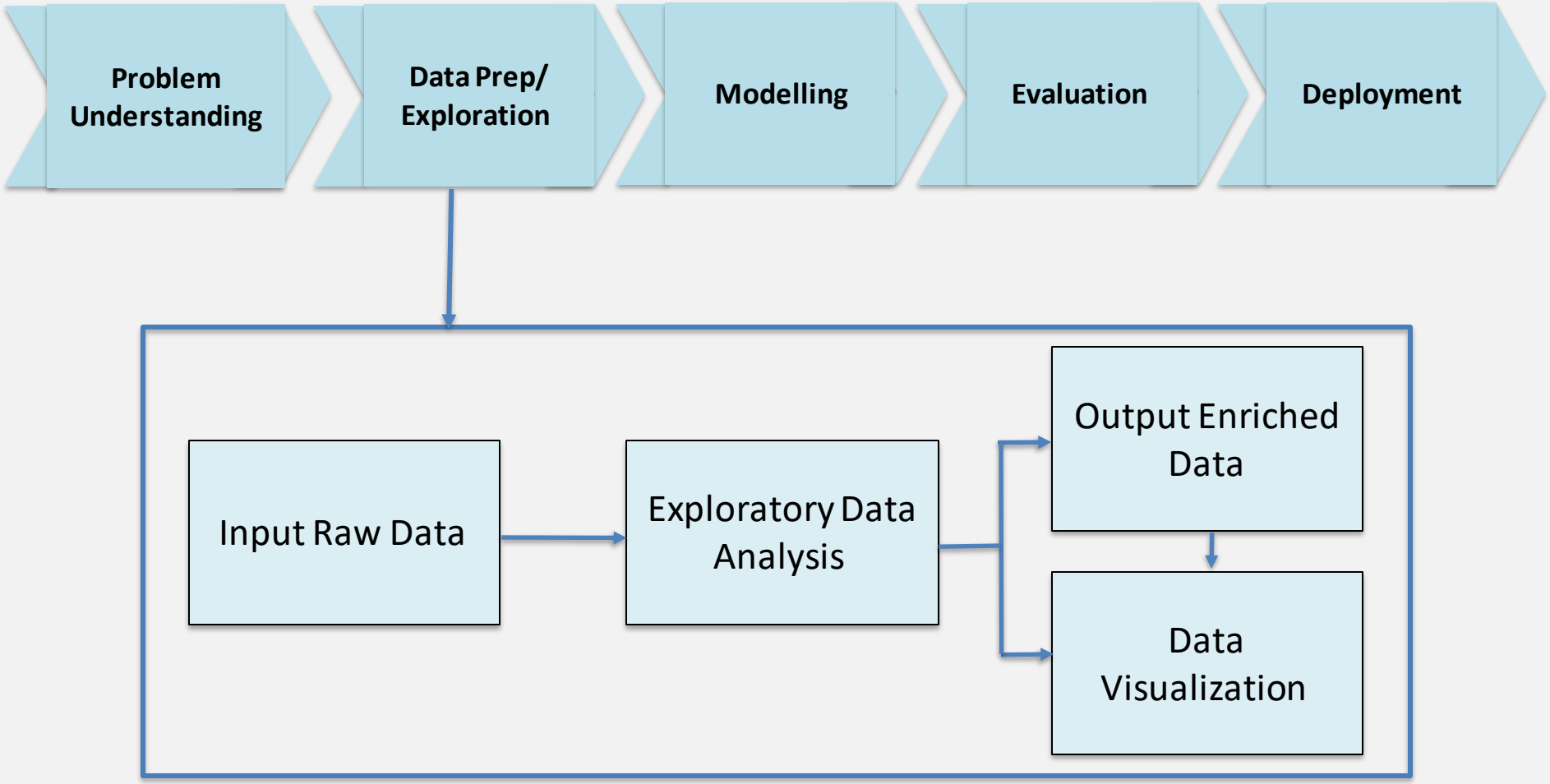
H1N1 Vaccine Predicition Journey

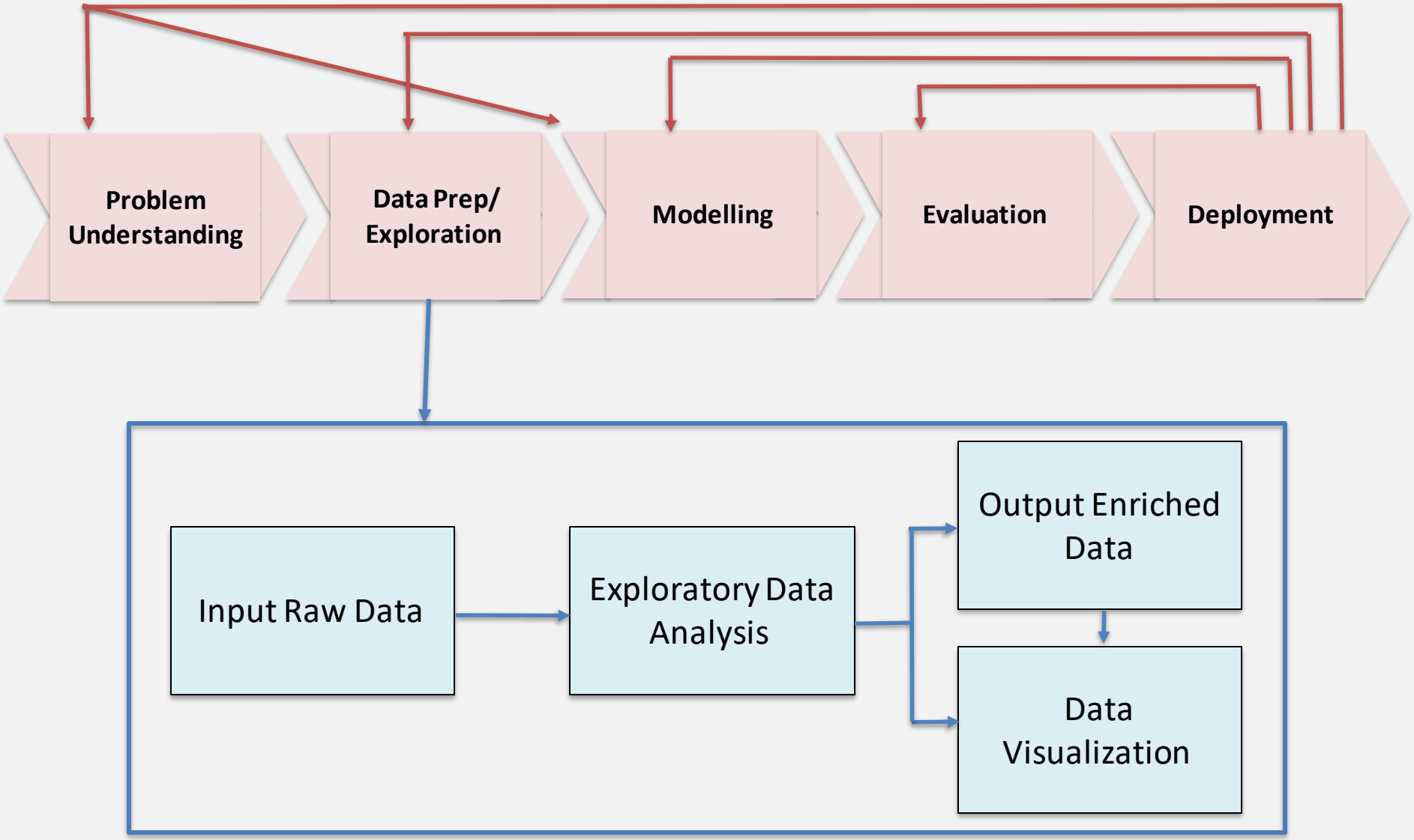
Team Woodbine

Smith SCHOOL OF BUSINESS | Queen's University

The Process

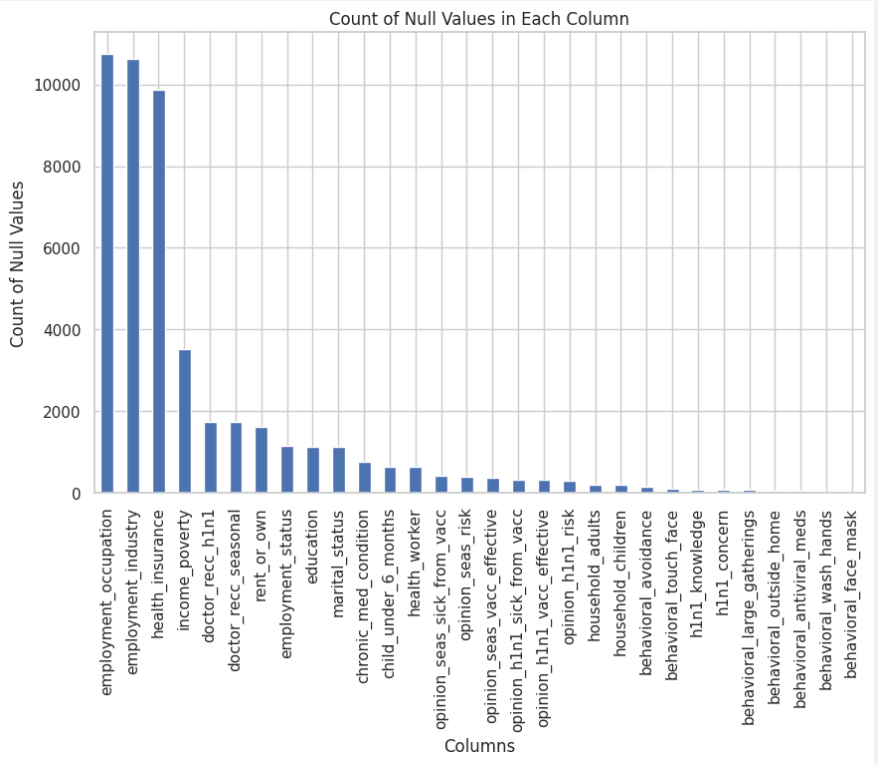

Problem Understanding
Data Prep/ Exploration
Modelling
Evaluation
Deployment
Input Raw Data
Exploratory Data Analysis
Output Enriched Data
Data Visualization

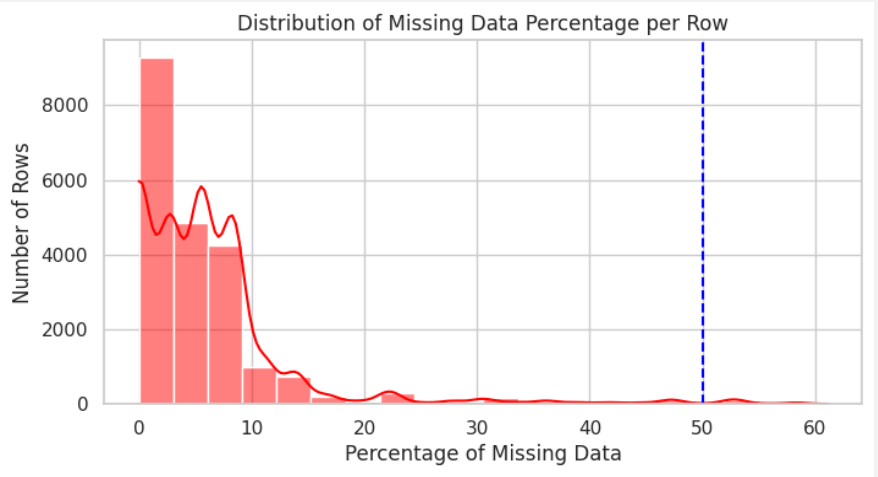
Reality


Problem Understanding
Data Prep/ Exploration
Modelling
Evaluation
Deployment
Input Raw Data
Exploratory Data Analysis
Output Enriched Data
Data Visualization

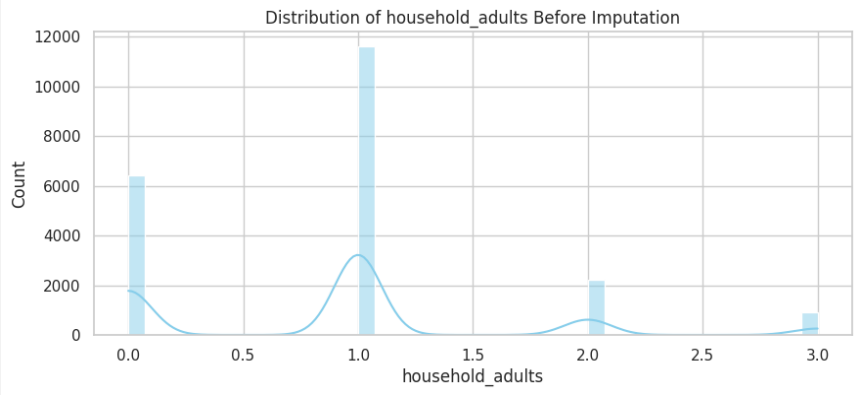
Cleaning and Preprocessing

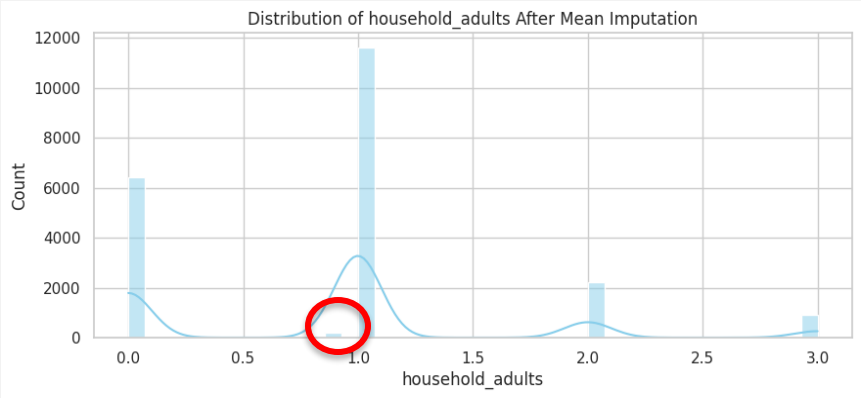
Drops	LGBM
	Training F1 Score
All Categorical	0.62
Instances Missing ? 50%	0.68
Instances Missing ? 55%	0.65
Features Missing > 8000	0.64

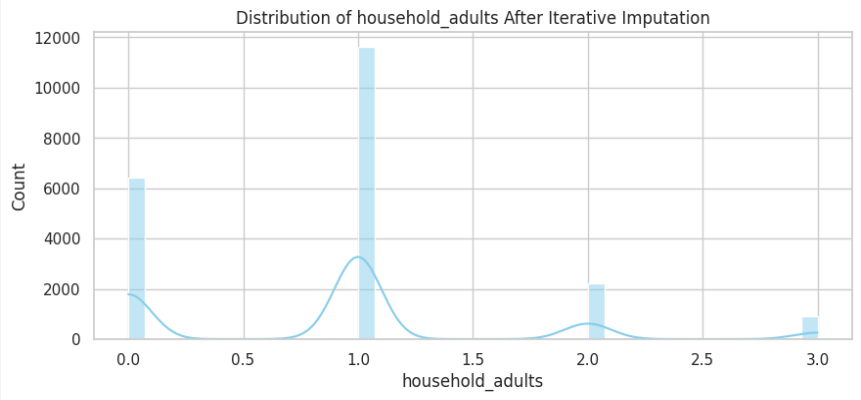




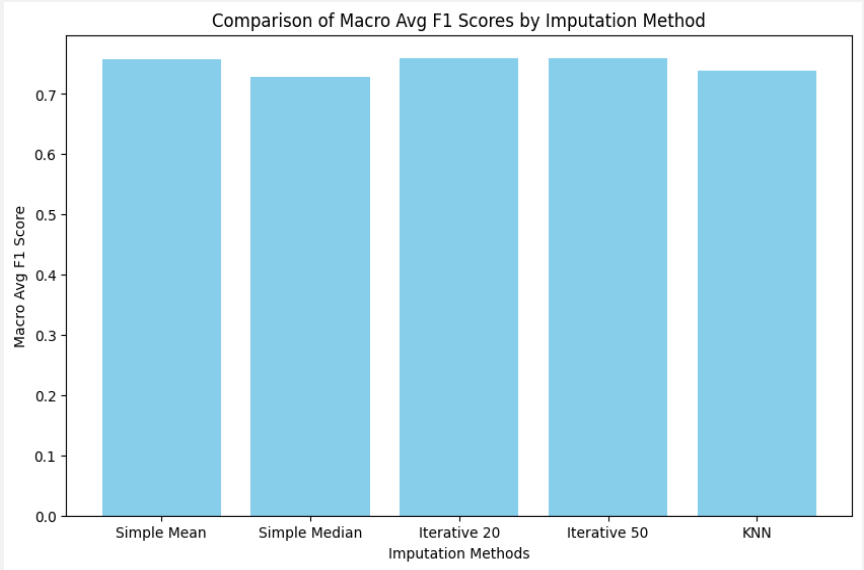
Cleaning and Preprocessing: Numerical Imputation







Encoder	Training F1 Score LGBM
Simple Mean	0.758113
Simple Median	0.727678
Iterative (20)	0.759903
Iterative (50)	0.759003
KNN imputer K =5	0.739303



Cleaning and Preprocessing: Categorial Imputation

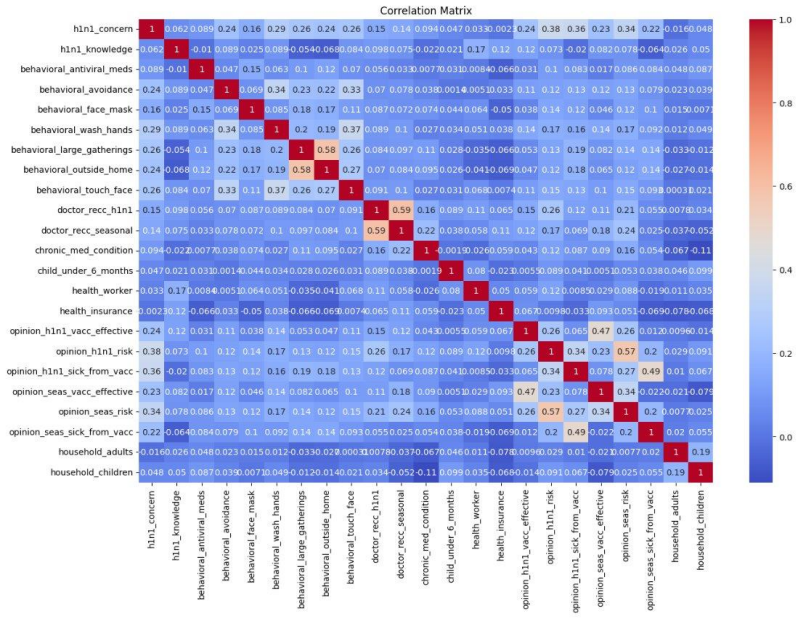
Comparison of encoding methods:

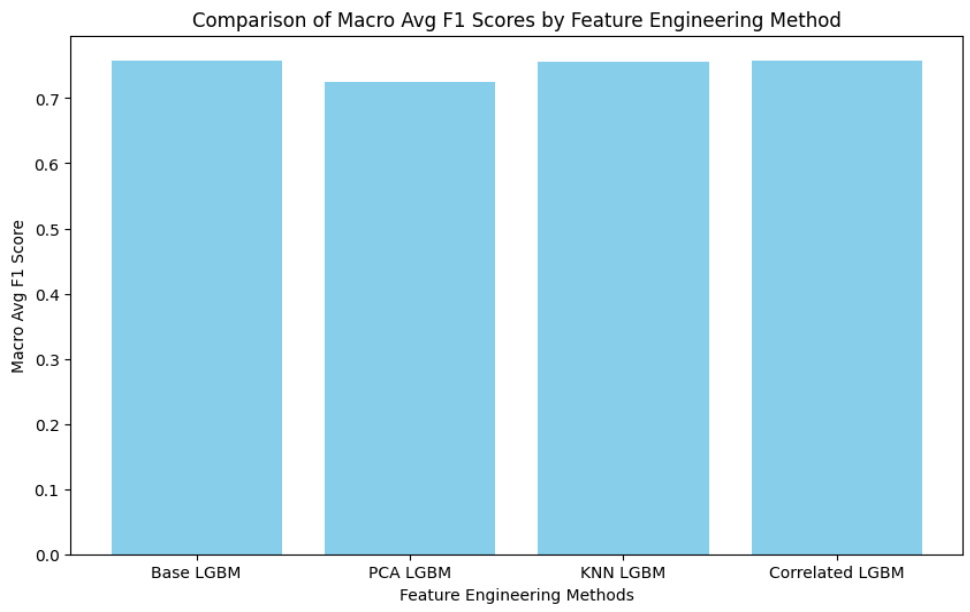
	Original	Label Encoded	Target Encoded	education_< 12 Years	education_College Graduate	education_Some College	education_nan
0	Some College	3	0.211613	0.0	0.0	1.0	0.0
1	College Graduate	2	0.245938	0.0	1.0	0.0	0.0
2	NaN	4	0.184000	0.0	0.0	0.0	1.0
3	College Graduate	2	0.245938	0.0	1.0	0.0	0.0
4	College Graduate	2	0.245938	0.0	1.0	0.0	0.0
5	College Graduate	2	0.245938	0.0	1.0	0.0	0.0
6	12 Years	0	0.183563	0.0	0.0	0.0	0.0
7	College Graduate	2	0.245938	0.0	1.0	0.0	0.0
8	College Graduate	2	0.245938	0.0	1.0	0.0	0.0
9	12 Years	0	0.183563	0.0	0.0	0.0	0.0

Encoder	Training F1 Score LGBM
Ordinal	0.732
Target	0.764
Label	0.762
One Hot	0.735

- Different Performances Based Different Models
- Best Models Shown for each

Feature Engineering and Selection





	Post FE: Training F1 Score	Leader Board Score
Combined Features based on Correlation Matrix	0.758	0.7426
Only using PCA Features	0.724	0.7253
Only using KNN features	0.755	0.7409

ML Algorithms Tried

Model	Numerical/ Categorial Encoders	Training CV F1 Score
Gradient Boost	• Simple Mean, Median, Iterative, KNN • One hot, Target, Label, Ordinal	• 0.757282
XGBoost	• Simple Mean, Median, Iterative, KNN • One hot, Target, Label, Ordinal	• 0.749845
LightGBM	• Iterative, • Target, Label	• 0.758113
Random Forest	• Simple Mean • Target, Label	• 0.724851
KNN	• Simple Mean • Target	• 0.685706

1) LGBM
   - Hyperparameter Tunning is the fastest, More Trials, better tunning.
2) XGBoost
   - Similar performance to LGBM takes longer to train parameters
3) Gradient Boost
   - Performed well, but LGBM outperformed with same tunning, encoding, much slower
4) Random Forest
   - Slower learning times, tunning takes significant time, Similar performance to Gradient Boost
5) KNN
   - Does not perform well

Hyperparameter Tuning

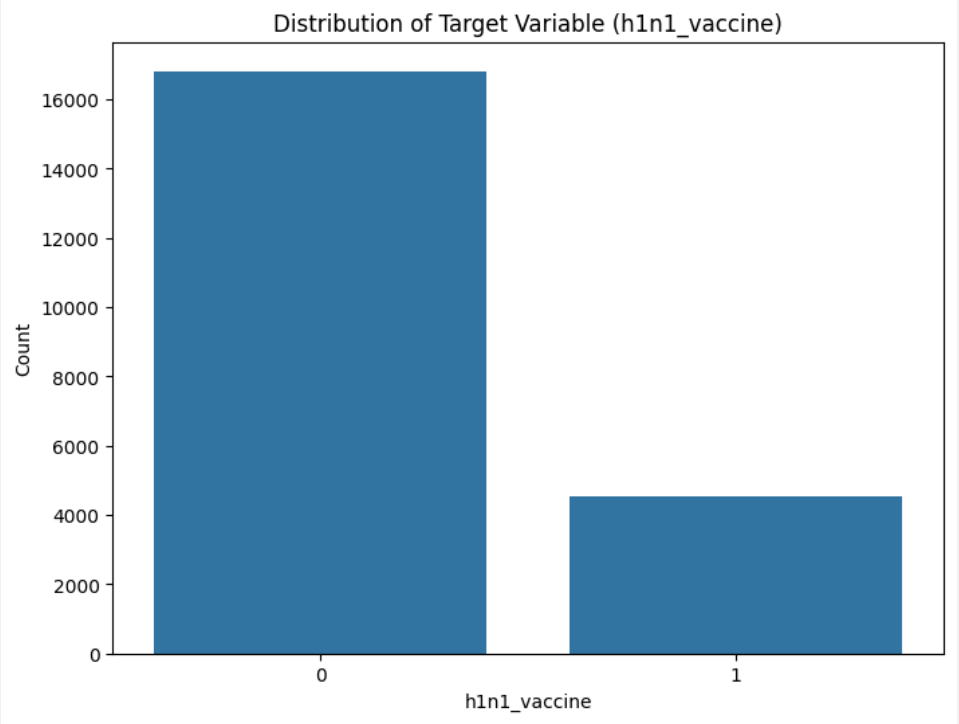
Model	Tuning	Trial/ Ranges	Encoder	Training F1	Leader Board F1
Gradient Boost	GridSearchCV	Estimators: (100-1000) Learning_rate: (0.1-0.5) max_depth: (3-15)	One Hot Simple Mean	0.7378	0.7253
XGBoost	GridSearchCV	Estimators: (100-1000) Learning_rate: (0.1-0.5) max_depth: (3-15)	One Hot Target Iterative	0.7550	0.7409
LightGBM	GridSearchCV	Estimators: (100-1000) Learning_rate: (0.1-0.5) max_depth: (3-15)	One Hot Target Label ordinal Iterative Simple mean	0.7551	0.7523
Random Forest	GridSearchCV	Estimators: (100-1000) Learning_rate: (0.1-0.5) max_depth: (3-15)	Target Simple mean	0.7248	X
XGBoost	Optuna	'learning_rate': (0.01, 0.1), 'max_depth': (5, 50), 'n_estimators': (500, 1200), 'num_leaves': (50, 160), 'min_data_in_leaf': (10, 50), 'feature_fraction': (0.1, 0.9), 'bagging_fraction': (0.1, 0.9), 'bagging_freq': (1, 8), 'lambda_l1': (0.00, 50), 'lambda_l2': (0.00, 50), 'min_split_gain: (0.1, 0.5)	Target Iterative	0.7611	0.7581
LightGBM	Optuna		Target Iterative	0.7655	0.7620


gif-finder.com

Eureka

Class Imbalance Impact:

- The model performs much better on the majority class (0) compared to the minority class (1).
- Precision, recall, and F1-score are significantly lower for class 1, indicating that the model struggles to correctly identify vaccinated individuals.

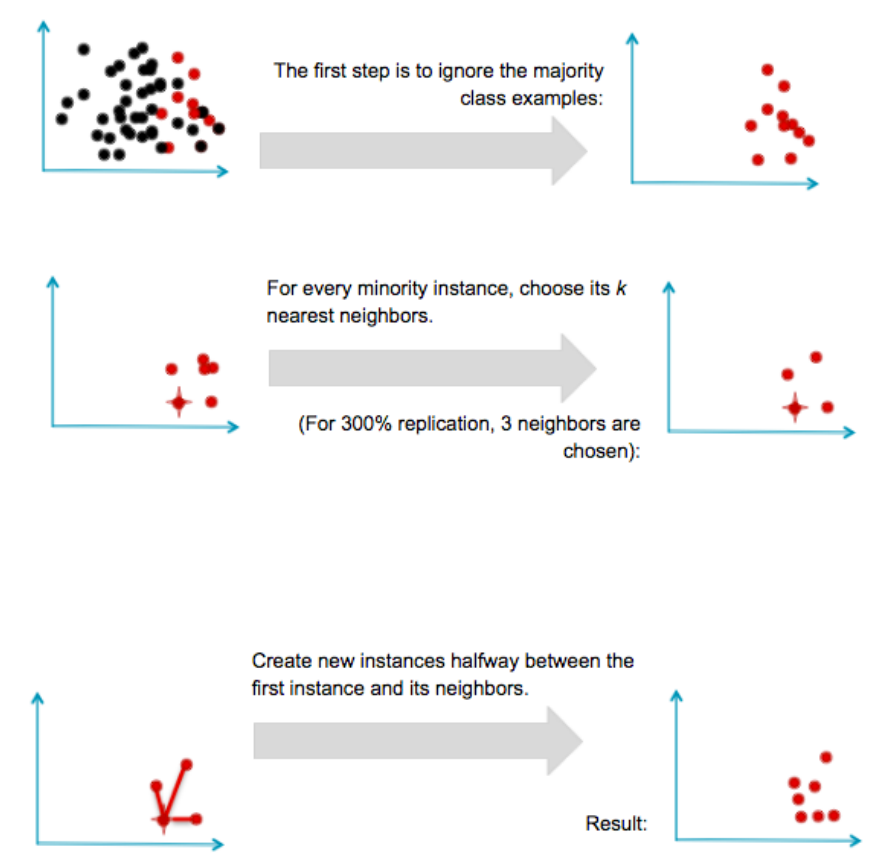


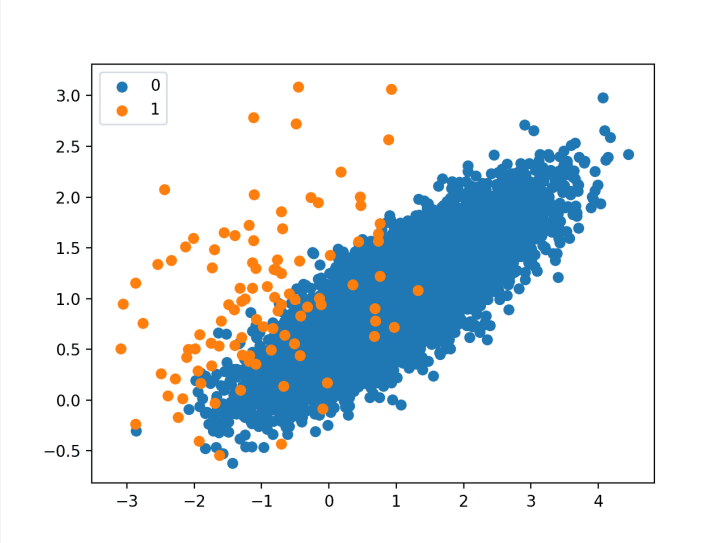
Basic Classifier Detailed Classification Report

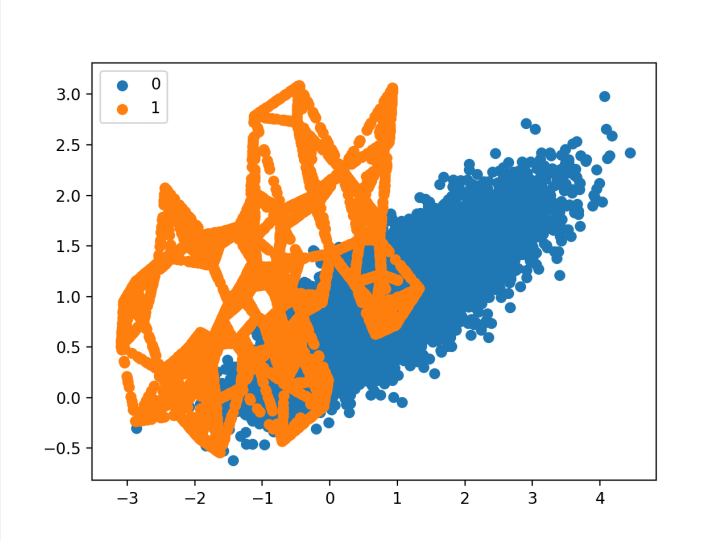
Metric	Class 0	Class 1	Macro Avg	Weighted Avg
Precision	0.858	0.680	0.769	0.820
Recall	0.946	0.421	0.684	0.835
F1-Score	0.900	0.520	0.710	0.819
Support	3364	909	-	4273

SMOTE
(Synthetic Minority Oversampling Technique)

- Analyzes the target variable to identify the minority class.
- Generates synthetic samples to balance the minority class with the majority class.

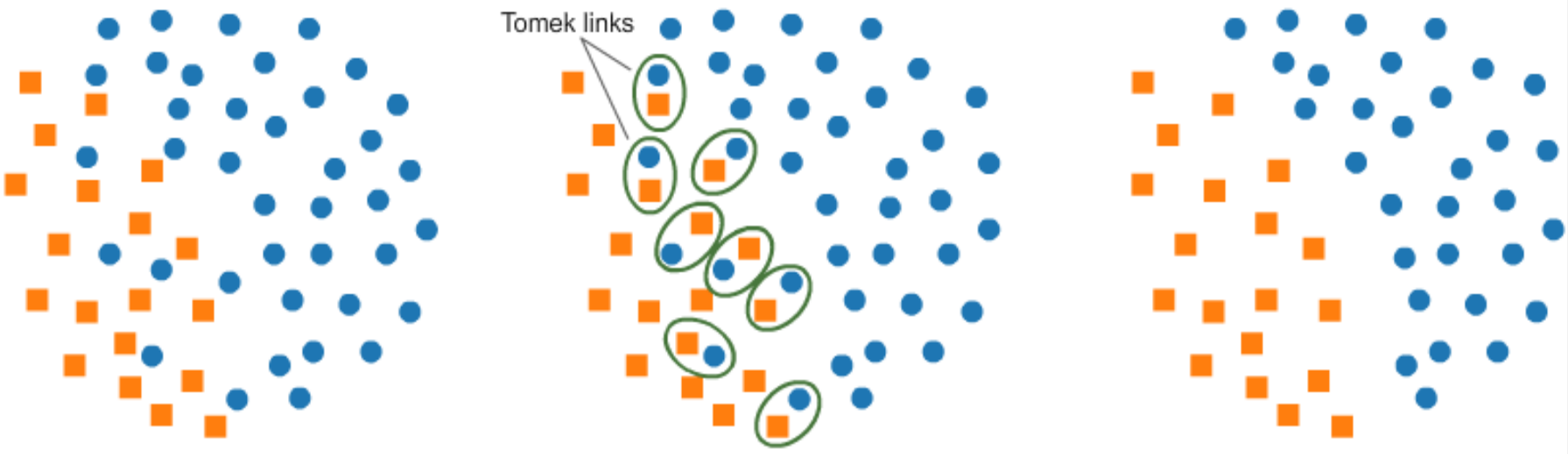




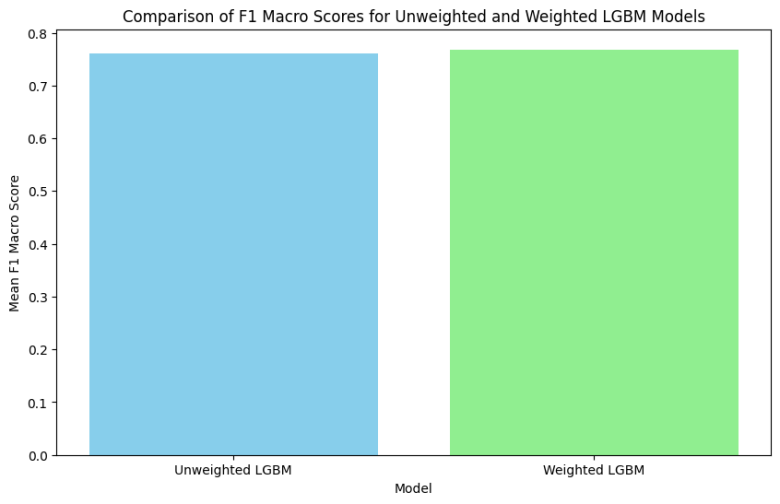


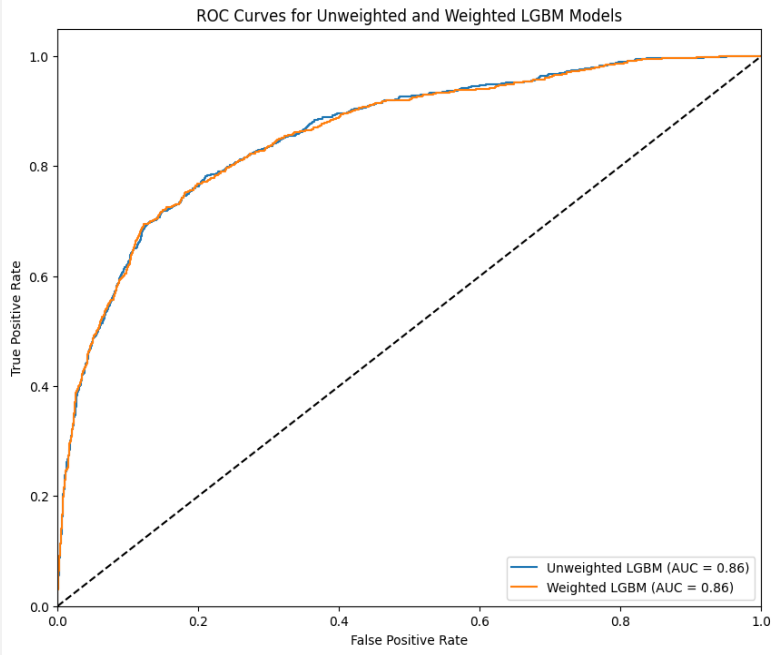
TOME K Links – Under Sampling

1) Identify Nearest Neighbors:
    - For each sample in the dataset, find its nearest neighbor (usually using Euclidean distance).
2) Check for Tomek Links: A pair of samples forms a Tomek Link if:
    - They are from different classes.
    - They are each other's nearest neighbors.
3) Remove Tomek Links:



Weighting



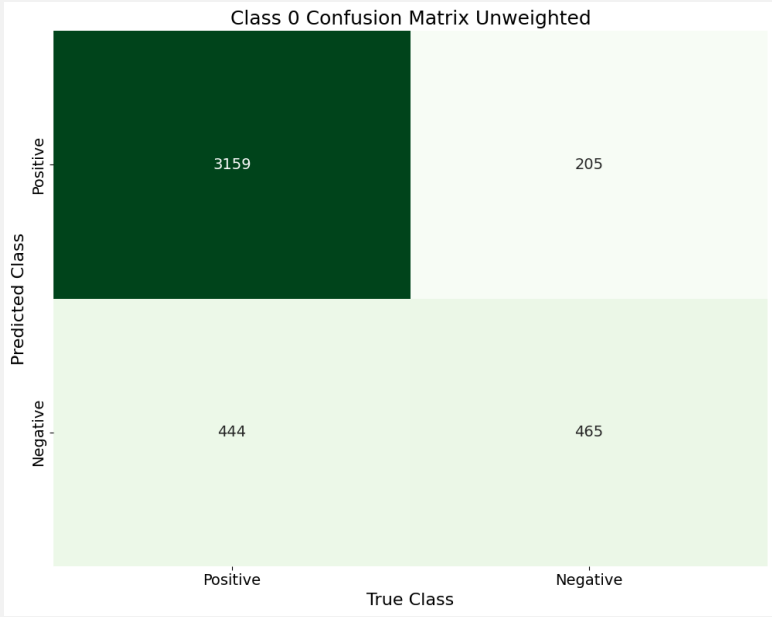


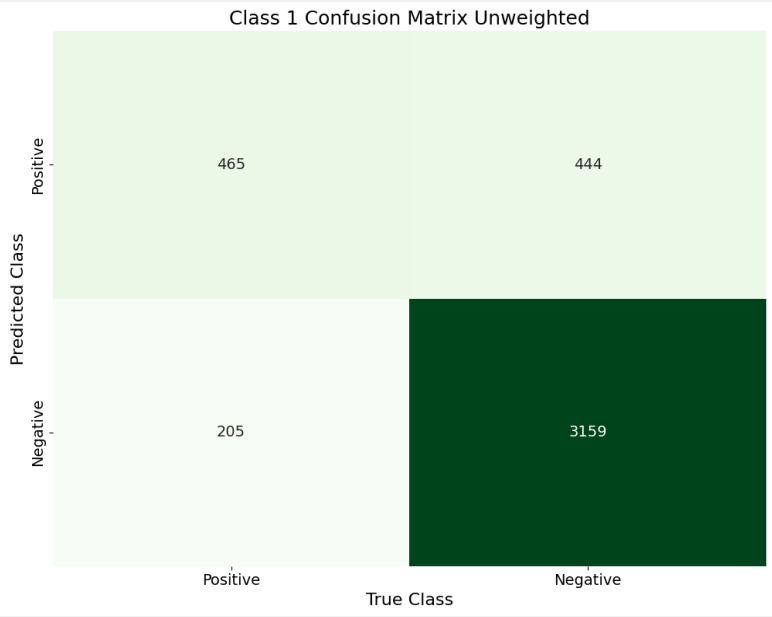

```
Initial Test Set F1 Macro Score unweighted: 0.747913464843263
Confusion Matrix for un weighted:
[[3159  205]
 [ 444  465]]
Test F1 Macro Score and CLA report unweighted: 0.747913464843263
              precision    recall  f1-score   support

           0       0.88      0.94      0.91      3364
           1       0.69      0.51      0.59       909

    accuracy                           0.85      4273
   macro avg       0.79      0.73      0.75      4273
weighted avg       0.84      0.85      0.84      4273

Initial Test Set F1 Macro Score weighted: 0.7683447484228735
Confusion Matrix for weighted:
[[2987  377]
 [ 305  604]]
Test F1 Macro Score and CLA report weighted: 0.7683447484228735
              precision    recall  f1-score   support

           0       0.91      0.89      0.90      3364
           1       0.62      0.66      0.64       909

    accuracy                           0.84      4273
   macro avg       0.76      0.78      0.77      4273
weighted avg       0.85      0.84      0.84      4273
```

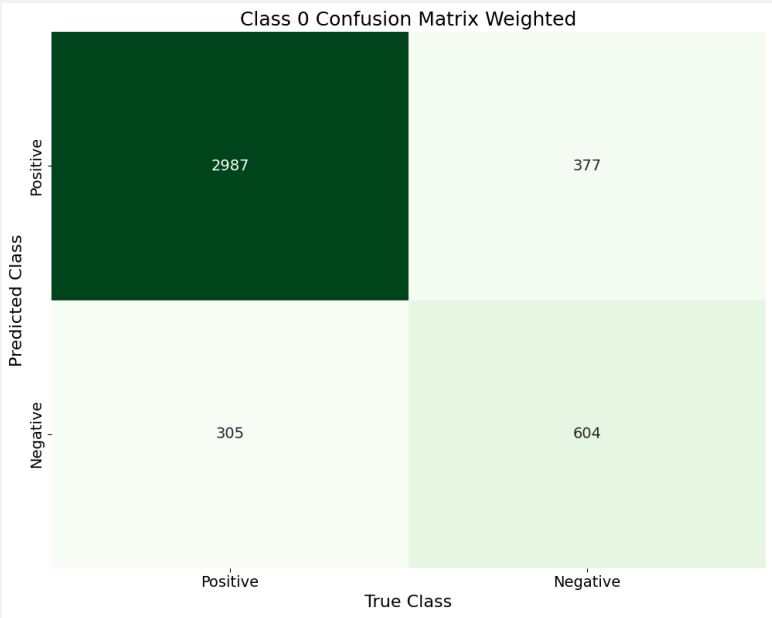
Tested Weight Class:

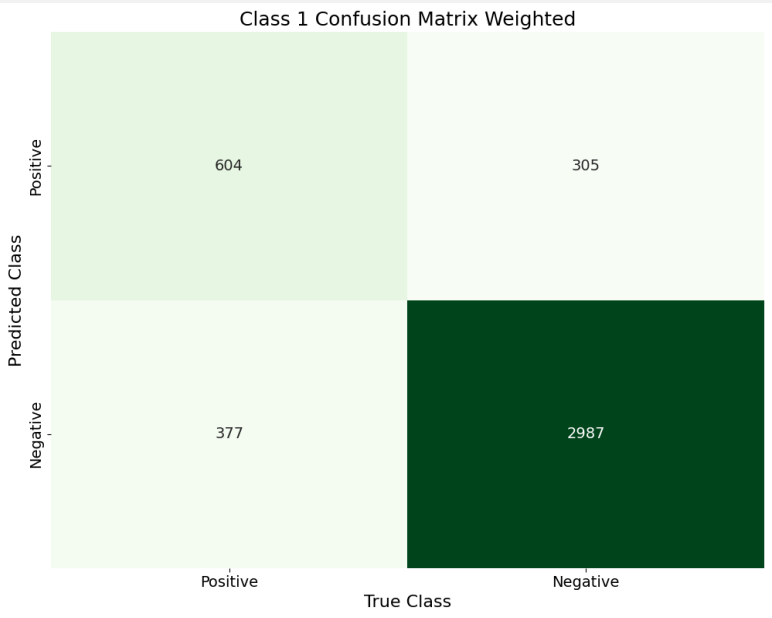
- 0:1, 1:2 ★
- 0:1, 1:3
- 0:1, 1:3.7 (Ratio)
- 0:1, 1:7

Weighting + SMOTE

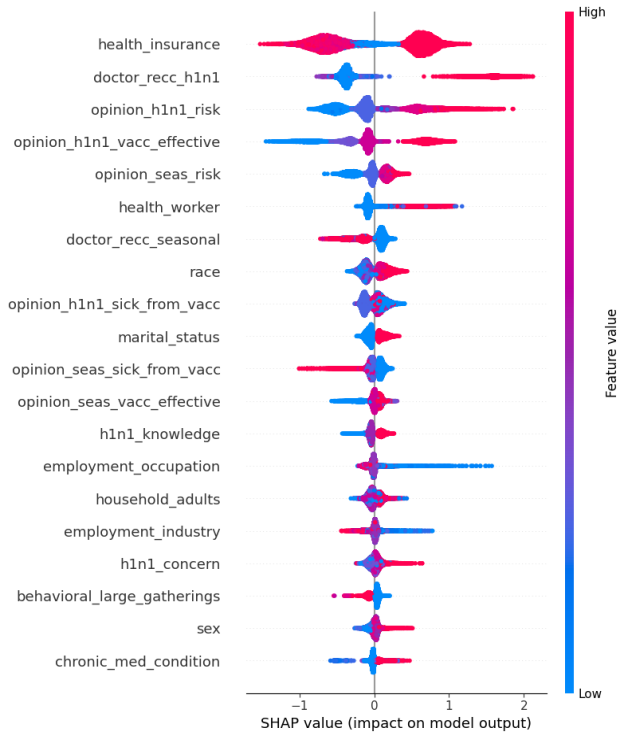


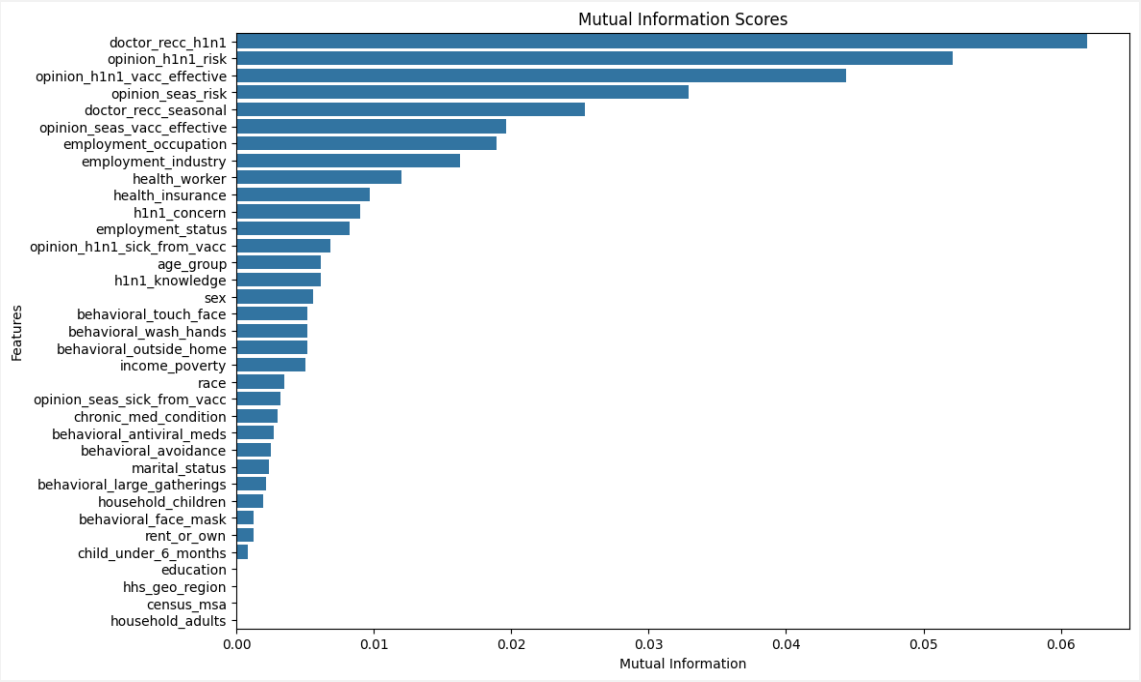






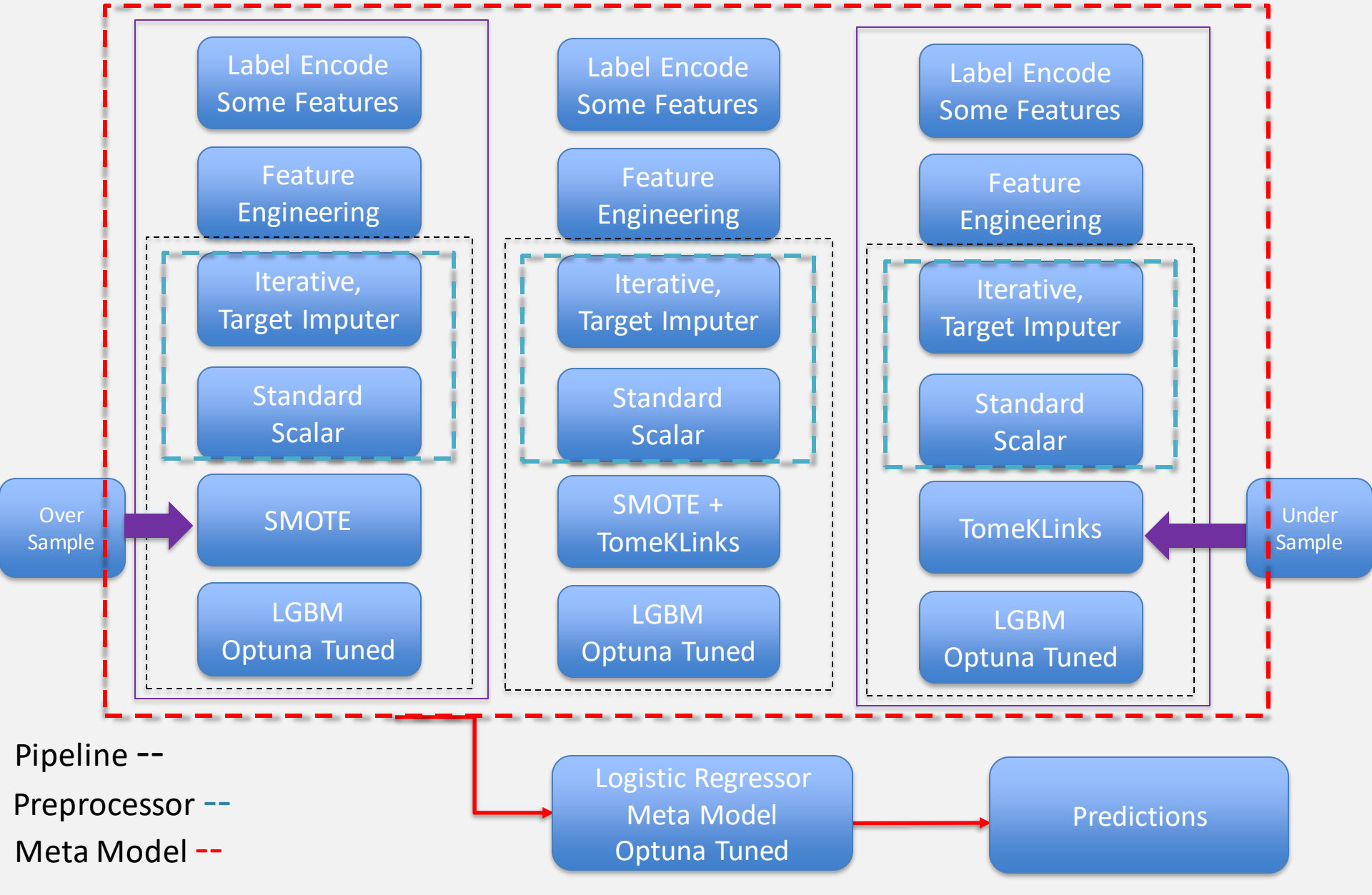
Feature Engineering and Selection






```
Cross-Validation F1 Macro Score without dropping features: 0.7586606275365219
Cross-Validation F1 Macro Score with dropping features: 0.7572252618782824
Keeping all features performs better.
Features: [], Remove Original: False, Cross-Validation F1 Macro Score: 0.7605133019145823
Features: [], Remove Original: True, Cross-Validation F1 Macro Score: 0.7605133019145823
Features: ['health_insurance_doctor_recc_h1n1'], Remove Original: False, Cross-Validation F1 Macro Score: 0.7611054851089655
Features: ['health_insurance_doctor_recc_h1n1'], Remove Original: True, Cross-Validation F1 Macro Score: 0.7516396701226415
Features: ['doctor_recc_h1n1_opinion_h1n1_risk'], Remove Original: False, Cross-Validation F1 Macro Score: 0.7625780436222299
Features: ['doctor_recc_h1n1_opinion_h1n1_risk'], Remove Original: True, Cross-Validation F1 Macro Score: 0.7532443930788658
Features: ['opinion_h1n1_risk_opinion_h1n1_vacc_effective'], Remove Original: False, Cross-Validation F1 Macro Score: 0.7600686555708033
Features: ['opinion_h1n1_risk_opinion_h1n1_vacc_effective'], Remove Original: True, Cross-Validation F1 Macro Score: 0.7628582147704284
Features: ['health_insurance_doctor_recc_h1n1', 'doctor_recc_h1n1_opinion_h1n1_risk'], Remove Original: False, Cross-Validation F1 Macro Score: 0.7597311847826884
Features: ['health_insurance_doctor_recc_h1n1', 'doctor_recc_h1n1_opinion_h1n1_risk'], Remove Original: True, Cross-Validation F1 Macro Score: 0.7494912731653032
Features: ['health_insurance_doctor_recc_h1n1', 'opinion_h1n1_risk_opinion_h1n1_vacc_effective'], Remove Original: False, Cross-Validation F1 Macro Score: 0.7611451893893445
Features: ['health_insurance_doctor_recc_h1n1', 'opinion_h1n1_risk_opinion_h1n1_vacc_effective'], Remove Original: True, Cross-Validation F1 Macro Score: 0.7514110934744335
Features: ['doctor_recc_h1n1_opinion_h1n1_risk', 'opinion_h1n1_risk_opinion_h1n1_vacc_effective'], Remove Original: False, Cross-Validation F1 Macro Score: 0.762447223151291
Features: ['doctor_recc_h1n1_opinion_h1n1_risk', 'opinion_h1n1_risk_opinion_h1n1_vacc_effective'], Remove Original: True, Cross-Validation F1 Macro Score: 0.7592102099162398
Features: ['health_insurance_doctor_recc_h1n1', 'doctor_recc_h1n1_opinion_h1n1_risk', 'opinion_h1n1_risk_opinion_h1n1_vacc_effective'], Remove Original: False, Cross-Validation F1 Macro Score: 0.7612858864561272
Features: ['health_insurance_doctor_recc_h1n1', 'doctor_recc_h1n1_opinion_h1n1_risk', 'opinion_h1n1_risk_opinion_h1n1_vacc_effective'], Remove Original: True, Cross-Validation F1 Macro Score: 0.753837803911815
Best Combination: ['opinion_h1n1_risk_opinion_h1n1_vacc_effective', '(without originals)'], Best Cross-Validation F1 Macro Score: 0.7628582147704284
```

Best Model and Performance



Final Performance

Based on Our Final Score of 0.7727
The main predictors of if the H1N1 vaccination Rates:

- Doctor recommendations and opinions on effectiveness
- Health Insurance

We would put this model into production:

Training Cross Validation Score

```
Best score: 0.774312614386875
Stacking Classifier F1 Macro Score: 0.7663145793595496
Test F1 Macro Score: [0 1 0 ... 0 0 0]
              precision    recall  f1-score   support

           0       0.90      0.90      0.90      3364
           1       0.63      0.63      0.63       909

    accuracy                           0.84      4273
   macro avg       0.77      0.77      0.77      4273
weighted avg       0.84      0.84      0.84      4273
```

Leaderboard Score

0.7727

Lesson's Learned

- Spend More time on Feature Analysis and EDA
  - Understand the whole dataset
    - Imbalances and how to deal with them !
  - Interdependencies and feature interactions matter
    - Highlight them for the model = better performance
- TUNE! TUNE! TUNE!
  - Hyperparameter's for everything
  - Change anything tune again
- Experiment: Models, Encoders, Feature Engineering

Next Steps

Future Work:

- Run with Mutiple Models: Random Forest, XGBoost, AdaBoost, CatBoost, Gradient Boost.
- Ensemble and Tune Meta Model.
  - Run Multiple models with under and over sampling in a meta model.
  - Use more complex meta models.
- Consider SVM for Data analysis and Feature engineering. (Further Research)
  - Health Insurance has the largest null values but is also the largest predictor.
- SMOT ENN.
- Run SFS and SBS with feature interactions and low feature importance, to find details which may have been missed.
- Use the weight parameter in LGBM and Tune

APPENDIX

Cleaning and Preprocessing: Numerical Imputation

Classification Report for Simple Mean:

	precision	recall	f1-score	support
0	0.878123	0.943897	0.909823	3351
1	0.719821	0.523861	0.606403	922
accuracy	0.853265	0.853265	0.853265	0.853265
macro avg	0.798972	0.733879	0.758113	4273
weighted avg	0.843966	0.853265	0.844353	4273

Classification Report for Simple Median:

	precision	recall	f1-score	support
0	0.865931	0.936735	0.899943	3351
1	0.67284	0.472885	0.555414	922
accuracy	0.836649	0.836649	0.836649	0.836649
macro avg	0.769385	0.70481	0.727678	4273
weighted avg	0.824267	0.836649	0.825603	4273

Classification Report for Iterative 20:

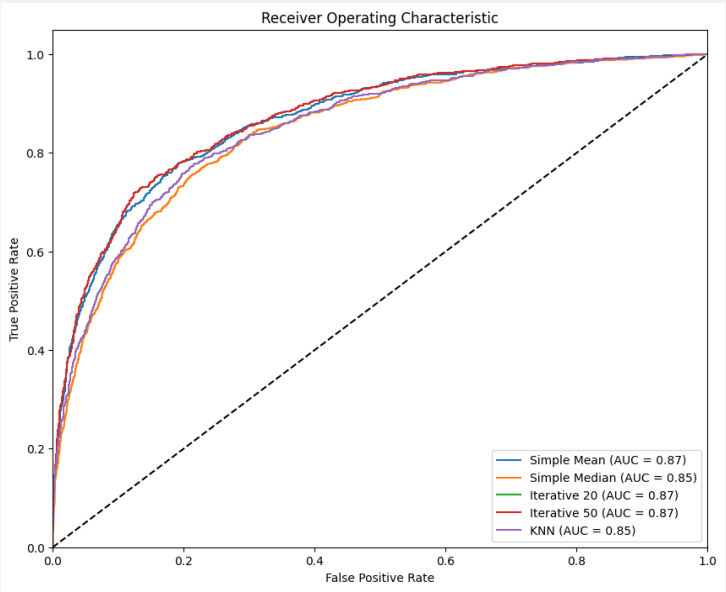
	precision	recall	f1-score	support
0	0.876029	0.953148	0.912963	3351
1	0.749601	0.509761	0.606843	922
accuracy	0.857477	0.857477	0.857477	0.857477
macro avg	0.812815	0.731455	0.759903	4273
weighted avg	0.848749	0.857477	0.84691	4273

Classification Report for Iterative 50:

	precision	recall	f1-score	support
0	0.876029	0.953148	0.912963	3351
1	0.749601	0.509761	0.606843	922
accuracy	0.857477	0.857477	0.857477	0.857477
macro avg	0.812815	0.731455	0.759903	4273
weighted avg	0.848749	0.857477	0.84691	4273

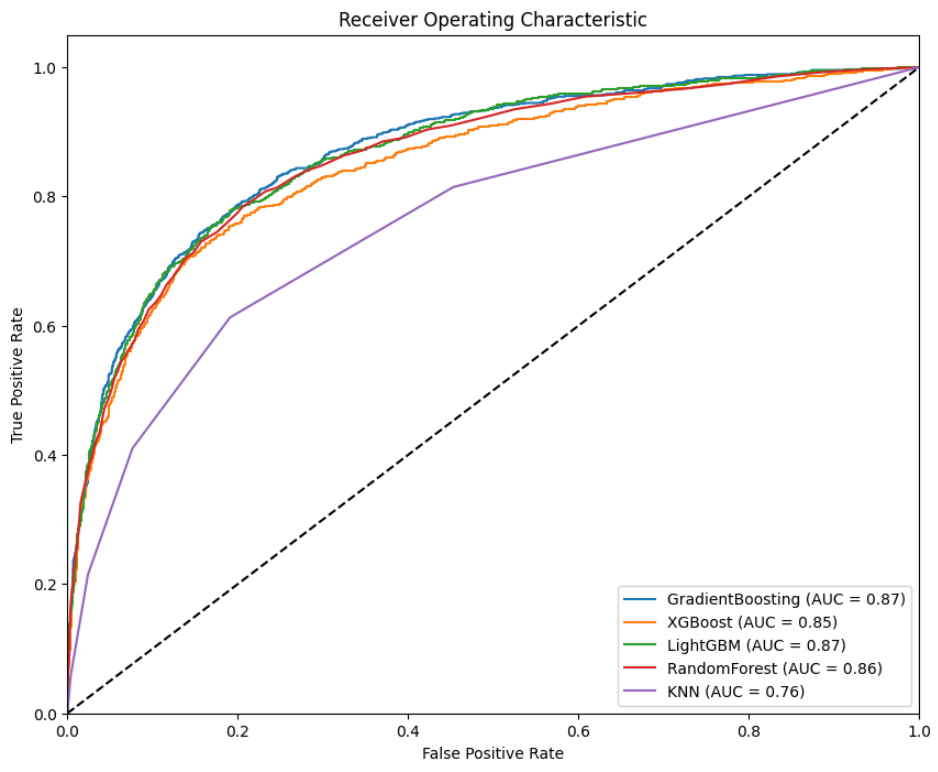
Classification Report for KNN:

	precision	recall	f1-score	support
0	0.872109	0.93405	0.902017	3351
1	0.676901	0.502169	0.576588	922
accuracy	0.840861	0.840861	0.840861	0.840861
macro avg	0.774505	0.718109	0.739303	4273
weighted avg	0.829988	0.840861	0.831798	4273

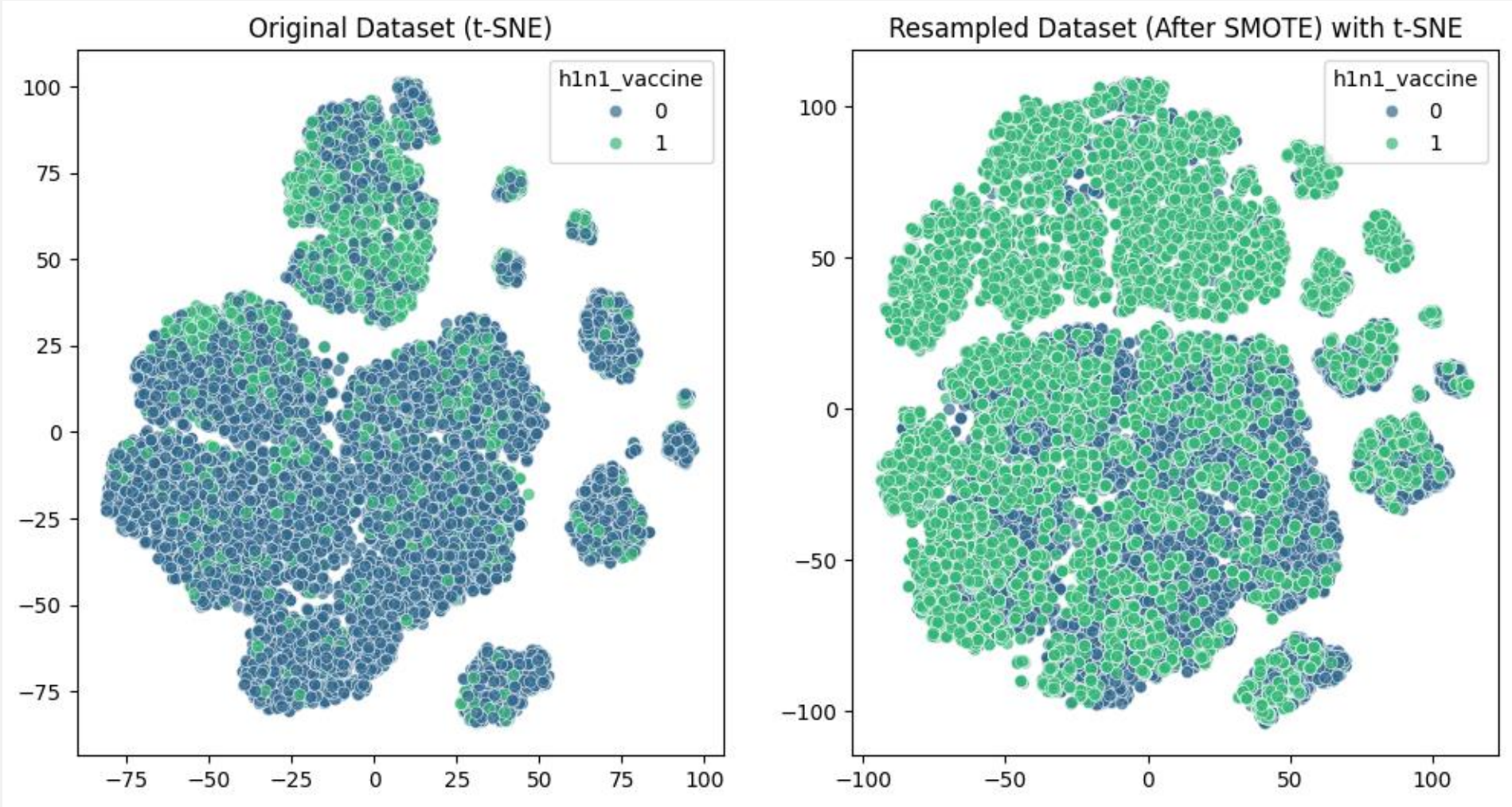


ML Algorithms Tried

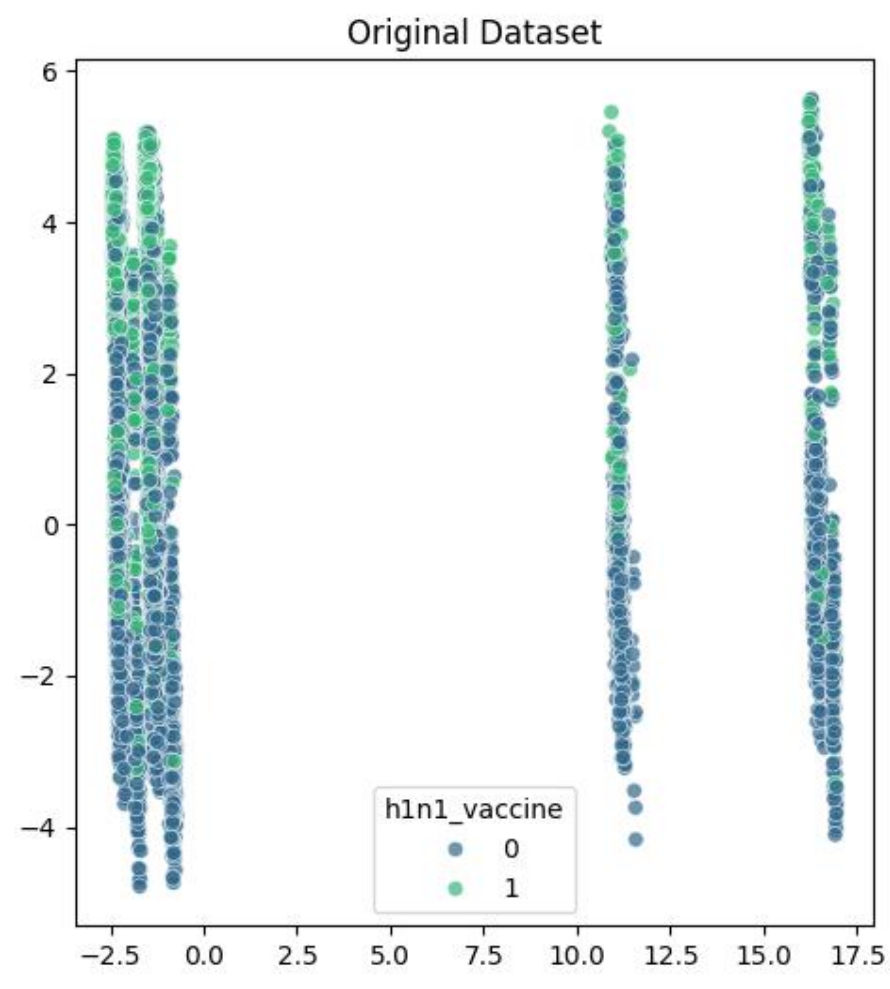
```
Classification Report for GradientBoosting:
              precision    recall  f1-score      support
0              0.873266  0.958221  0.913773  3351.000000
1              0.765101  0.494577  0.600791   922.000000
accuracy       0.858179  0.858179  0.858179     0.858179
macro avg      0.819183  0.726399  0.757282  4273.000000
weighted avg   0.849927  0.858179  0.846240  4273.000000
Classification Report for XGBoost:
              precision    recall  f1-score      support
0              0.875452  0.939719  0.906448  3351.000000
1              0.701183  0.514100  0.593242   922.000000
accuracy       0.847882  0.847882  0.847882     0.847882
macro avg      0.788318  0.726910  0.749845  4273.000000
weighted avg   0.837849  0.847882  0.838866  4273.000000
Classification Report for LightGBM:
              precision    recall  f1-score      support
0              0.878123  0.943897  0.909823  3351.000000
1              0.719821  0.523861  0.606403   922.000000
accuracy       0.853265  0.853265  0.853265     0.853265
macro avg      0.798972  0.733879  0.758113  4273.000000
weighted avg   0.843966  0.853265  0.844353  4273.000000
Classification Report for RandomForest:
              precision    recall  f1-score      support
0              0.859129  0.959117  0.906373  3351.000000
1              0.742481  0.428416  0.543329   922.000000
accuracy       0.844606  0.844606  0.844606     0.844606
macro avg      0.800805  0.693767  0.724851  4273.000000
weighted avg   0.833959  0.844606  0.828038  4273.000000
Classification Report for KNN:
              precision    recall  f1-score      support
0              0.850508  0.923605  0.885551  3351.000000
1              0.596215  0.409978  0.485861   922.000000
accuracy       0.812778  0.812778  0.812778     0.812778
macro avg      0.723361  0.666792  0.685706  4273.000000
weighted avg   0.795639  0.812778  0.799308  4273.000000
```

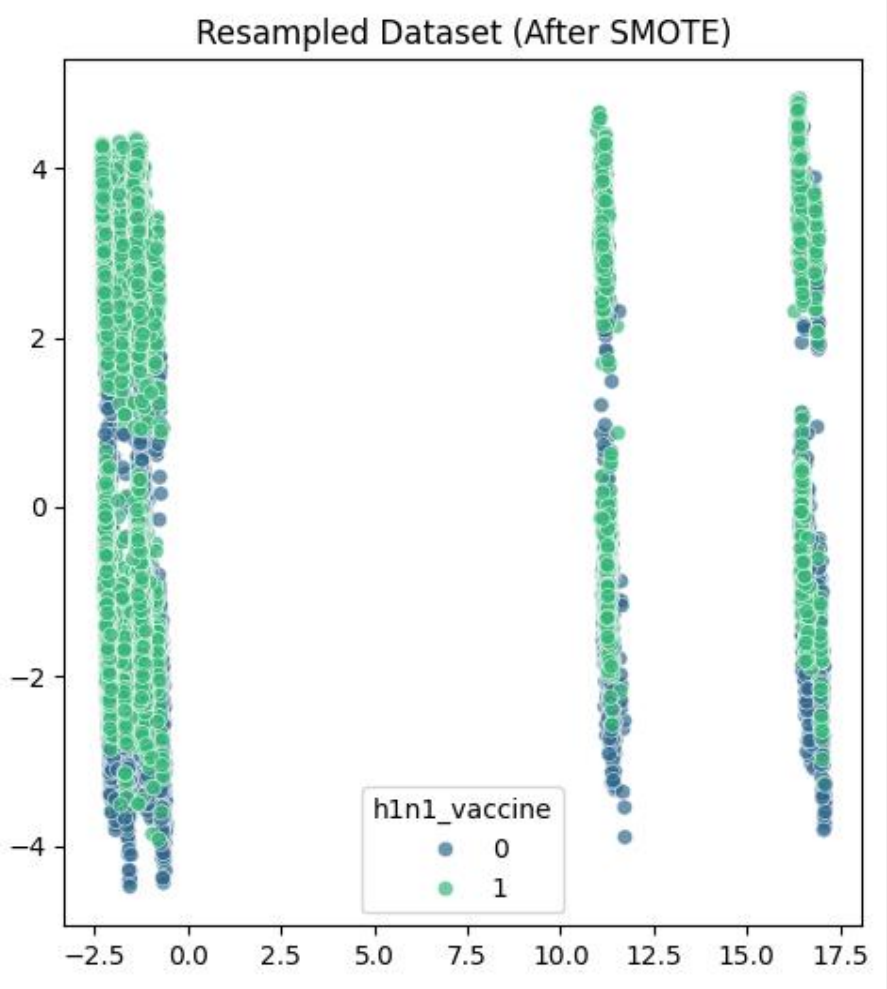


SMOTE


Original Dataset (t-SNE)
h1n1_vaccine
0
1
100
75
50
25
0
−25
−50
−75
−75
−50
−25
0
25
50
75
100
Resampled Dataset (After SMOTE) with t-SNE
h1n1_vaccine
0
1
100
50
0
−50
−100
−100
−50
0
50
100

SMOTE


Original Dataset
h1n1_vaccine
0
1
6
4
2
0
−2
−4
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

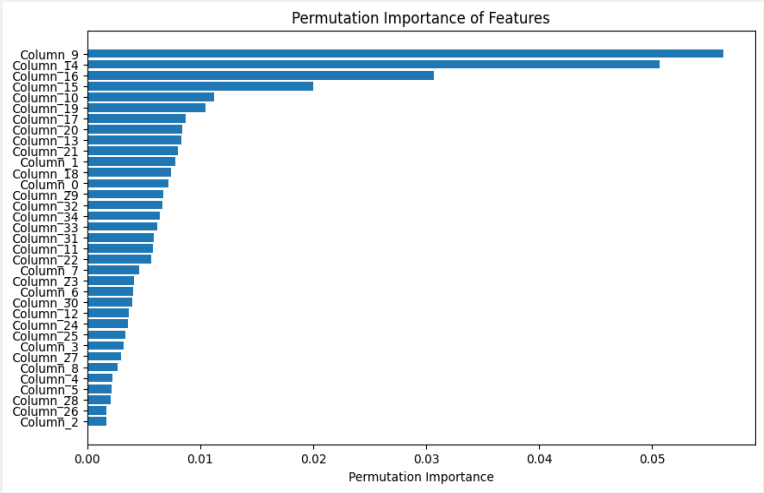

Resampled Dataset (After SMOTE)
h1n1_vaccine
0
1
4
2
0
−2
−4
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

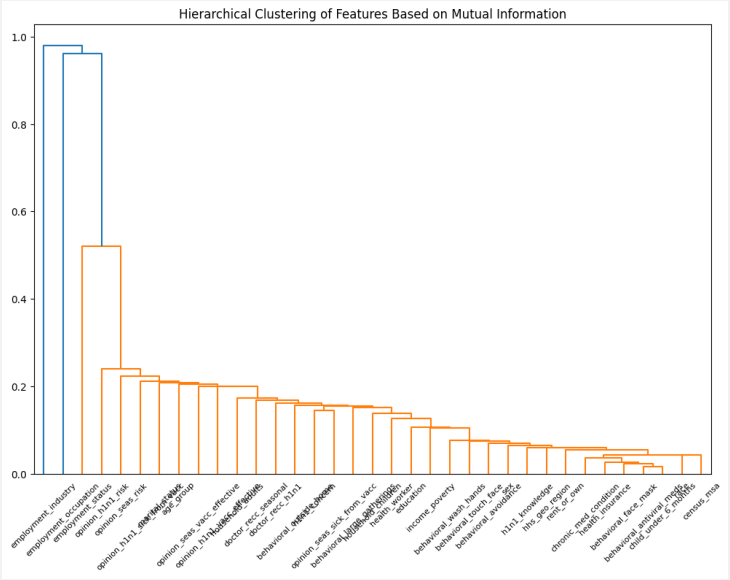
Feature Engineering: Inter-dependencies

	Mean Correlation	Correlation with Target
h1n1_concern	0.139076	0.117341
h1n1_knowledge	0.056581	0.125287
doctor_recc_h1n1	0.167866	0.395725
doctor_recc_seasonal	0.158876	0.212834
chronic_med_condition	0.100429	0.095254
child_under_6_months	0.057094	0.071826
health_worker	0.043071	0.176329
health_insurance	0.048961	0.123325
opinion_h1n1_vacc_effective	0.120494	0.271821
opinion_h1n1_risk	0.164236	0.308568
opinion_seas_vacc_effective	0.121952	0.179892
age_group	0.050924	0.041919
education	0.043374	0.031399
sex	-0.010801	-0.021600
marital_status	0.024131	-0.052433
employment_status	0.092286	-0.027578
hhs_geo_region	0.037658	0.010446
census_msa	0.030670	-0.009963
household_adults	0.020542	0.012916
household_children	0.021941	-0.009112
employment_industry	0.075402	-0.054751
employment_occupation	0.068342	-0.082198
behavioral_combined	0.117826	0.077963
risk_perception	0.171123	0.318286
vaccine_worry	0.097690	0.056825
doctor_recc_combined	0.182654	0.334414
h1n1_vaccine	0.137213	1.000000

Columns with mean correlation <= 0.05:

	Mean Correlation	Correlation with Target
health_worker	0.043071	0.176329
health_insurance	0.048961	0.123325
education	0.043374	0.031399
sex	-0.010801	-0.021600
marital_status	0.024131	-0.052433
hhs_geo_region	0.037658	0.010446
census_msa	0.030670	-0.009963
household_adults	0.020542	0.012916
household_children	0.021941	-0.009112





Feature Engineering and Selection

```python
# Load and preprocess the dataset
df = pd.read_csv("https://drive.google.com/uc?export=download&id=1eYCKuqJda4bpzXBVnqXylg0qQwvpUuum")

# Define feature and target variables
X = df.drop('h1n1_vaccine', axis=1)
y = df['h1n1_vaccine']


# Create new features based on the analysis
X['doctor_opinion_interaction'] = X['doctor_recc_h1n1'] + X['opinion_h1n1_vacc_effective']
X['behavioral_risk_score'] = X[['behavioral_face_mask', 'behavioral_antiviral_meds', 'behavioral_large_gatherings']].sum(axis=1)
```

```
Initial Test Set F1 Macro Score: 0.7656229330156725
Confusion Matrix:
[[3007  357]
 [ 323  586]]
Test F1 Macro Score: 0.7656229330156725
              precision    recall  f1-score   support

           0       0.90      0.89      0.90      3364
           1       0.62      0.64      0.63       909

    accuracy                           0.84      4273
   macro avg       0.76      0.77      0.77      4273
weighted avg       0.84      0.84      0.84      4273
```

```
Initial Test Set F1 Macro Score: 0.7683447484228735
Confusion Matrix:
[[2987  377]
 [ 305  604]]
Test F1 Macro Score: 0.7683447484228735
              precision    recall  f1-score   support

           0       0.91      0.89      0.90      3364
           1       0.62      0.66      0.64       909

    accuracy                           0.84      4273
   macro avg       0.76      0.78      0.77      4273
weighted avg       0.85      0.84      0.84      4273
```

Best Model Parameters

```
# Define the baseline parameters with class weights
class_weight_one = {0: 1, 1: 2}
baseline_params_one = {
    'learning_rate': 0.016616727317418083,
    'max_depth': 7,
    'n_estimators': 900,
    'num_leaves': 142,
    'min_data_in_leaf': 27,
    'feature_fraction': 0.5400049177903017,
    'bagging_fraction': 0.847447849938375 7,
    'bagging_freq': 8,
    'lambda_l1': 3.962383999230598,
    'lambda_l2': 2.288789088218 9534,
    'min_split_gain': 0.2913249686949 2824,
    'verbosity': -1,
    'class_weight': class_weight_one
}
```

Best Model Parameters

```
# Create the base models with updated parameters
model_one = ImbPipeline(steps=[
    ('preprocessor', preprocessor),
    ('smote', SMOTE(random_state=42, sampling_strategy=0.5512254075965978, k_neighbors=2)),
    ('model', lgb.LGBMClassifier(**baseline_params_one))
])

model_two = ImbPipeline(steps=[
    ('preprocessor', preprocessor),
    ('smote_tomek', SMOTETomek(
        smote=SMOTE(sampling_strategy=0.5539439320056955, k_neighbors=5),
        tomek=TomekLinks())),
    ('model', lgb.LGBMClassifier(**baseline_params_one))
])

model_three = ImbPipeline(steps=[
    ('preprocessor', preprocessor),
    ('tomek', TomekLinks()),
    ('model', lgb.LGBMClassifier(**baseline_params_one))
])
```

Final Performance

```
Initial Test Set F1 Macro Score: 0.7656905198433266
Confusion Matrix:
[[2951  413]
 [ 290  619]]
Test F1 Macro Score: 0.7656905198433266
              precision    recall  f1-score   support

           0       0.91      0.88      0.89      3364
           1       0.60      0.68      0.64       909

    accuracy                           0.84      4273
   macro avg       0.76      0.78      0.77      4273
weighted avg       0.84      0.84      0.84      4273
```

```
Initial Test Set F1 Macro Score: 0.7674321067338749
Confusion Matrix:
[[2997  367]
 [ 313  596]]
Test F1 Macro Score: 0.7674321067338749
              precision    recall  f1-score   support

           0       0.91      0.89      0.90      3364
           1       0.62      0.66      0.64       909

    accuracy                           0.84      4273
   macro avg       0.76      0.77      0.77      4273
weighted avg       0.84      0.84      0.84      4273
```

```
Initial Test Set F1 Macro Score: 0.7667480317290685
Confusion Matrix:
[[3008  356]
 [ 321  588]]
Test F1 Macro Score: 0.7667480317290685
              precision    recall  f1-score   support

           0       0.90      0.89      0.90      3364
           1       0.62      0.65      0.63       909

    accuracy                           0.84      4273
   macro avg       0.76      0.77      0.77      4273
weighted avg       0.84      0.84      0.84      4273
```

```
Best score: 0.774312614386875
Stacking Classifier F1 Macro Score: 0.7663145793595496
Test F1 Macro Score: [0 1 0 ... 0 0 0]
              precision    recall  f1-score   support

           0       0.90      0.90      0.90      3364
           1       0.63      0.63      0.63       909

    accuracy                           0.84      4273
   macro avg       0.77      0.77      0.77      4273
weighted avg       0.84      0.84      0.84      4273
```

0.7727

Predictions

Predicted Good

education	race	sex	income_po	marital_st	rent_or_ow	employme	hhs_geo_r	census_m	household	household	employme	employme	doctor_opi	behavioral	actual	predicted
College Gr	White	Female	<= $75,00(	Not Marrie	Own	Employed	qufhixun	Non-MSA	0	0	saaquncn	xgwztkwe	4	1	0	0
Some Coll	White	Male		Married	Own	Not in Labo	oxchjgsf	Non-MSA	1	0			5	1	0	0
< 12 Years	White	Female	<= $75,00(	Married	Own	Not in Labo	dqpwygqj	MSA, Not F	1	0			6	1	1	1
< 12 Years	Black	Female	Below Pov	Not Marrie	Own	Unemploy	lzgpxyit	MSA, Princ	1	0			3	1	0	0
College Gr	White	Male	<= $75,00(	Married	Own	Not in Labo	bhuqouqj	MSA, Not F	1	0			2	1	0	0

Predicted Missed

race	sex	income_po	marital_st	rent_or_ow	employme	hhs_geo_r	census_m	household	household	employme	employme	doctor_opinion_interaction	behavioral_risk_score	actual	predicted
White	Male	> $75,000	Married	Own	Employed	kbazzjca	MSA, Not F	1	0	wlfvacwt	xtkaffoo	6	0	0	1
White	Female	<= $75,00(	Married	Own	Not in Labo	lzgpxyit	MSA, Not F	1	0			3	0	1	0
White	Female	<= $75,00(	Not Marrie	Own	Not in Labo	kbazzjca	MSA, Princ	1	0			5	1	0	1
White	Female	> $75,000	Married	Own	Employed	mlyzmhmf	MSA, Not F	1	2	fcxhlnwr	cmhcxjea	4	0	1	0
White	Female	<= $75,00(	Not Marrie	Own	Employed	lzgpxyit	MSA, Princ	0	1	mfikgejo	hfxkjkmi	4	1	1	0

The Feature interaction created for doctor and opinion of effectiveness may be the reason for a bad prediction. Need to investigate what offsets this prediction and build a new interaction