In [134]:
```
import pandas as pd
import numpy as np
```

In [135]:
```
data = pd.read_csv("Data4Modelling.csv")
```

In [136]:
```
data = data.replace([np.inf, -np.inf], 0)
```

In [137]:
```
data
```

Out[137]:

	race	gender	age	weight	admission_type_id	discharge_disposition_id	admi
0	Caucasian	Female	5	NaN	5	18	
1	Caucasian	Female	15	NaN	1	1	
2	AfricanAmerican	Female	25	NaN	1	1	
3	Caucasian	Male	35	NaN	1	1	
4	Caucasian	Male	45	NaN	1	1	
...	...	...	...	...	...	...	
100109	AfricanAmerican	Male	75	NaN	1	2	
100110	AfricanAmerican	Female	85	NaN	1	2	
100111	Caucasian	Male	75	NaN	1	1	
100112	Caucasian	Female	85	NaN	1	2	
100113	Caucasian	Male	75	NaN	1	1	

100114 rows × 42 columns

In [138]:
```
non_numeric_columns = data.select_dtypes(exclude=[float, int])
```

In [139]:
```
non_numeric_columns
```

Out[139]:

	race	gender	weight	medical_specialty	diag_1	diag_2	diag_3
0	Caucasian	Female	NaN	pediatrics	diabetes	diabetes	diabetes
1	Caucasian	Female	NaN	missing	other	other	other
2	AfricanAmerican	Female	NaN	missing	pregnecy	pregnecy	other
3	Caucasian	Male	NaN	missing	other	other	circulatory
4	Caucasian	Male	NaN	missing	neoplasms	neoplasms	diabetes
...	...	...	...	...	...	...	...
100109	AfricanAmerican	Male	NaN	missing	diabetes	diabetes	circulatory
100110	AfricanAmerican	Female	NaN	missing	digestive	digestive	digestive
100111	Caucasian	Male	NaN	missing	other	other	other
100112	Caucasian	Female	NaN	surgery	injury	injury	injury
100113	Caucasian	Male	NaN	missing	digestive	digestive	digestive

100114 rows × 7 columns

In [140]:
```
data["weight"]  = ["UNK" if str(i) == str(np.nan) else i for i in data["weight
data.weight.value_counts()
```

Out[140]:
```
weight
UNK          96958
[75-100)      1320
[50-75)        881
[100-125)      622
[125-150)      143
[25-50)         94
[0-25)          48
[150-175)       34
[175-200)       11
>200             3
Name: count, dtype: int64
```

In [141]:
```
weightDict = {'[50-75)' : '62',
'[75-100)' : '87',
'[100-125)' : '112',
'[125-150)' : '137',
'[25-50)' : '37',
'[0-25)' : '12',
'[150-175)' : '162',
'[175-200)' : '187',
'>200' : '200',
'UNK' : f"{np.nan}"}

data['weight'] = data['weight'].apply(lambda x : weightDict[x])
```

In [142]:
```
data.weight = data.weight.astype(float)
```

In [143]:
```
data_checkout = data.copy()
```

In [144]:
```
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
```

In [145]:
```
for column in data.columns:
    if data[column].dtype == 'object':
        data[column] = label_encoder.fit_transform(data[column])
```

In [146]:
```
data['readmitted'].value_counts()
```

Out[146]:
```
readmitted
0    88757
1    11357
Name: count, dtype: int64
```

In [147]:
```
from sklearn.model_selection import train_test_split
```

In [148]:
```
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split,KFold,StratifiedKFold,crc
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score, f1_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import AdaBoostClassifier,GradientBoostingClassifier,Rar
from lightgbm import LGBMClassifier
from catboost import CatBoostClassifier
from xgboost import XGBClassifier
from tabulate import tabulate
```

In [149]:
```
def BasedModel():
    basedModels = []
    basedModels.append(('LR'   , LogisticRegression()))
    basedModels.append(('LDA'  , LinearDiscriminantAnalysis()))
    basedModels.append(('RF'   , RandomForestClassifier()))
    basedModels.append(('NB'   , GaussianNB()))
    basedModels.append(('AB'   , AdaBoostClassifier()))
    basedModels.append(('GBM'  , GradientBoostingClassifier()))
    basedModels.append(('ET'   , ExtraTreesClassifier()))
    basedModels.append(('XG'   , XGBClassifier()))
    basedModels.append(('LG'   , LGBMClassifier()))
    basedModels.append(('CAT'   , CatBoostClassifier(silent=True)))
    return basedModels
```

In [150]:
```
def BasedLine(df, method, models, drop = False):
    df_check = df.copy()

    df_check.weight = df_check.weight.fillna(method)
    if drop == True:
        df_check.drop("weight",axis = 'columns', inplace = True)

    y = df_check['readmitted']
    X = df_check.drop(columns = 'readmitted')

    # split data into train and validation set
    X_train, X_valid, y_train, y_valid = train_test_split(X, y, test_size=0.2,
    # Test options and evaluation metric
    scoring = 'accuracy'
    results, results_weigh = [],[]
    names = []
    scores, scores_weigh = [],[]
    data = []
    for name, model in models:
        model.fit(X_train, y_train)

        cv_results = cross_validate(model, X_train, y_train, scoring=['f1_weig
        cv_weigh = cv_results["test_f1_weighted"].mean()
        cv_non = cv_results["test_f1"].mean()
        score_non = f1_score(model.predict(X_valid), y_valid)
        score_weigh = f1_score(model.predict(X_valid), y_valid,  average='weig

        results.append(cv_non)
        results_weigh.append(cv_weigh)
        names.append(name)
        scores.append(score_non)
        scores_weigh.append(score_weigh)
        data.append([name,cv_non, score_non, cv_weigh,score_weigh])
    print(tabulate(data, headers=["Model", "CV F1 Score", "Model F1 Score","CV

    return names, results, scores
```

In [151]:
```
models = BasedModel()
```

In [152]:
```
weight_mean = data.weight.mean()
weight_median = data.weight.median()
weight_mode = data.weight.mode()
```

In [153]:
```
weight_mean, weight_median, weight_mode
```

Out[153]:
```
(85.84790874524715,
 87.0,
 0    87.0
 Name: weight, dtype: float64)
```

Median and Mode are the same

Mean Weight Imputation

In [155]:
```
#result for filling with mean
names,results, scores = BasedLine(df = data, method=weight_mean, models = mode
```

```
C:\Users\user\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.p
y:460: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown i
n:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://sc
ikit-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-reg
ression (https://scikit-learn.org/stable/modules/linear_model.html#logisti
c-regression)
  n_iter_i = _check_optimize_result(
C:\Users\user\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.p
y:460: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown i
n:
```

In [156]:
```
data
```

Out[156]:

	race	gender	age	weight	admission_type_id	discharge_disposition_id	admission_sour
0	2	0	5	NaN	5	18	
1	2	0	15	NaN	1	1	
2	0	0	25	NaN	1	1	
3	2	1	35	NaN	1	1	
4	2	1	45	NaN	1	1	
...	...	...	...	...	...	...	
100109	0	1	75	NaN	1	2	
100110	0	0	85	NaN	1	2	
100111	2	1	75	NaN	1	1	
100112	2	0	85	NaN	1	2	
100113	2	1	75	NaN	1	1	

100114 rows × 42 columns

Median Weight Imputation

In [157]:
```
#result for filling with mean
names,results, scores = BasedLine(df = data,method=weight_median, models = moc
C:\Users\user\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py:4
60: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://sciki
t-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regres
sion (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regr
ession)
  n_iter_i = _check_optimize_result(
C:\Users\user\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py:4
60: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://sciki
t-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regres
sion (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regr
ession)
  n_iter_i = _check_optimize_result(
C:\Users\user\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py:4
60: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://sciki
t-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regres
sion (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regr
ession)
  n_iter_i = _check_optimize_result(
C:\Users\user\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py:4
60: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://sciki
t-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regres
sion (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regr
ession)
  n_iter_i = _check_optimize_result(
C:\Users\user\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py:4
60: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://sciki
t-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regres
```

```
sion (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regr
ession)
  n_iter_i = _check_optimize_result(
C:\Users\user\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py:4
60: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://sciki
t-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regres
sion (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regr
ession)
  n_iter_i = _check_optimize_result(

| Model    |   CV F1 Score |   Model F1 Score |   CV F1 Weighted |   Model F1
Weighted |
|---------+--------------+-----------------+-----------------+------------
---------|
| LR       |     0.0310411 |       0.0266437  |         0.836322 |
0.936712 |
| LDA      |     0.0738366 |       0.0664775  |         0.840131 |
0.929524 |
| RF       |     0.0239893 |       0.0216169  |         0.835545 |
0.937412 |
| NB       |     0.211293  |       0.207757   |         0.823763 |
0.822954 |
| AB       |     0.0237583 |       0.0241796  |         0.835488 |
0.937356 |
| GBM      |     0.0206729 |       0.0104895  |         0.83531  |
0.938554 |
| ET       |     0.0272913 |       0.0224428  |         0.835834 |
0.937171 |
| XG       |     0.0401864 |       0.0297746  |         0.836966 |
0.935172 |
| LG       |     0.0189771 |       0.00614574 |         0.835098 |
0.939029 |
| CAT      |     0.0310322 |       0.0224138  |         0.83629  |
0.93695  |
```

```
In [ ]:
```

Without Weight Imputation

```
In [158]: #result for filling with mean
          names,results, scores = BasedLine(df = data,method=weight_mode, models = model
C:\Users\user\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py:4
60: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://sciki
t-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regres
sion (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regr
ession)
  n_iter_i = _check_optimize_result(
C:\Users\user\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py:4
60: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://sciki
t-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regres
sion (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regr
ession)
  n_iter_i = _check_optimize_result(
C:\Users\user\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py:4
60: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://sciki
t-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regres
sion (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regr
ession)
  n_iter_i = _check_optimize_result(
C:\Users\user\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py:4
60: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://sciki
t-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regres
sion (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regr
ession)
  n_iter_i = _check_optimize_result(
C:\Users\user\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py:4
60: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://sciki
t-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regres
```

```
sion (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regr
ession)
  n_iter_i = _check_optimize_result(
C:\Users\user\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py:4
60: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://sciki
t-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regres
sion (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regr
ession)
  n_iter_i = _check_optimize_result(

| Model   |   CV F1 Score |   Model F1 Score |   CV F1 Weighted |   Model F1
Weighted |
|---------+---------------+------------------+------------------+------------
---------|
| LR      |     0.0298246 |       0.0207254  |         0.836231 |
0.937134 |
| LDA     |     0.073835  |       0.0664506  |         0.840131 |
0.929452 |
| RF      |     0.026061  |       0.0258621  |         0.835748 |
0.937172 |
| NB      |     0.211187  |       0.208157   |         0.823906 |
0.822986 |
| AB      |     0.0239732 |       0.0241796  |         0.835524 |
0.937356 |
| GBM     |     0.0198089 |       0.00788782 |         0.835165 |
0.938838 |
| ET      |     0.0272836 |       0.0190229  |         0.83582  |
0.937246 |
| XG      |     0.0425175 |       0.0372881  |         0.83716  |
0.935025 |
| LG      |     0.0185404 |       0.00876808 |         0.835031 |
0.938967 |
| CAT     |     0.0308211 |       0.0215703  |         0.836254 |
0.937042 |
```

In [ ]: