Prediction of Diabetes in Pima Indian Women

Woodrow Reese

Introduction

The Pima Indian tribe resides in both Arizona and Mexico. Within Arizona, there are two reservations with a total population of 20,000 people. 34.2% of men and 40.8% of women within the Pima community have type 2 diabetes. The Arizona Pima community has the highest rate of diabetes in the world! **Scientifically**, the presence of diabetes is described to be due to a genetic mutation which causes β-cell dysfunction, which results in obesity and insulin resistance. [1] These two factors are strongly associated with type 2 diabetes. **Historically**, the presence of diabetes can be due to the government's negligence of the community during the nationwide financial/agricultural (drought) crisis through the 70s-80s. The most notable events include: The Russian Grain Transactions, The Soybean Embargo and The Farm Credit and Debt Situation, which resulted in agricultural monopolies, policy change, and the bankruptcy of smaller farms. Native American communities didn't have representation throughout this era. As a result, these communities became food desserts. 40% of the Arizona Pima Indian's calories came from fat.[2] On a wider scale, 9.3% of the United States' general population and 16% of the entire Native American population have type 2 diabetes.

Other researchers within the Kaggle community have explored this dataset in a very structured way. One that has captured my mind has yielded a high performing Gradient

Booster (LightGBN) and KNN model with 90% accuracy. We will follow the pipeline in the listed order:

ETL + ML

1: Extract

- Import/Load the dataset

2: Transform

- Clean and impute the data
- Perform EDA to determine necessary transformations or feature engineering

3: Load

- Prepare data for CV and modeling

4: ML

- Train model and evaluate performance

Dataset

The dataset[3] is from the National Institute of Diabetes and Digestive Kidney Diseases. There are 8 predictors: Number of pregnancies, glucose, blood pressure, skin thickness (of triceps fold), insulin, BMI, diabetes pedigree function, and age. The target variable is a prediction of the presence of diabetes represented by a 1 or a 0. The data is recorded on 768 women over the age of 21 where 268 are predicted to have type two diabetes. The predictors are:

- Pregnancies: The number of pregnancies a person has had [0-17]

- Glucose: An oral test measuring the average glucose concentration after a 2 hour fast. The typical measurement is 70-105 mg/dL. Low glucose signals excessive insulin while high glucose signals insufficient insulin production. Both may be caused by insulin resistance.
- Blood Pressure: Diastolic blood pressure measurement (while the heart is relaxed between beats). Typical rates are 60-80 mm*Hg. High blood pressure is a typical effect of type 2 diabetes.
- Skin Thickness: The skin thickness of the skin fold connecting the triceps and armpit. Typical values vary between age and body composition which may not be vividly captured through BMI.
- Insulin: Blood test calculated glucose load (μU/mL) after an 8 hour fast. A typical insulin level is between 18-56 μU/mL. Higher levels signal *insulin resistance* and early-stage type 2 diabetes. Lower levels signal late-stage type 2 diabetes.
- BMI: Body mass index = weight / height^2 (kg/m)
- Diabetes Pedigree Function: measurement of the relevance of genetics to diabetes in a person. It is based on the number of diabetic relatives, the age of relatives at the time of diagnosis, and the degree of relationship:
    - 1st degree: 50% genetic similarity. Parents, children and siblings
    - 2nd degree: 25% genetic similarity i.e. Grandparents, aunts/uncles, half siblings
    - 3rd degree: 12.5% genetic similarity i.e. First cousins and great grandparents

A low value ranges from 0-0.3, middle values from 0.3-1, and high values above 1. The highest values indicate a person was more likely to have multiple first-degree relatives diagnosed with diabetes at a young age.

- Age: 21-81 years

There are some missing values for predictors (excluding pregnancies and age), they are predicted using KNN imputation.

ETL + ML

Extract

We downloaded the dataset from Kaggle and imported it into the Python notebook using Pandas.

Transform

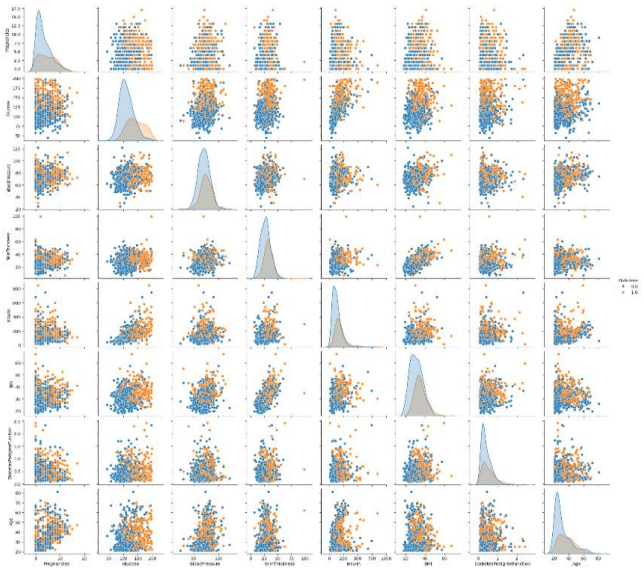
Some predictors in the dataset, such as **Glucose**, **Blood Pressure**, **Skin Thickness**, **Insulin**, and **BMI**, have missing values. To address this, the Kaggler[4] decided to impute these missing values by calculating the median for each predictor given the an Outcome. For example:

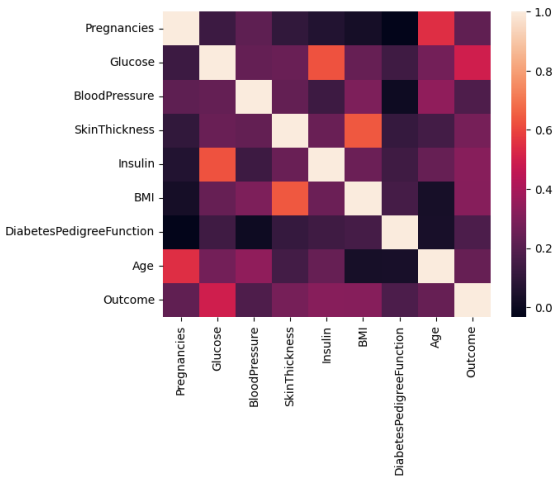
```
data.loc[(data['Outcome'] == 0 ) & (data['Insulin'].isnull()), 'Insulin'] = 102.5
data.loc[(data['Outcome'] == 1 ) & (data['Insulin'].isnull()), 'Insulin'] = 169.5
```

In general, this is a bad practice because it introduces data leakage. Data leakage occurs when information from the target variable (Outcome) is used to inform the predictors during data preprocessing. This leads to biased results and an unfair advantage while predicting. Instead, the imputation process should be performed without referencing the

target variable to maintain the integrity of the model. I have decided to perform KNN imputation to fill the missing values because we have outliers, different scales and ranges for each predictor. But first, we scale the predictors with the following: score = (x - μ) / σ. This scaling step ensures that all predictors have a mean of 0 and a standard deviation of 1 which is essential for distance-based methods like KNN.

EDA: Let's first visualize the linear relationships between the predictors.





The following linear relationships are observed: Insulin vs Glucose, SkinThickness vs BMI, Age vs Pregnancies. Glucose is a strong predictor where people usually have diabetes if Glucose > 125. The same can be observed in the correlation graph.

Feature Engineering

We attempt to create and encode binary features for specific predictor thresholds/ratios that signal a higher risk of diabetes. The Kaggler made up some features in a similar fashion, but they keep all 16 features made! With the same exact features, the model becomes less accurate in all metrics of performance.

```
DiabetesImputed['N0'] = DiabetesImputed['BMI'] * DiabetesImputed['SkinThickness']
DiabetesImputed['N8'] =  DiabetesImputed['Pregnancies'] / DiabetesImputed['Age']
DiabetesImputed['N13'] = DiabetesImputed['Glucose'] / DiabetesImputed['DiabetesPedigreeFunction']
DiabetesImputed['N12'] = DiabetesImputed['Age'] * DiabetesImputed['DiabetesPedigreeFunction']
DiabetesImputed['N14'] = DiabetesImputed['Age'] / DiabetesImputed['Insulin']

DiabetesImputed.loc[:,'N15']=0
DiabetesImputed.loc[(DiabetesImputed['N0']<1034) ,'N15']=1
```

```
DiabetesImputed.loc[:,'N2']=0
DiabetesImputed.loc[(DiabetesImputed['BMI']<=30),'N2']=1
DiabetesImputed.loc[:,'N1']=0
DiabetesImputed.loc[(DiabetesImputed['Age']<=35) & (DiabetesImputed['Glucose']<=120),'N1']=1
DiabetesImputed.loc[:,'N3']=0
DiabetesImputed.loc[(DiabetesImputed['Age']<=35) & (DiabetesImputed['Pregnancies']<=6),'N3']=1
DiabetesImputed.loc[:,'N4']=0
DiabetesImputed.loc[(DiabetesImputed['Glucose']<=125) & (DiabetesImputed['BloodPressure']<=80),'N4']=1
DiabetesImputed.loc[:,'N5']=0
DiabetesImputed.loc[(DiabetesImputed['SkinThickness']<=20) ,'N5']=1
DiabetesImputed.loc[:,'N6']=0
DiabetesImputed.loc[(DiabetesImputed['BMI']<30) & (DiabetesImputed['SkinThickness']<=20),'N6']=1
DiabetesImputed.loc[:,'N7']=0
DiabetesImputed.loc[(DiabetesImputed['Glucose']<=125) & (DiabetesImputed['BMI']<=30),'N7']=1
DiabetesImputed.loc[:,'N9']=0
DiabetesImputed.loc[(DiabetesImputed['Insulin']<200),'N9']=1
DiabetesImputed.loc[:,'N10']=0
DiabetesImputed.loc[(DiabetesImputed['BloodPressure']<80),'N10']=1
DiabetesImputed.loc[:,'N11']=0
DiabetesImputed.loc[(DiabetesImputed['Pregnancies']<4) & (DiabetesImputed['Pregnancies']!=0) ,'N11']=1
```

Load:

Following the Pipeline of the Kaggler, we will first prepare our data for a model by encoding the engineered features. The loading process will help answer **Research Question 1**;

How does the performance of different predictive models compare?

The models being used are Logistic Regression, KNN and Random Forest Classification. We will answer this by cross validating the models with default parameters. Then we will tune hyperparameters and cross validate the models with the goal of an increase in recall and a similar F1 score. Cross validation will quantify the model's performance with the following calculations:

- TP: True Positive

- TN: True Negative
- FP: False Positive
- FN: False Negative

Accuracy – The overall rate of correct predictions

= (TP + TN) / (TP + TN + FP + FN)

Precision – When I predict something positive, how often am I correct?

= TP / (TP + FP)

Recall – Of all the actual positives, how many did I correctly identify?

= TP / (TP + FN)

F1 Score – Harmonic mean of precision and recall.

= 2 * (Precision * Recall) / (Precision + Recall)

CV on models with default parameters:

Model	Accuracy %	Variance (+/-) %	Precision %	Recall %	F1
Logistic Regression	76	4.45	72	52	.60
KNN	77	2.71	69	63	.66
Random Forest	79	3.37	76	57	.65

CV with hyperparameters:

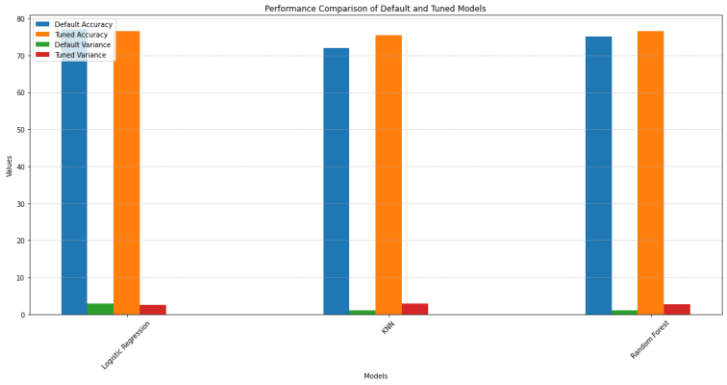
Model	Accuracy %	Variance (+/-) %	Precision %	Recall %	F1
Logistic Regression	76	2.53	72	52	.60
KNN	73	2.15	65	52	.58
Random Forest	80	2.20	74	71	.72

Model: Logistic Regression {'C': 0.2, 'solver': 'lbfgs'}

Model: KNN {'n_neighbors': 46, 'weights': 'distance'}

Model: Random Forest {'max_depth': 3, 'min_samples_split': 10, 'n_estimators': 14}

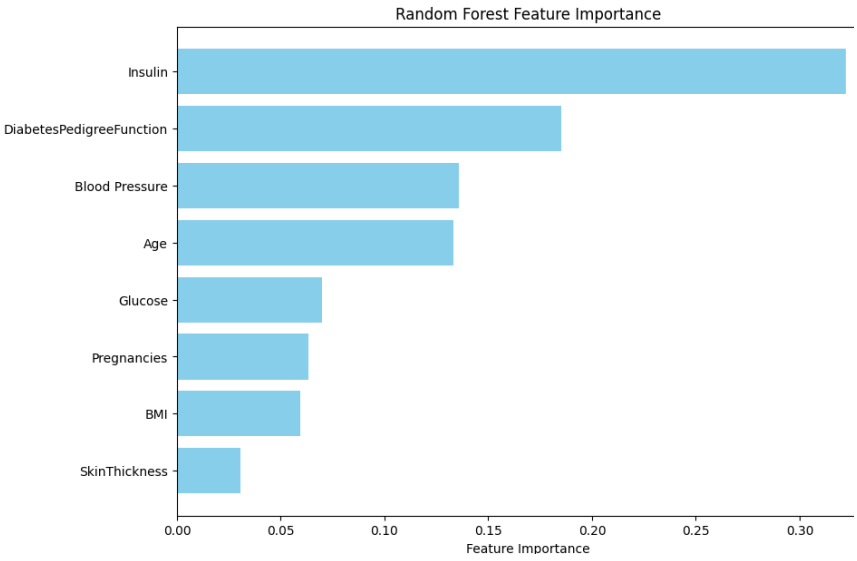
The best performing model is Random Forest Classification across the board! The biggest concern is false negatives in the case of predicting diabetes, so we should attempt to increase the recall. A recall of 0.71 means that the model fails to identify 29% of people who have diabetes. This may be addressable by modifying/removing some of our engineered features, we could also perform a *gradient booster*. Below is an illustration of the change in accuracy and variance using default vs tuned parameters.



Research Question 2

Scientifically, the presence of diabetes is described to be due to a genetic mutation which causes β-cell dysfunction, which results in obesity and insulin resistance. [1] *How important are these predictors?*

Let's analyze BMI and Insulin's impact on the model by performing feature analysis.



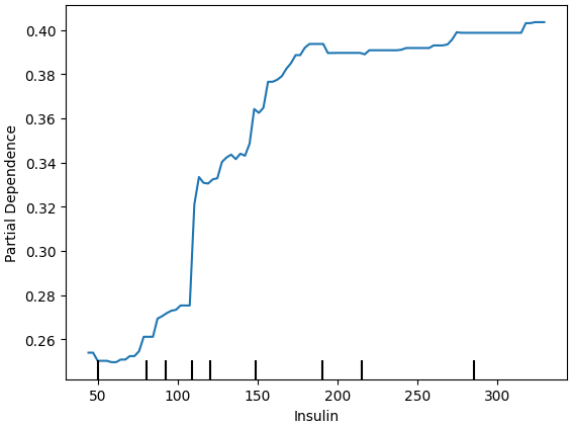
Insulin: 32%

- The most critical predictor which aligns with the scientific understanding that β-cell dysfunction results in insulin *resistance*, the main reasoning for diabetes.

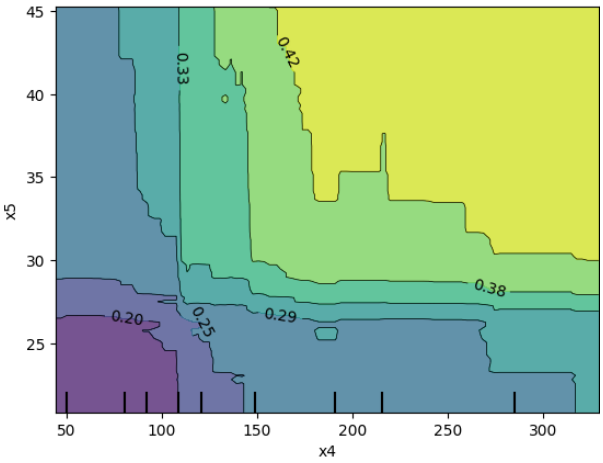
BMI: 6%

- This predictor has a moderate impact in predicting diabetes. BMI is a well-known risk factor, but its impact may be less prominent because it is being captured in other features such as DiabetesPedigreeFunction and Glucose. Its importance may be further emphasized by engineering more/less features containing its relationship to other predictors.

The figure (left) shows the importance of Insulin increases as its value does. When Insulin is under ~110, its importance is minimal. Insulin is the most sensitive from ~110-180. An insulin level of 100 is considered to be moderately high while 200 signals severe insulin



resistance, which is where the line begins to plateau, indicating a minimal increase in prediction importance.

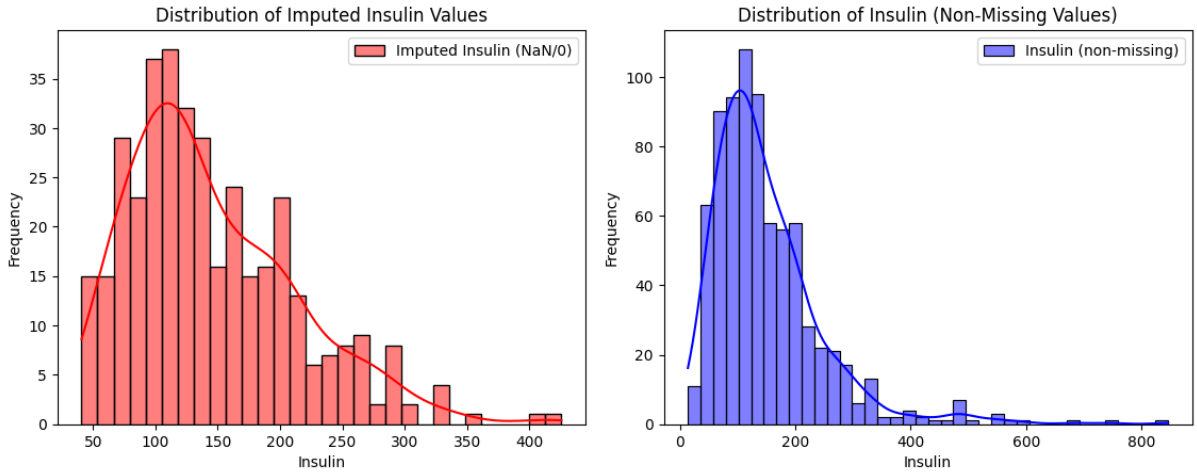


The figure (right) shows that the importance of BMI increases as its value does. BMI is the most sensitive from ~27-~32, and 30 is the threshold for obesity. BMI plateaus after 35, indicating the further increase results in an insignificant increase in importance.

The x-axis is Insulin and the y-axis is BMI. It is clear the prediction that a person has diabetes gets stronger as each axis increases. The sharp change in contour lines means the model predicts a rapid increase in diabetes probability when Insulin exceeds 100, then 150, and when BMI exceeds 30. When the lines begin to flatten, it illustrates that these predictors have reached a threshold in confidence, any further values will have a small increase on the predicted probability. In other words, the model is sensitive to changes in the predictors up until the thresholds.

Overall, Insulin and BMI are strong predictors when paired together. They have different ranges of sensitivity, but it is clear the predicted probability of diabetes increases significantly as both increase. BMI alone has a smaller relevance but is more pronounced when insulin is high. Insulin alone will significantly increase the probability of diabetes.

Insulin is recorded in only half of the observations. Is there a correlation between the target value in the people who are tested vs those who are not tested?



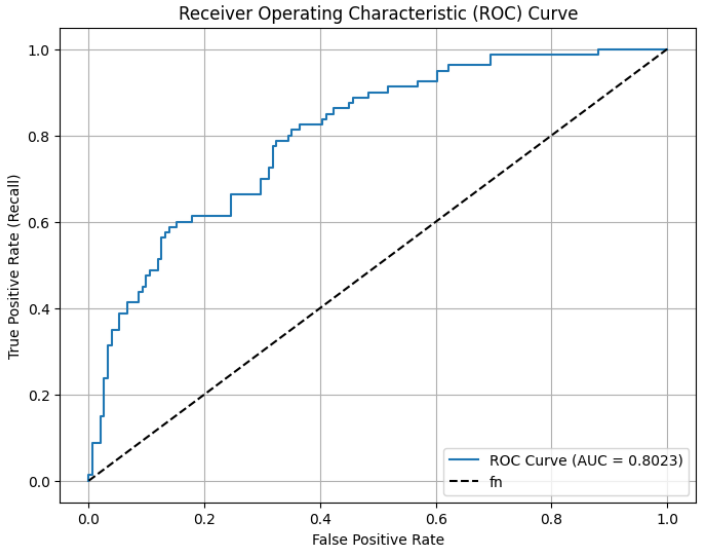
The distribution of the Insulin values for a recorded sample vs an imputed sample is very similar. There are 374 missing insulin values vs 394 recorded values, which is approximately 50%. There are no outliers in the imputed data which will reduce variance when replacing missing values. When combined, there is a p-value of 0.15, indicating there is no significant difference between the splits and the imputation will not introduce a bias when merged into the dataset. After observing the T-statistic of 1.5, we can be assured that the difference in the groups is not significant, nor should the feature be removed.

This question arose out of the curiosity that specific people may have been tested based on external factors such as their appearance (Skin Thickness, BMI, Age), which may swindle the prediction of diabetes. I speculated that some people may have been tested because they *look* healthy. The inverse may have also been true. This assumption may be captured through other predictors, but there is no clear evidence with respect to insulin,

which is further proven through the following Logistic Regression model. We did not use Insulin in the model, instead we replaced it with an engineered binary feature titled Insulin_missing. The model yields the following performance:

Accuracy: 0.75
Variance: 5.23
Recall: 0.60
Precision: 0.65
F1: .62

Compared to the previous Logistic Regression Model using the standard predictors, the performance is similar. The difference in accuracy and variance is negligible. The precision is 7% higher while the recall is 8% lower, yielding an F1 score 2% higher. In terms of accuracy, the classification of insulin does not change, but the rate of positive predictions increases yielding more false positives and negatives. This is not necessarily a bad thing because the goal is to identify as many diabetic people as possible. The rate of true positives vs false positives for the model is illustrated within the ROC curve below.



In conclusion, the analysis of the Pima Indian Diabetes Dataset[3] yields a statistical explanation of the factors that contribute to the alarmingly high rate of type 2 diabetes in women within the community. With the implementation of Random Forest Classification, we further prove that Insulin is the most significant predictor which aligns with the scientific understanding that β-cell dysfunction causes insulin resistance and obesity, which are the leading factors of type two diabetes. BMI plays a strong role in the prediction, but its difference in importance may be due to the value being captured in other predictors such as SkinThickness and Glucose. BMI becomes incredibly relevant when paired with Insulin which emphasizes the importance of engineering features to represent these relationships between other predictors. BMI and Insulin have sensitivity thresholds that prove that the probability of diabetes is high when someone is identified to be obese or have high insulin (resistance) levels.

While exploring several models, it was evident that Random Forest Classification, KNN, and Logistic Regression are the most effective for capturing and predicting diabetes. After cross validating and tuning hyperparameters, Random Forest Classification leads with an accuracy of 80%, but it did struggle with recall as 29% of diabetic people fly under the radar, marked as not diabetic. Gradient boosting, Lasso Regression and PDA were

explored, but they did not improve recall, and they'd need further exploration to make the model perform better.

Approximately 50% of the dataset was missing an Insulin measurement which called for further analysis after imputing. Given the feature importance, Insulin cannot be removed as it is the most important predictor. A contingency table analysis illustrates that the imputed values do not introduce a bias given a p-value of 0.15. A T-statistic of 1.5 further indicates there is no significant difference between the recorded and imputed distributions. Replacing missing values with an engineered binary feature 'Insulin_missing' yields a model of comparable performance with default parameters, proving there is no difference in the people who were tested vs. not tested. Whoever administered the Insulin test did not display any systematic bias for Insulin testing. Insulin is completely independent of Outcome.

Beyond our findings, there are historical and genetic factors that have compounded the risk of type 2 diabetes within the community. The compounding effect of diminishing water rights and droughts has led to the inability to harvest nutrient-rich foods native to the community. Restricted access to healthy foods leads to adverse health issues, which are passed down for generations to come. This is especially magnified by the genetic mutation causing β-cell dysfunction. Living in a food desert intensifies these effects as a community is forced to rely on processed, high-calorie diets.

Citations

- [1]Microsoft Word. (2017). *PPAR - Vol. 7, No. 3, 2017*. Retrieved from Core.ac.uk.
- [2]Ravussin, E., Valencia, M. E., Esparza, J., Bennett, P. H., & Schulz, L. O. (1994). Effects of a traditional lifestyle on obesity in Pima Indians. *Diabetes Care, 17*(9), 1067–1074.
- [3]UCI Machine Learning Repository. (n.d.). *Pima Indians Diabetes Database*. Retrieved from https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database.
- [4]Lugat, V. (n.d.). *Pima Indians Diabetes - EDA & Prediction (0.906)*. Retrieved from https://www.kaggle.com/code/vincentlugat/pima-indians-diabetes-eda-prediction-0-906/notebook.

- My code: https://github.com/woooocoder/machine-learning/blob/main/diabetes-ml/diabetes.ipynb