone', 'glimepiride-pioglitazone': 0.717

Other Preprocessing:

NULL Values:

Replaced all NULL Values with ‘most_frequent’ or mode.
Removing any of the columns with high percentage of NULL values was leading to decreased accuracy.

One-Hot Encoding:

One-hot encoded all categorical columns (columns with ‘object’ as the data type).

New Neural Network Experiments:

Baseline: NN with 4 layers, 5 epochs, 512 batch size, 1

- 4 layers

- 5 epochs
- 512 batch size
- 1e-3 learning rate
- No data normalization
- ReLU for all hidden layers
- Cross Entropy Loss with no weights
- Adam Optimisation
- model = ANN(
  in_dim = in_dim,
  hidden_dim_1 = in_dim // 2,
  hidden_dim_2 = in_dim // 4,
  hidden_dim_3 = in_dim // 8,
  hidden_dim_4 = 3
).to(device)

Layers:

New: model = ANN(
in_dim = in_dim,
hidden_dim_1 = 512,
hidden_dim_2 = 128,
hidden_dim_3 = 32,
hidden_dim_4 = 4
).to(device) : 0.71 -> sticking to old one but doesn't make a meaningful difference

- 3 layers: 0.676, sticking to the old 4 hidden layer system.
- 5 layers instead of 4: 0.711 -> doesn't matter

Batch Size:

- 64 size batches instead of 512: 0.699 -> worse
- 1024 size batch: 0.714 -> same

Learning Rate:

- Learning Rate 1e-4 instead of 1e-3: 0.576
- Learning Rate 5e-2: 0.687

Data Normalization:

- Normalizing Data: 0.702

Activation Functions:

- Sigmoid in the last layer instead of ReLU: 0.711
- LeakyReLU all the layers: 0.716

Loss Functions:

- Weighted loss function: [12, 7, 1] as weights: 0.55
- Multi Margin Loss instead of Cross Entropy Loss: 0.711

Optimisation:

- RMSProp instead of Adam: 0.621

Dropout:

- 0.2 dropout instead of 0.3: 0.629
- 0.4 dropout instead of 0.3: 0.714
- 0.5 dropout: 0.713

Epochs:

- 10 epochs instead of 5: 0.709

Conclusion:

Nothing we tried beats our baseline model. A lot of things are equivalent, some are worse.

```
========= EPOCH 1 STARTED =========
100%|██████████| 112/112 [00:26<00:00,  4.24it/s]
========= TRAIN EVALUATION STARTED =========
100%|██████████| 112/112 [00:12<00:00,  8.65it/s]
========= TEST EVALUATION STARTED =========
100%|██████████| 28/28 [00:03<00:00,  7.51it/s]
END OF 1 EPOCH
| Time taken:  50.544 |
| Train Loss:   0.967 | Train acc:  0.69625 | Train f1:  0.48251 |
| Test Loss:   0.969  | Test acc:  0.68866  | Test f1:  0.47701  |
========= EPOCH 2 STARTED =========
100%|██████████| 112/112 [00:25<00:00,  4.41it/s]
========= TRAIN EVALUATION STARTED =========
100%|██████████| 112/112 [00:12<00:00,  8.66it/s]
========= TEST EVALUATION STARTED =========
100%|██████████| 28/28 [00:02<00:00, 11.03it/s]
END OF 2 EPOCH
| Time taken:  49.217 |
| Train Loss:   0.919 | Train acc:  0.69516 | Train f1:  0.48002 |
| Test Loss:   0.925  | Test acc:  0.68276  | Test f1:  0.47083  |
========= EPOCH 3 STARTED =========
100%|██████████| 112/112 [00:25<00:00,  4.36it/s]
========= TRAIN EVALUATION STARTED =========
100%|██████████| 112/112 [00:13<00:00,  8.54it/s]
========= TEST EVALUATION STARTED =========
100%|██████████| 28/28 [00:02<00:00, 11.08it/s]
END OF 3 EPOCH
| Time taken:  50.298 |
| Train Loss:   0.858 | Train acc:  0.71659 | Train f1:  0.48817 |
| Test Loss:   0.865  | Test acc:  0.70768  | Test f1:  0.48095  |
========= EPOCH 4 STARTED =========
100%|██████████| 112/112 [00:26<00:00,  4.24it/s]
========= TRAIN EVALUATION STARTED =========
100%|██████████| 112/112 [00:13<00:00,  8.20it/s]
========= TEST EVALUATION STARTED =========
100%|██████████| 28/28 [00:02<00:00,  9.81it/s]
END OF 4 EPOCH
| Time taken:  51.879 |
| Train Loss:   0.856 | Train acc:  0.71127 | Train f1:  0.48877 |
| Test Loss:   0.866  | Test acc:  0.70010  | Test f1:  0.47974  |
========= EPOCH 5 STARTED =========
```

SVM:

On running the SVM model with our existing pre processing (with mapping and dummy encoding) we got really less predicted value.

- First we run the dataset on sklearn svm (kernel = rgb, C=1). accuracy = 0.551

  Weighted SVM giving more priority to least weights.
- SVM is not that good with imbalanced dataset Our dataset is imbalanced. Trying out Weighted svm. Accuracy = **0.557**

  Grid Search
- We tried running grid search but it took way too long. We decided to stop it.

We then decided to change our pre processing. We decided to do similar pre processing as what we did for neural networks.

On the train data set:

```
Accuracy: 0.6348961257720381
0.41336029311356626
Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00      1710
           1       0.61      0.39      0.48      4989
           2       0.64      0.94      0.76      7549

    accuracy                           0.63     14248
   macro avg       0.42      0.44      0.41     14248
weighted avg       0.55      0.63      0.57     14248
```

Inference:

We can see that our model is not able to classify 0 correctly. The reason to this is that 0 is very few in number and SVM are not well dealt with imbalanced dataset. Yet the accuracy has increased.

On the test data set we got:

- When kernel = poly: **0.644**
- When kernel = rbf: **0.665**

This took us more than 3 hours to train+test.
To tackle the imbalance nature of dataset, we decided to go for weighted SVM with the pre processing. However it took more than 3 hours to train so we decided to terminate it.