Determine Best Model for Diabetes Prediction using Feature and Model Selection

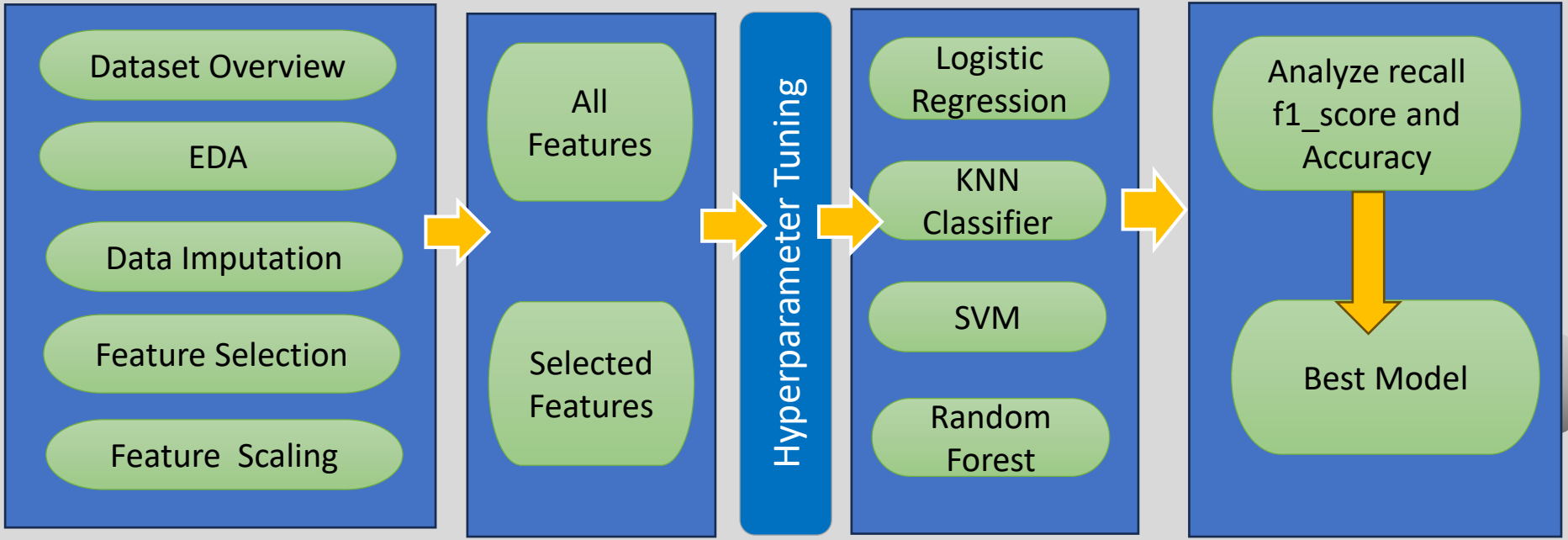
Chinmayee Raman

MET CS 677

Spring 2024

BOSTON UNIVERSITY

Model Selection Process



BOSTON UNIVERSITY

Dataset Overview

Class Label: Outcome
Outcome = 0 **(negative for diabetes)**
Outcome = 1 (**positive for diabetes)**

8 numerical features
Size : 768 rows and 9 columns
No null values

Feature Variables	Description
Pregnancies	Number of times pregnant
Glucose	Plasma glucose concentration at 2 Hours in an oral glucose tolerance test (GTIT)
Blood Pressure	Diastolic Blood Pressure (mm Hg)
Skin Thickness	Triceps skin fold thickness (mm)
Insulin	2-Hour Serum insulin (µh/ml)
BMI	Body mass index [weight in kg/(Height in m)]
Diabetes Pedigree Function	Diabetes pedigree function
Age	Age (years)

BOSTON UNIVERSITY

EDA – Statistical Summary and Inference

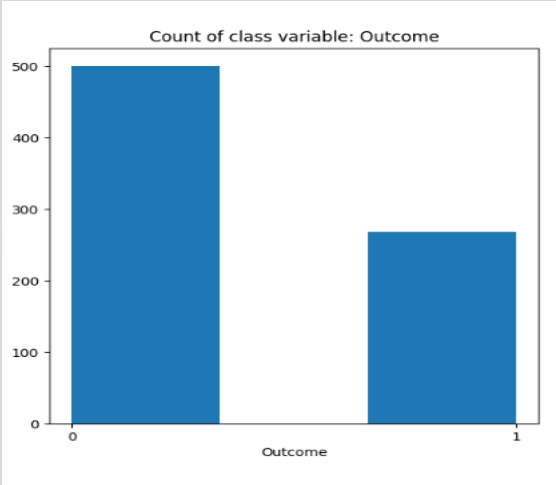
- Has 0 values for all features except age and diabetes pedigree function
- Okay to have 0 value for pregnancies
- Insulin mean and median far apart indicating likely skewed distribution
- Need scaling as data ranges are varied

Feature Variables	Pregnancies	Glucose	Blood Pressure	Skin Thickness	Insulin	BMI	Diabetes Pedigree Function	Age
Standard Deviation	3.370	31.973	19.356	15.952	115.244	7.884	0.331	11.760
Min	0	0	0	0	0	0	0.078	21
25%	1.00	99.00	62.00	0.00	0.00	27.30	0.24	24.00
Median	3.00	117.00	72.00	23.00	30.50	32.00	0.37	29.00
Mean	3.845	120.895	69.105	20.536	79.799	31.993	0.472	33.241
75%	6.00	140.25	80.00	32.00	127.25	36.60	0.63	41.00
Max	17	199	122	99	846	67.1	2.42	81

BOSTON UNIVERSITY

EDA: Class Label Imbalance

- Count of class Labels: Outcome
  - 0 500
  - 1 268
- Negative labels almost double that of positive
- May lead to decrease accuracy in positive(decease prediction)
- Balancing class weight via model parameters wherever feasible

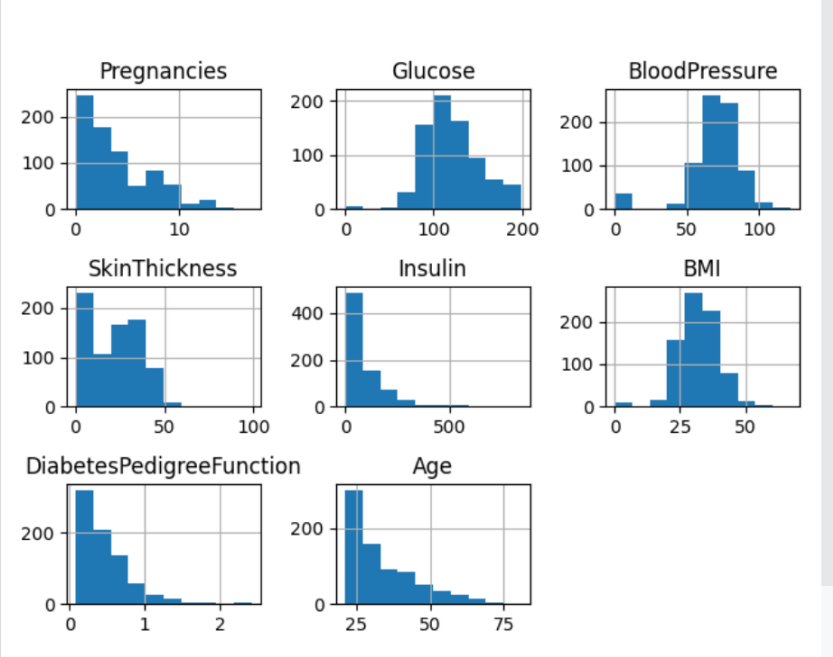


BOSTON UNIVERSITY

EDA: Univariate Analysis – Imputing Zero Values

Count of zeros values by column:

Pregnancies : 111
Glucose : 5
Blood Pressure : 35
Skin Thickness : 227
Insulin : 374
BMI : 11
Diabetes Pedigree Function : 0
Age: 0

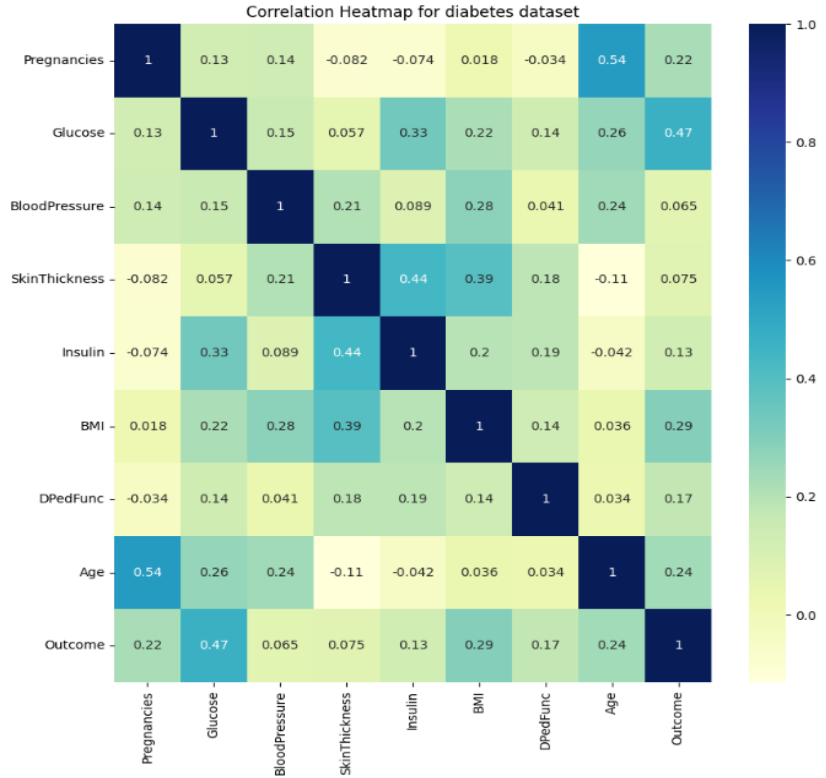


Methodology for Imputation zero values

- split dataset was split into train and test - 75/25 ratio
- scaled the data using Standard Scaler
- imputed Train and Test separately
- imputed the feature zero values to median for features with skewed distribution:
  - Insulin, and Skin Thickness
- imputed the feature zero values to mean for feature with normal distribution:
  - 'Glucose', 'Blood Pressure', 'BMI'
- pregnancies not imputed as a person can have 0 pregnancies
- repeated above steps: for all features and for selected features

BOSTON UNIVERSITY

EDA: Feature Correlation



Highest Correlation of Outcome:

- Glucose: 0.47
- BMI: 0.29
- Age: 0.24
- Pregnancies: 0.22
- Diabetes Pedigree Function: 0.17
- Insulin: 0.13
- Blood Pressure: 0.065
- Skin Thickness: 0.075

Highest Correlation among features:

- Pregnancies and age: 0.54
- Skin Thickness and Insulin: 0.44
- Skin Thickness and BMI: 0.39

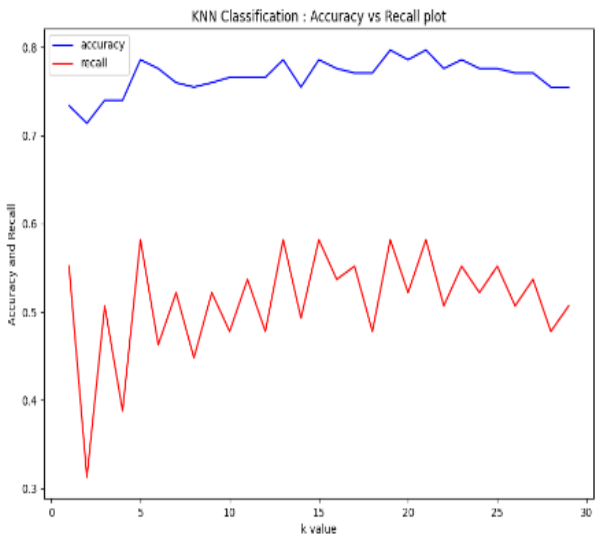
BOSTON UNIVERSITY

Feature Selection

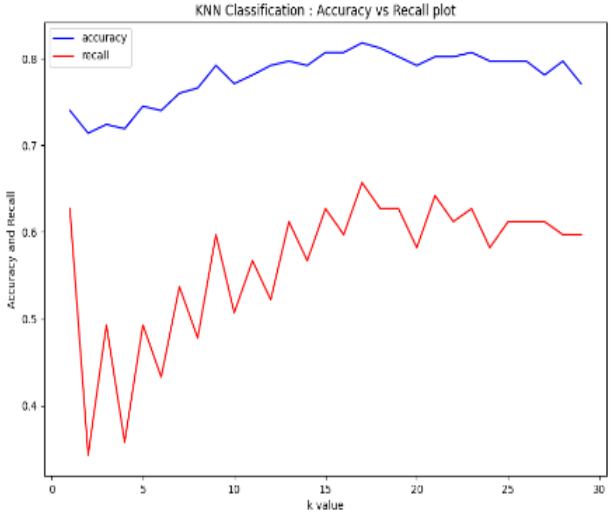
- Following Features with highest correlation to class label('Outcome') were chosen:
  - Glucose
    - Above normal Glucose in blood indicates diabetes
  - BMI
    - Higher BMI indicates higher weight of the person
  - Age
    - Age of the person
  - Diabetes Prediction Function
    - Numerical value of the effect of genetics on getting diabetes
- Even though Pregnancies had a high correlation to 'Outcome' it was not chosen as:
  - It has a high correlation with another input feature Age
  - It has values of very high pregnancies like 15,16,17 which may be unwanted outliers

kNN Classifier: Tuning Hyperparameter k iteratively (n neighbors)

All Features : best k = 19 (for max recall and accuracy)



Selected Features : best k = 17 (for max recall and accuracy)

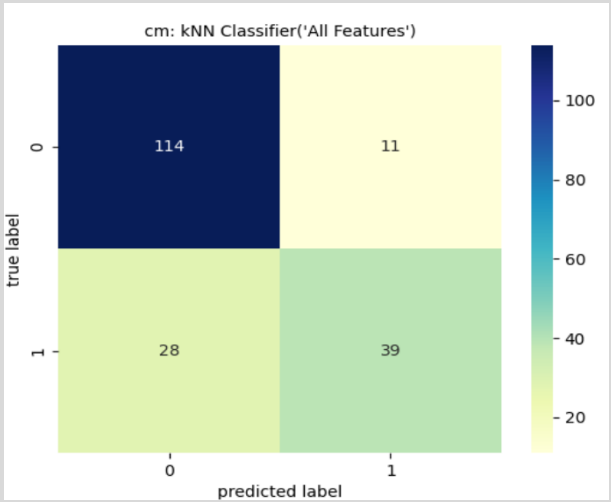


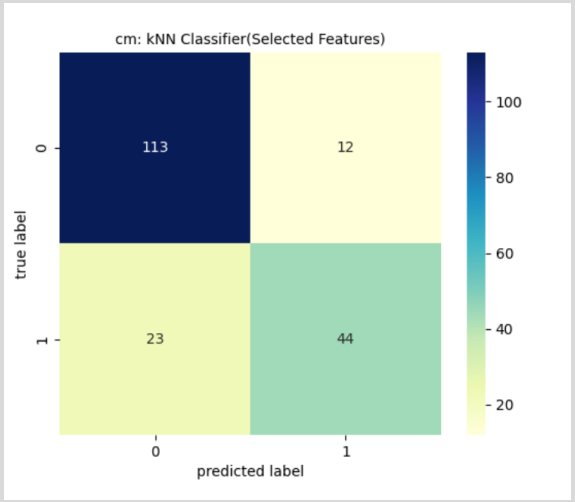
BOSTON UNIVERSITY

kNN Classifier: Performance Metrics

Features	Accuracy	Precision	Recall	f1_score	TNR	TN	FP	FN	TP
ALL	0.7969	0.78	0.5821	0.6667	0.912	114	11	28	39
Selected	0.8177	0.7857	0.6567	0.7154	0.904	113	12	23	44

Accuracy,Recall,f1_score improved after feature selection



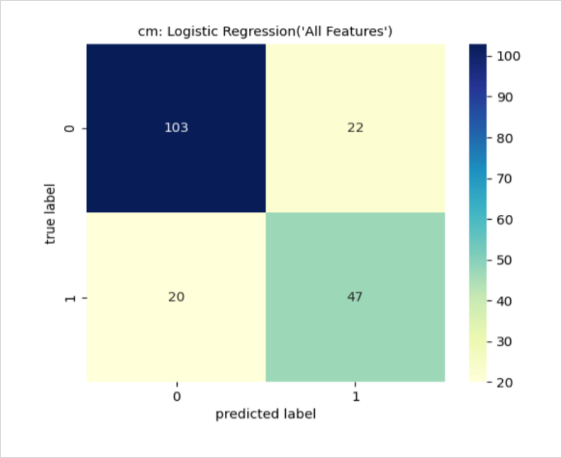


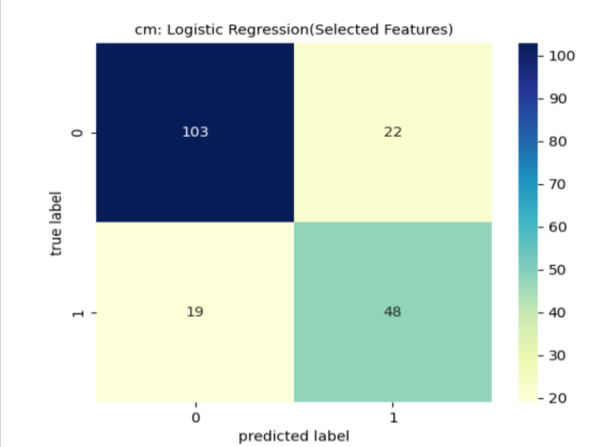
BOSTON UNIVERSITY

Logistic Regression: Tuning hyperparameter C iteratively (Inverse of regularization strength)

Features	Tuning C for Best Accuracy	Accuracy	Precision	Recall	f1_score	TNR	TN	FP	FN	TP
ALL	best C: 1	0.7812	0.6812	0.7015	0.6912	0.82	103	22	20	47
Selected	best C: 0.001	0.7865	0.6857	0.7164	0.7007	0.82	103	22	19	48

- Accuracy, Recall,f1_score improved by around 1% after feature selection
- To balance the class imbalance used parameter class_weights = ‘balanced’





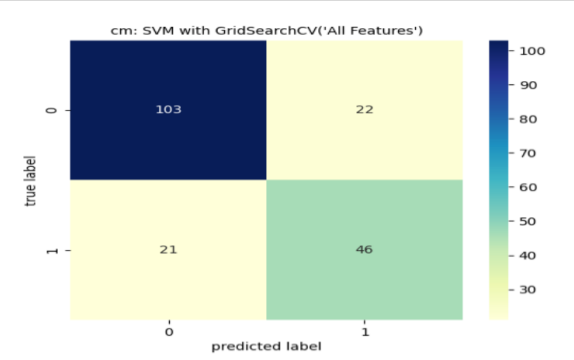
BOSTON UNIVERSITY

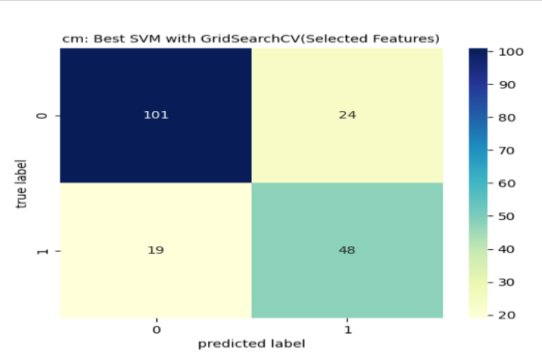
SVC: Hyperparameter Tuning for best C and gamma using GridSearchCV

- param_grid = {'C': [0.1,1, 10, 100, 1000], 'gamma': [1, 0.1, 0.01, 0.001, 0.0001] }
- All features: best params : C =100,gamma= 0.001,class_weight = 'balanced',kernel = 'rbf'
- Selected features: best params: C=10, gamma=0.01,class_weight='balanced',kernel = 'rbf'

Features	Accuracy	Precision	Recall	f1_score	TNR	TN	FP	FN	TP
ALL	0.776	0.6765	0.6866	0.6815	0.824	103	22	21	46
Selected	0.776	0.6667	0.7164	0.6906	0.808	101	24	19	48

- Recall improved by 3 % and f1_score improved by around 1% slightly after feature selection. No change in accuracy with feature selection.

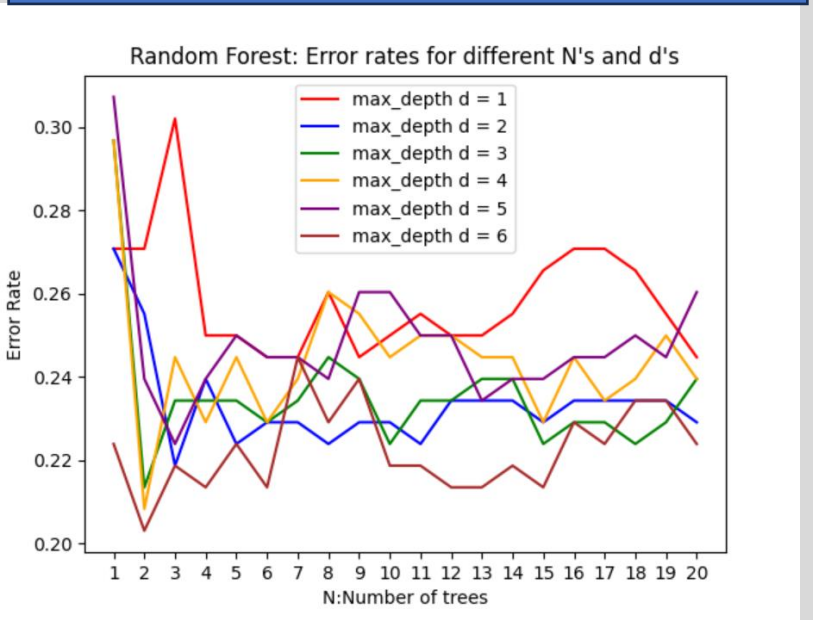




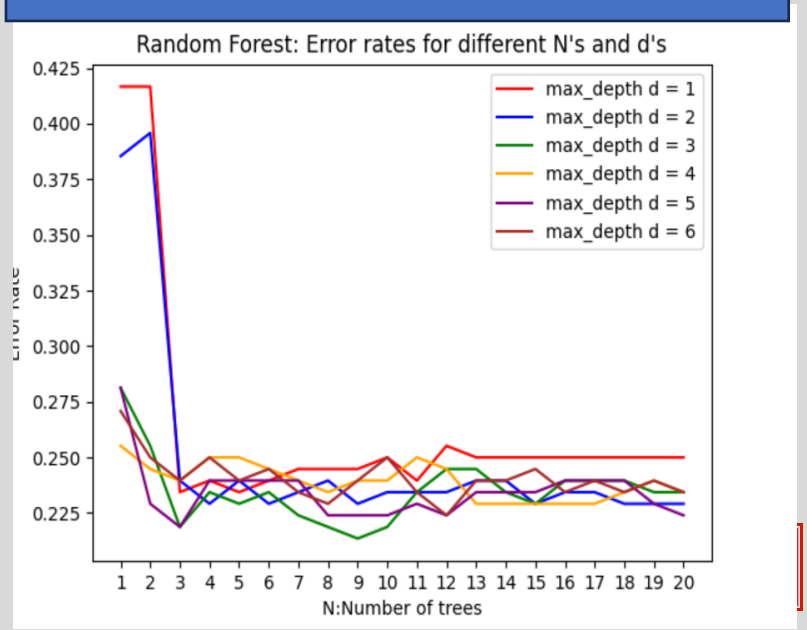
BOSTON UNIVERSITY

Random Forest: Hyperparameter tuning iteratively for N (number of trees) and d(max depth)

All Features : Best N: 2 Best d: 6 (for maximum accuracy)



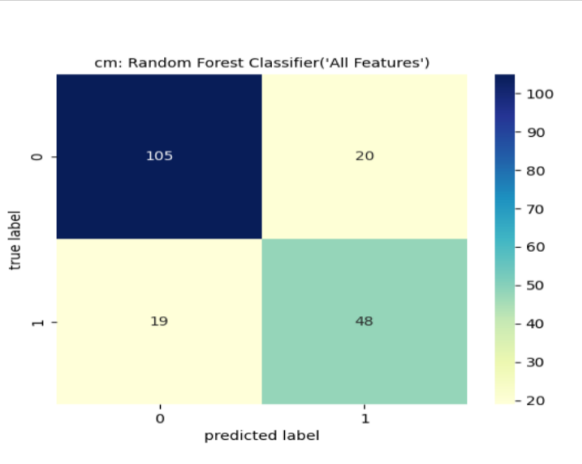
Selected Features : Best N: 9 Best d: 3 (for maximum accuracy)

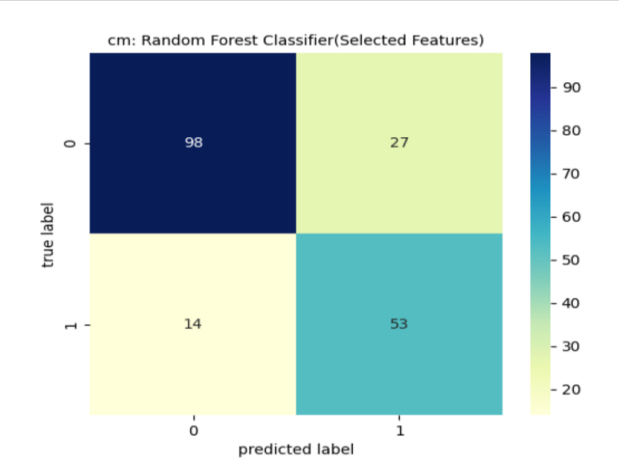


Random Forest Performance Metrics

Features	Accuracy	Precision	Recall	f1_score	TNR	TN	FP	FN	TP
ALL	0.7969	0.7059	0.7164	0.7111	0.84	105	20	19	48
Selected	0.7865	0.6625	0.791	0.7211	0.784	98	27	14	53

- Recall improved significantly to 79.1 percent after feature selection
- Accuracy decreased by 1% and f1_score increased by 1%





BOSTON UNIVERSITY

Result Analysis

Model	Features	Accuracy	Precision	Recall	f1_score	TNR	TN	FP	FN	TP
kNN	ALL	0.7969	0.7800	0.5821	0.6667	0.9120	114	11	28	39
kNN	Selected	0.8177	0.7857	0.6567	0.7154	0.9040	113	12	23	44
Logisitc Regression	ALL	0.7812	0.6812	0.7015	0.6912	0.8240	103	22	20	47
Logisitc Regression	Selected	0.7865	0.6857	0.7164	0.7007	0.8240	103	22	19	48
SVC(GridSearch)	ALL	0.7760	0.6765	0.6866	0.6815	0.8240	103	22	21	46
SVC(GridSearch)	Selected	0.7760	0.6667	0.7164	0.6906	0.8080	101	24	19	48
Random Forest	ALL	0.7969	0.7059	0.7164	0.7111	0.8400	105	20	19	48
Random Forest	Selected	0.7865	0.6625	0.7910	0.7211	0.7840	98	27	14	53

- Ran all models individually for ‘All Feature’ and ‘Selected Features’ of Glucose, BMI, Age and Diabetes Pedigree Function
- kNN with selected features has the highest accuracy at 81.77 % and highest precision 78.57% with lower recall 65.7 % and f1_score of 71.54%. This could be because I could not use a parameter(knn has no class_weight) to offset the class imbalance.
- Random forest with selected features has the best recall at 79.1 percent and best f1_score of 72.11%, but lower accuracy 78.65% and lower precision 66.25%.
- Overall Random forest showed the maximum improvement with ‘Selected Features’ while Logistic Regression showed marginal improvement.

Conclusion and Next Steps

- Best Model: Random Forest Classifier
  - The Random Forest Model has the best recall at 79.1 % and best f1_score at 72.11%
  - Hence it is the best model to predict accurately whether a person has diabetes or is likely to get diabetic
- Using an ensemble model, the random forest helped to get best recall and f1 score over the individual models

- Next Steps: Though a recall of 79.1% and f1_score of 72.11% is not bad I feel we could improve on the performance metrics by:
  - Trying out other models(specially ensemble)
  - Trying to impute data in with other imputation methods like kNN impute
  - Trying to reduce class imbalance by resampling

BOSTON UNIVERSITY