Sit720 11.1 HD

Question 1

Results of the classifications methods in the article:

Table 4. Performances (%) of ML and NN algorithm on Pima dataset using train-test and CV.

Protocol	Algorithm	Accuracy	Precision	Recall	F1-Score
Train-Test	DT	65.08	.65	.65	.65
	RF	79.33	.80	.79	.79
	SVM	69.03	.69	.69	.69
	Staking Ensemble	75.03	.75	.75	.75
Cross-validation	DT	68.31	.65	.68	.67
	RF	76.81	.77	.79	.78
	SVM	68.61	.68	.70	.69
	Staking Ensemble	77.10	.68	.70	.69

The study used three datasets, however this report is only interested in one of those:

PIMA Indian Diabetes Dataset: 768 patients (268 positive, 500 negative for diabetes), with 8 independent features.

The methods used included:

1. Machine learning algorithms: Decision Tree, Random Forest, SVM.
2. Ensemble method: Stacking with logistic regression as meta-learner.

The experiment protocol:

Data preprocessing stages:

1. Outlier Detection and handling: The z-score was used to find how much data deviated from the mean. Interquartile range was employed to detect outliers and the outliers were replaced with median of their respective feature.
2. For the PIMA dataset, zero values in certain attributes (e.g. BMI among others) were considered biologically implausible and were treated as missing values. The median was used to fill in missing values.
3. Scaling: All data was standardized to ensure all features were on a similar scaled.
4. SMOTE was used to balance the class skewedness.

The data was then split into 70/30 training and test data. Cross-validation was employed to ensure the model's performance was robust and generalizable. This study used a 5-fold CV.

The entire dataset was divided into 5 equally sized subsets (folds), in each iteration (5 total) 4 folds were used as the training set and 1 fold was the validation set.

Each ML and stacking ensemble method was trained on the 4 training folds. The trained model was then evaluated on the validation fold. Performance metrics such as accuracy, precision, recall and F1-score were calculated for each iteration. After all iterations the metrics were averaged and that's what's reported in the above table.

Table 4 shows the results of the ML techniques. The accuracy from the stacked ML models was 75.03% and 77.10% from the CV stacked method. RF classifier gave the highest accuracy of 79.33%, 80% precision, 79% recall and 79% f1-score, outperforming the other method by 2-14%.

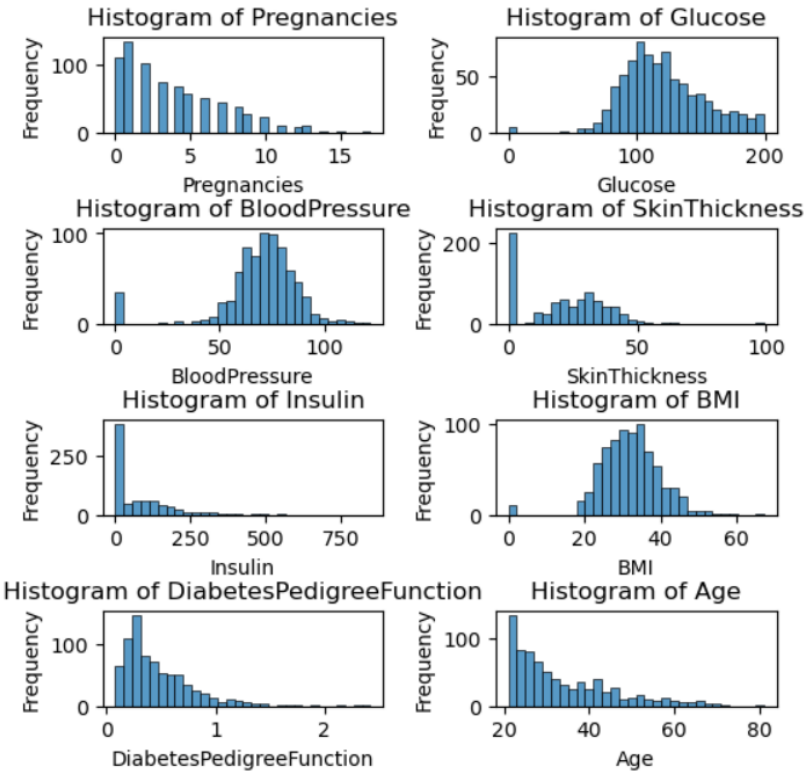
In reproducing this experiment I followed all the steps they did in the study. However, many steps were just stated and didn't explicitly state the method used for each step. These include just mentioning they standardized the data, used SMOTE, built a decision tree, SVM and Random forest model without stating what parameters they used. Also that they just used 5-fold cross validation, and not what type of CV method they implemented. These short insights made it impossible to reproduce the results in the study. In building the models the results were very similar, however, the cross-validation results differed quite a bit.

Question 2

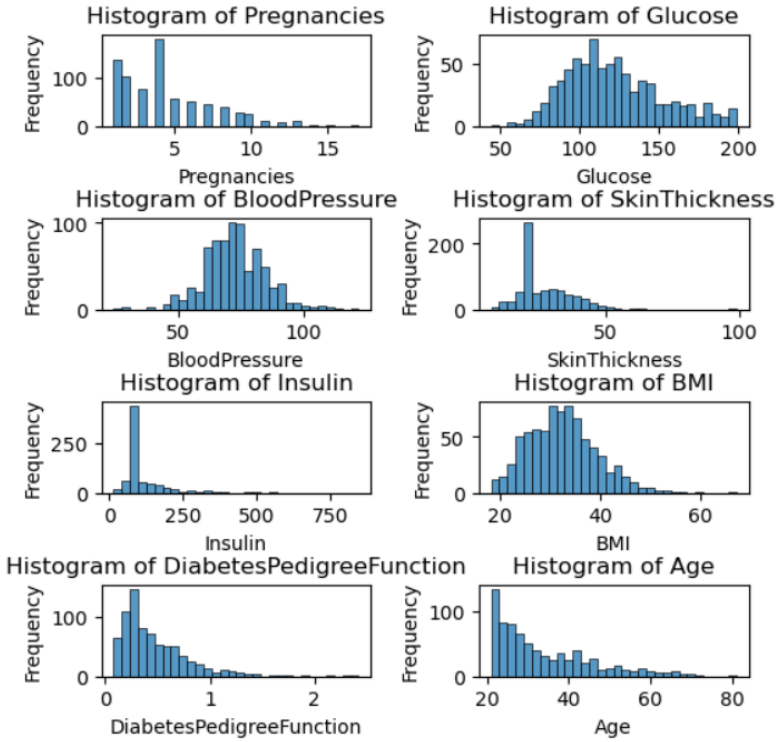
The motivation behind my process was to implement a different process then the one used in the study and to ensure the method was more articulate. I choose Random Forest as it was the best performing model in the study, but I wanted to fine tune the method tweaking it my own way and being specific in my process.

The preprocessing stage began by handling the 0 values by replacing them with mean value instead of median. Using domain knowledge, having 0 values was impossible so they needed to be dealt with. I choose mean to explore how it would shape the date differently.

Data distribution before changing:

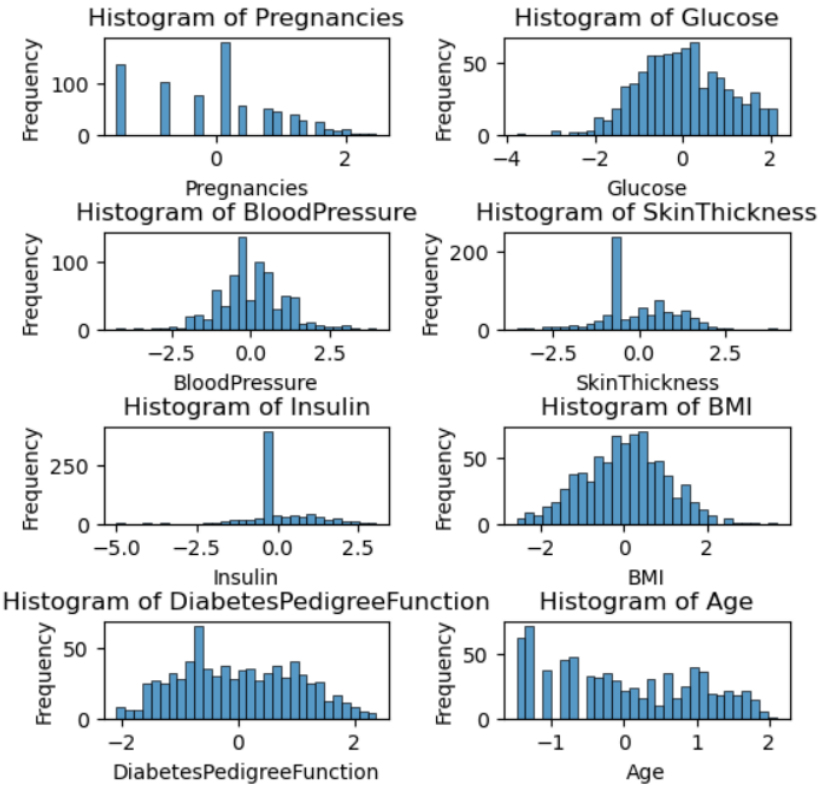


The data after changing 0 values to mean:

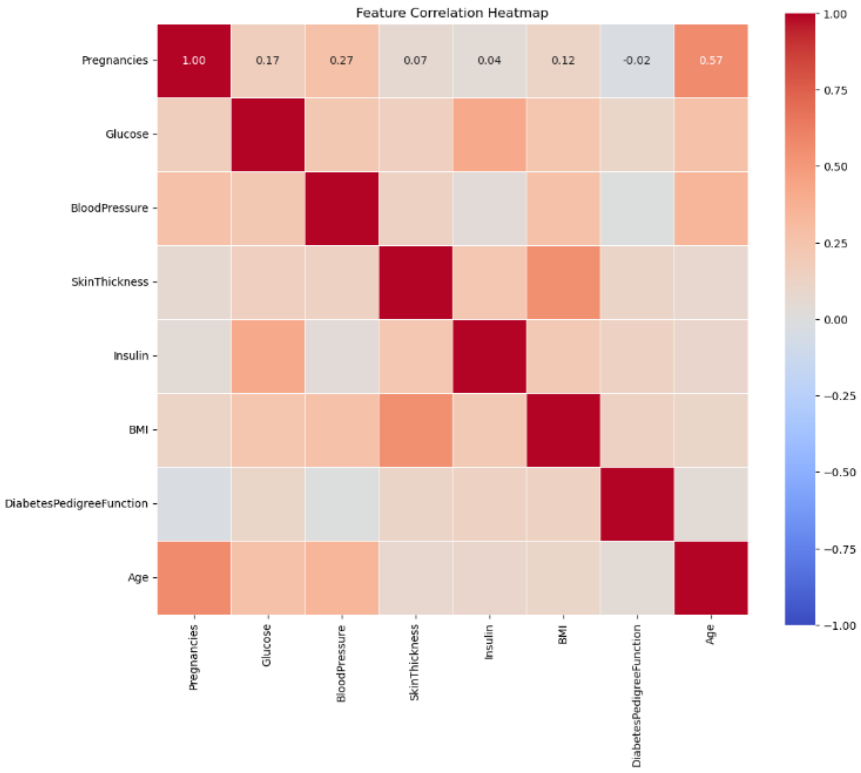


I didn't bother with changing outliers as Random Forest is robust to outliers. I also transformed the data to deal with the skewness, which also smooths out outliers.

This is how the data looks after applying Power Transform:



I also got a heatmap of correlations between features to see if any are highly correlated and need to be removed.



None of the features were highly correlated so none were removed.

Main differences in preprocessing method:

1. Mean replacement instead of median.

2. Not altering outliers.
3. Using a Power Transform scaler
4. Exploring feature correlation

Since the dataset was small I also decided to use a 80/20 training/test split to try and help train the model better instead of the 70/30 split in the study. I also didn't use SMOTE to balance the classes as I felt it better to try and train model on current proportions of classes.

I then used a Grid Search to search through 144 different Random Forest models with hyperparameter tuning. This was not done in the study. I then built a Random Forest model with the proposed best parameters.

The Grid Search explored:

n_estimators: [20, 50, 100,200],

criterion:('gini','entropy','log_loss'),

max_depth: [5,10,50,100],

max_features: ('sqrt','log2',None)

The best parameters from Grid Search were:

criterion: 'entropy',

max_depth: 5,

max_features: None,

n_estimators': 100

The results from this Random Forest model were:

	precision	recall	f1-score	support
0	0.81	0.80	0.81	99
1	0.65	0.67	0.66	55
accuracy			0.75	154
macro avg	0.73	0.74	0.73	154
weighted avg	0.76	0.75	0.75	154

I performed my own hyperparameter tuning and found the best parameters were:

criterion= 'gini' ,

n_estimators = 100,

max_depth = 50,

max_features = None

The results were:

	precision	recall	f1-score	support
0	0.84	0.81	0.82	99
1	0.68	0.73	0.70	55
accuracy			0.78	154
macro avg	0.76	0.77	0.76	154
weighted avg	0.78	0.78	0.78	154

There was a small improvement in accuracy, but a decent improvement of nearly 10% in recall for class 1 (predicting diabetes). I changed max_depth to higher number as I thought 5 in the Grid Search was too low. I changed criterion to gini to test if it can improve model.

I didn't perform cross-validation as in the study as I believed the dataset to be too small to effectively perform cross-validation.

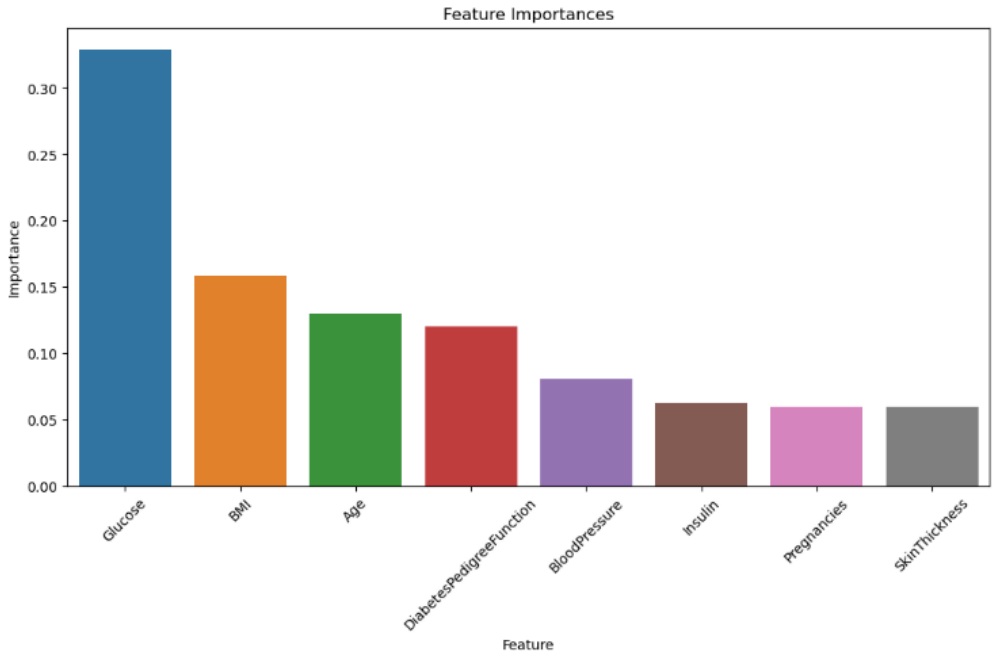
The results from Random Forest model in part 1 were:

	precision	recall	f1-score	support
0	0.85	0.78	0.81	151
1	0.64	0.74	0.69	80
accuracy			0.77	231
macro avg	0.75	0.76	0.75	231
weighted avg	0.78	0.77	0.77	231

The model I built was a slight improvement. Overall accuracy, precision, recall and f1-score was 0.01 higher. Note I am comparing to this model and not the model study as it wasn't stated how they performed the Random Forest in the study. In part 1 this was a baseline Random Forest with no hyperparameter tuning.

The differences in my model include setting max_depth to 50 vs None in, and max_features to None vs 'sqrt in base model. We can see even though the dataset is small, hyperparameter tuning can improve results.

To improve on the study I explicitly stated how I performed each step, so any reader can perform the exact same steps and get the same results. I also found the feature importances to see which factors contribute to predicting diabetes most.



We can see that Glucose is the most important feature by more than twice the second most important feature BMI.

The main limitations for this model building experiment is the severe lack of data. Only having 768 data points is no where near enough to build a predictive model for diabetes. This preliminary analysis however is good in telling us what kinds of data we need to collect.

References

Reza, M. S., Amin, R., Yasmin, R., Kulsum, W., & Ruhi, S. (2024). Improving diabetes disease patients classification using stacking ensemble method with PIMA and local healthcare data. *Heliyon, 10*(2), e24536. https://doi.org/10.1016/j.heliyon.2024.e24536

Brownlee, J. (n.d.). *Power transforms with scikit-learn*. Machine Learning Mastery. Retrieved July 25, 2024, from https://machinelearningmastery.com/power-transforms-with-scikit-learn/

Scikit-Learn Developers. (n.d.). *sklearn.model_selection.GridSearchCV*. Scikit-Learn. Retrieved July 25, 2024, from https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

Scikit-Learn Developers. (n.d.). *sklearn.ensemble.RandomForestClassifier*. Scikit-Learn. Retrieved July 25, 2024, from https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

Scikit-Learn Developers. (n.d.). *sklearn.ensemble.StackingClassifier*. Scikit-Learn. Retrieved July 25, 2024, from https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.StackingClassifier.html

Scikit-Learn Developers. (n.d.). *sklearn.model_selection.StratifiedKFold*. Scikit-Learn. Retrieved July 25, 2024, from https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html

Machine Learning for Diabetes

Result Summary

My Model

	precision	recall	f1-score	support
0	0.84	0.81	0.82	99
1	0.68	0.73	0.70	55
accuracy			0.78	154
macro avg	0.76	0.77	0.76	154
weighted avg	0.78	0.78	0.78	154

Original Model

	precision	recall	f1-score	support
0	0.85	0.78	0.81	151
1	0.64	0.74	0.69	80
accuracy			0.77	231
macro avg	0.75	0.76	0.75	231
weighted avg	0.78	0.77	0.77	231

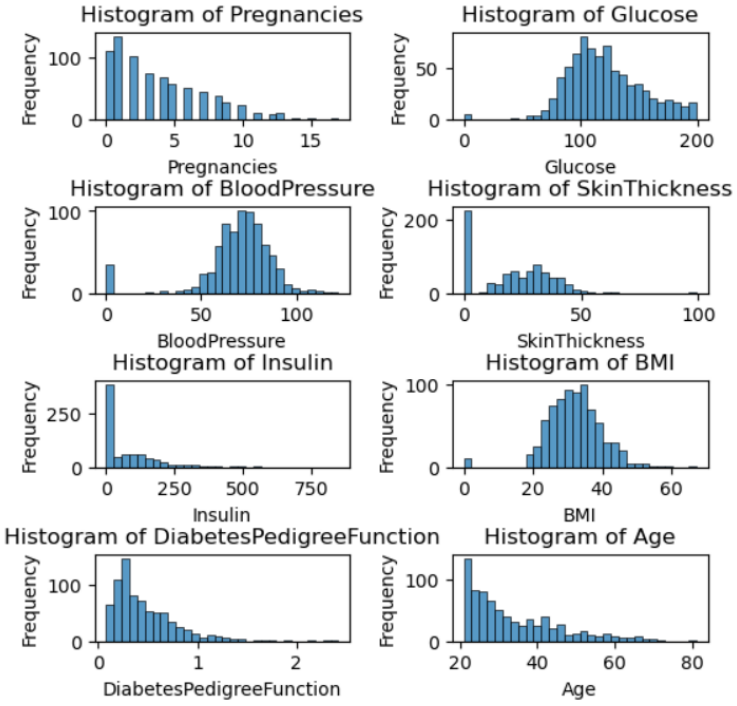
- Through specific hyperparameter tuning my Random Forest was able to improve on original model slightly.
- However, severe limitation was the small size of the data set. This was mentioned in the study. To get more precise results we would need a lot more data.

Preprocessing

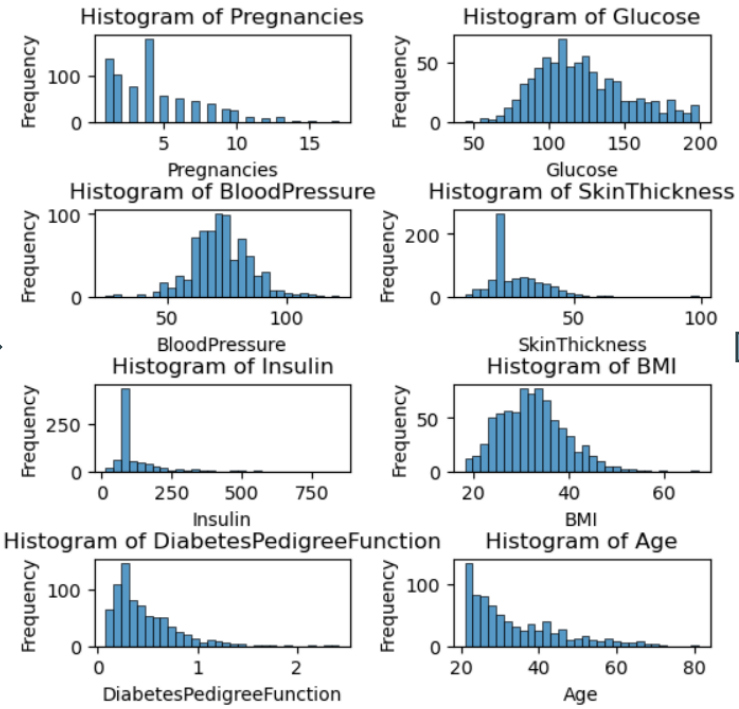
1. Replace 0 values with mean
2. Not altering outliers
3. Applying Power Transform to scale data
4. Examining Feature Correlation

Transforming Data Distribution Shape

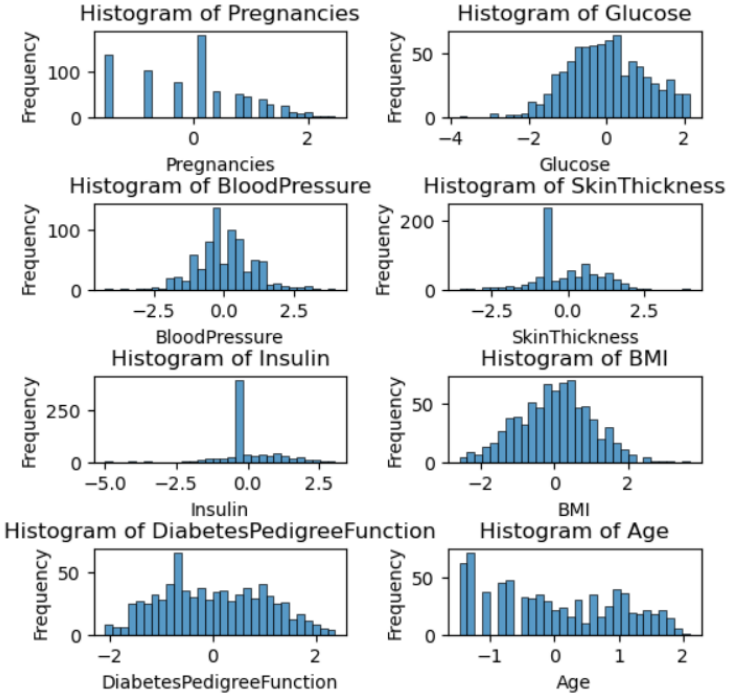
Original Data



Changing 0 to mean



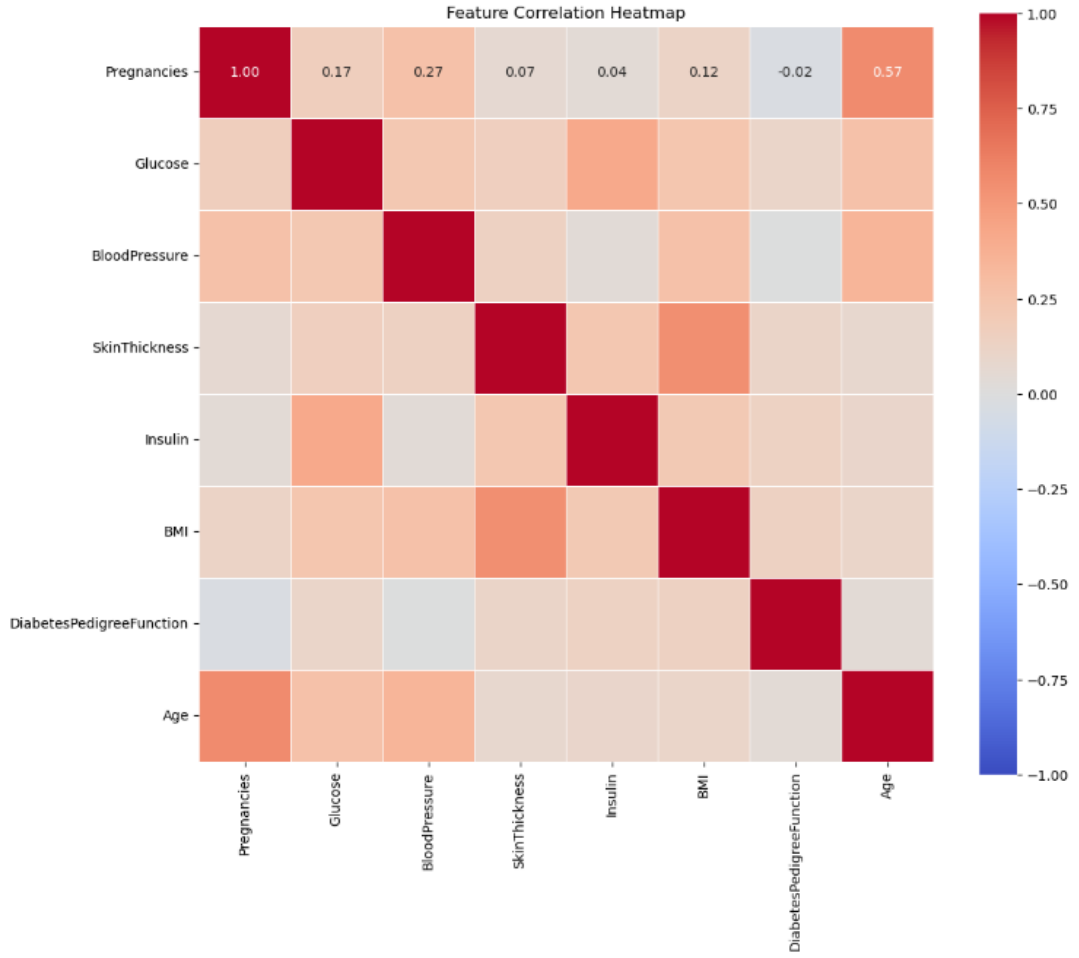
Power Transform



Transforming data we seen final data looks like normal distribution mostly and data is centralised, which is ideal for modelling.

Correlation Between Features

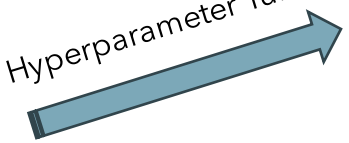
- None of the features were highly correlated so none had to be dropped.



Hyperparameter Tuning

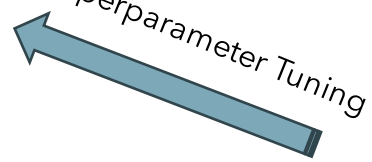
Grid Search Results

	precision	recall	f1-score	support
0	0.81	0.80	0.81	99
1	0.65	0.67	0.66	55
accuracy			0.75	154
macro avg	0.73	0.74	0.73	154
weighted avg	0.76	0.75	0.75	154



Best Model Results

	precision	recall	f1-score	support
0	0.84	0.81	0.82	99
1	0.68	0.73	0.70	55
accuracy			0.78	154
macro avg	0.76	0.77	0.76	154
weighted avg	0.78	0.78	0.78	154

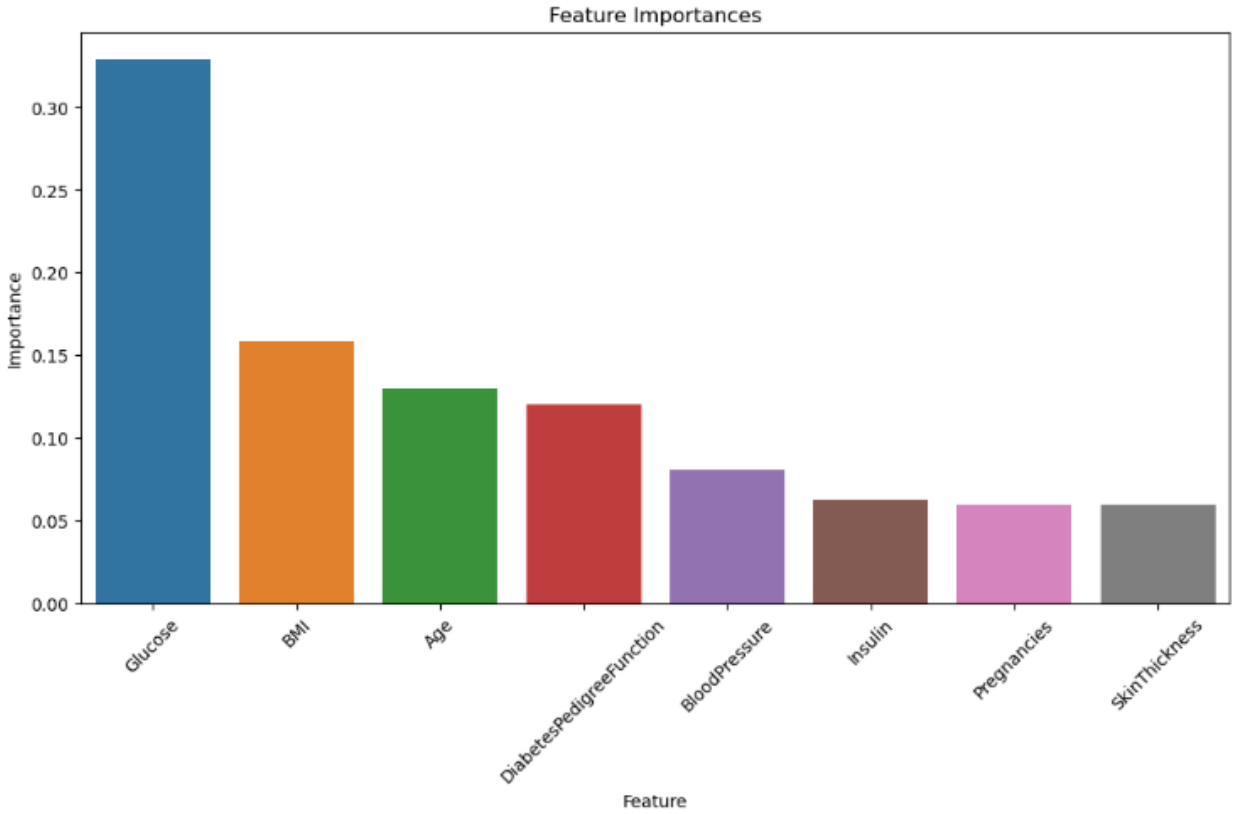


Original Results

	precision	recall	f1-score	support
0	0.85	0.78	0.81	151
1	0.64	0.74	0.69	80
accuracy			0.77	231
macro avg	0.75	0.76	0.75	231
weighted avg	0.78	0.77	0.77	231

- Combining Grid Search and extra hyperparameter tuning results in best model results.

Feature Importance



- Glucose levels is by far the most important feature in predicting diabetes, more than twice the importance of second most important feature, BMI.