Data Preparation/Feature Engineering

1. Overview

Data preparation and feature engineering are crucial steps in a machine learning project, especially when it comes to predicting chronic diseases. This is the process of cleaning and transforming raw data before it's used in a machine learning model. In the context of chronic disease prediction, we prepared three different which are diabetes dataset, heart disease dataset and Parkinson disease dataset, we have applied Exploratory Data Analysis techniques to analyzed our data and visualized them.

2. Data Collection

The data used for this study are collected from Kaggle which is an open-source platform for datasets. It's a collection of three different datasets. Each dataset represents 1 type of chronic diseases. Totally we have 3 different chronic diseases which are: diabetes, heart disease and Parkinson diseases. Each dataset size is different from the other.

3. Data Cleaning

The datasets that we collected from Kaggle are almost cleaned datasets. There are no missing values, we checked also the information about the datatypes of our features, all are float and integer data. However, in heart disease dataset, there was one duplicated rows, so we used dropna() panda's functionality to drop one of the duplicated rows. In Parkinson dataset, there is a text column called 'name' so we drop this column because it's not really relevant for our dataset while training the model.

4. Exploratory Data Analysis (EDA)

The total of diabetes dataset is 768 entries and 8 independent features which are: Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age and the Outcome feature which is the target that will going to be trained as y_train and predicted later. For heart disease, the total entries are 303 rows and we have 14 attributes in which one attribute is the target feature that need to be predicted. About Parkinson disease, 195 entries are registred with 24 different features in which 23 are independent and 1 feature represent the target dependent feature.

In this study, each of the 3 datasets has a binary input which means all our targets have 0 and 1 values. To analyse these data, extract insights and make decision about these data, Exploratory Data Analysis(EDA) techniques are used. To visualize our data, we used matplotlib, seaborn and also pandas profiling EDA python libraries.

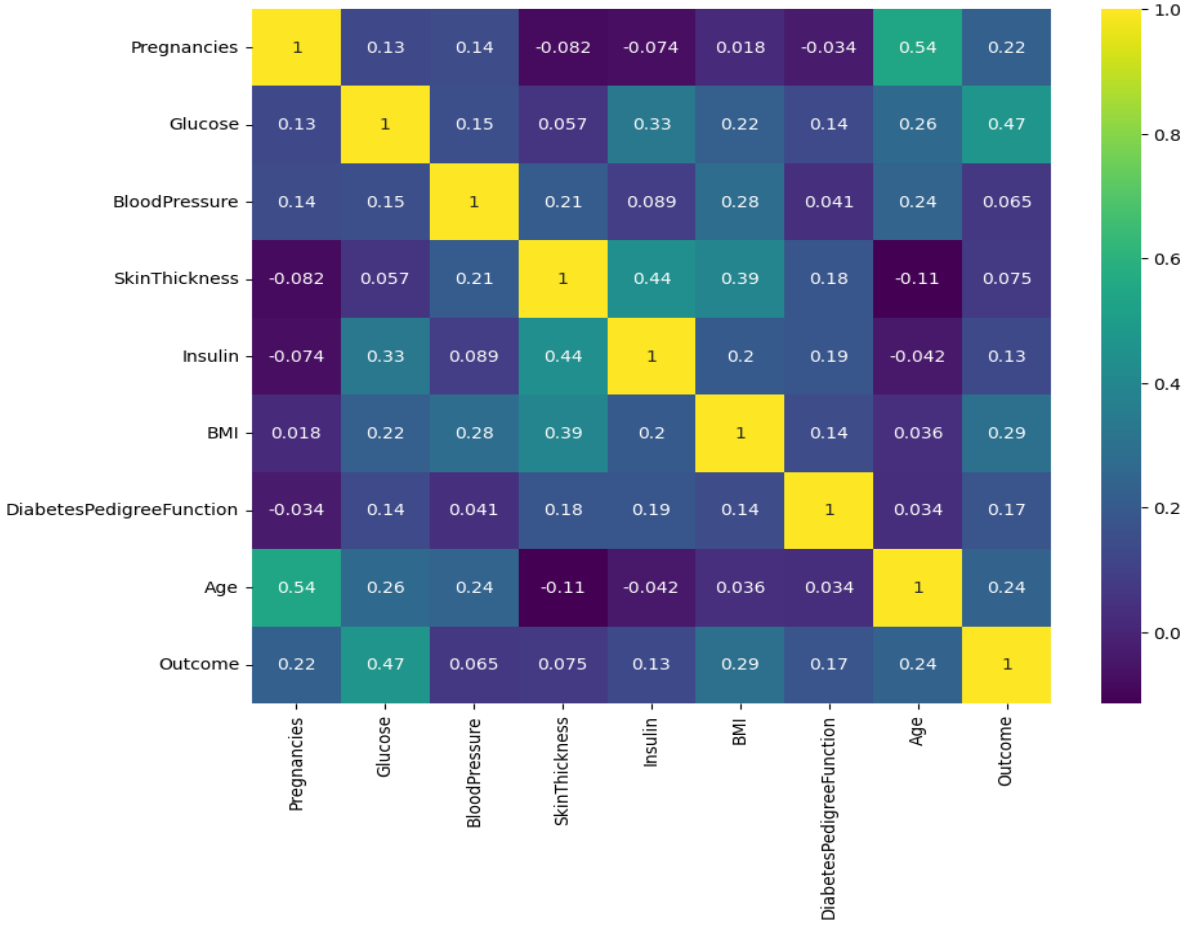


Figure 1-features correlation

This graph shows us the correlation between all features used in diabetes dataset.

Pregnancies is highly overall correlated with Age, SkinThickness is highly overall correlated with Insulin

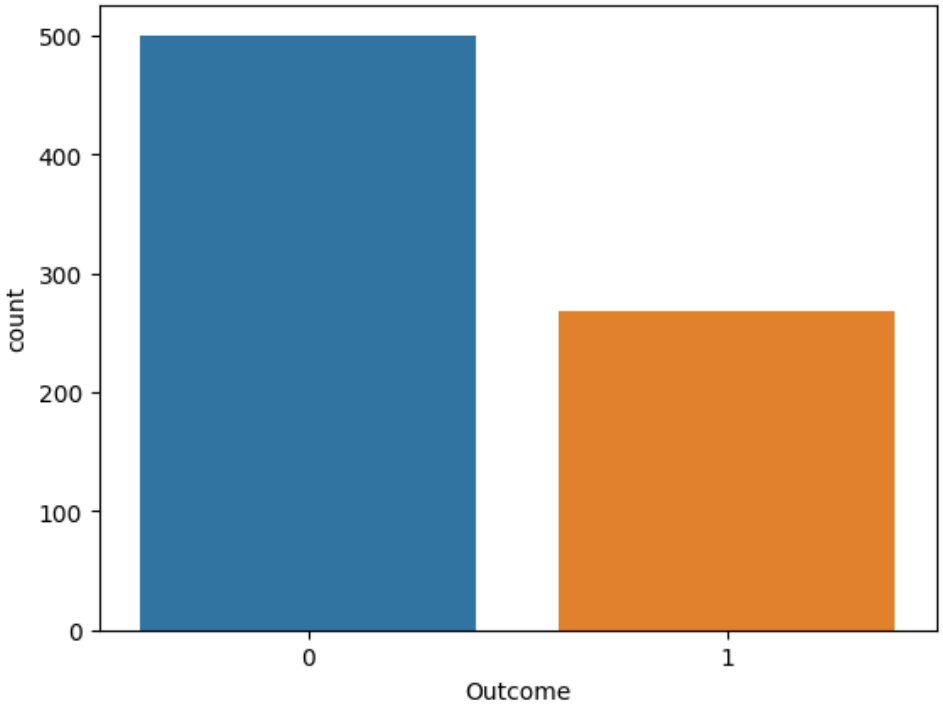


Figure 2-The output label insight of Diabetes disease

According to this graph, the person who are not diabetic are more than the person who have diabetes. Around 500 are non-diabetics and 268 persons are affected by diabetes.

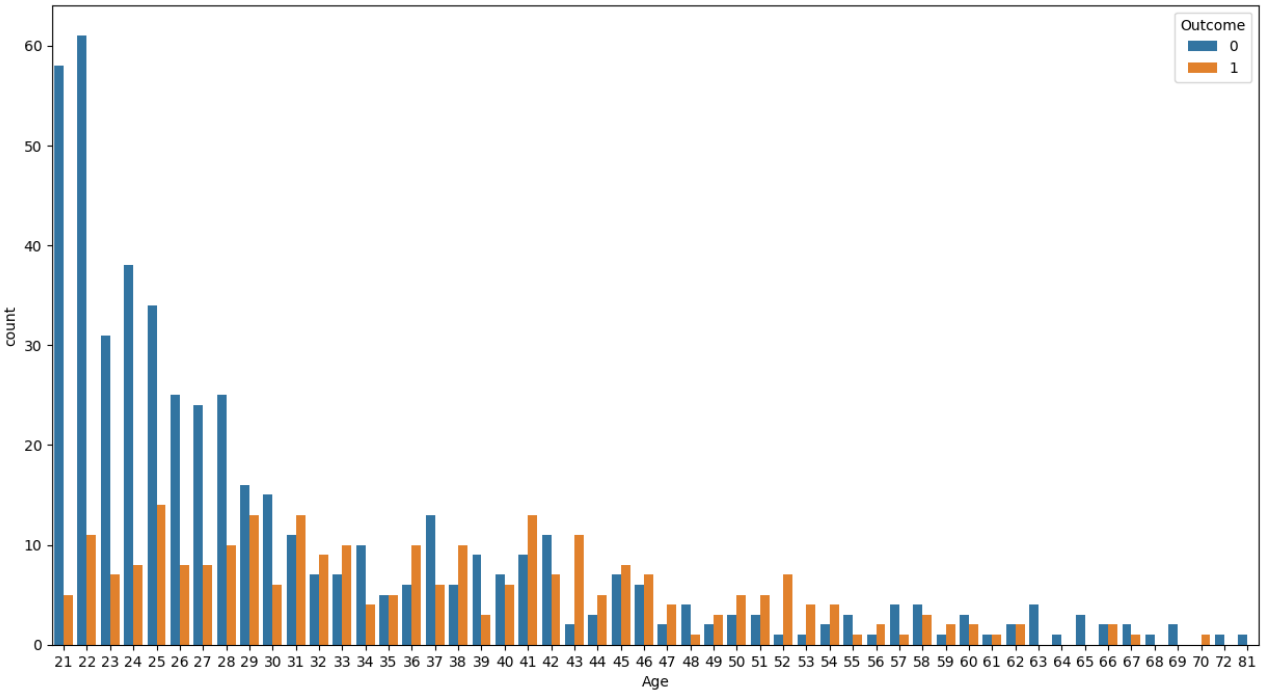
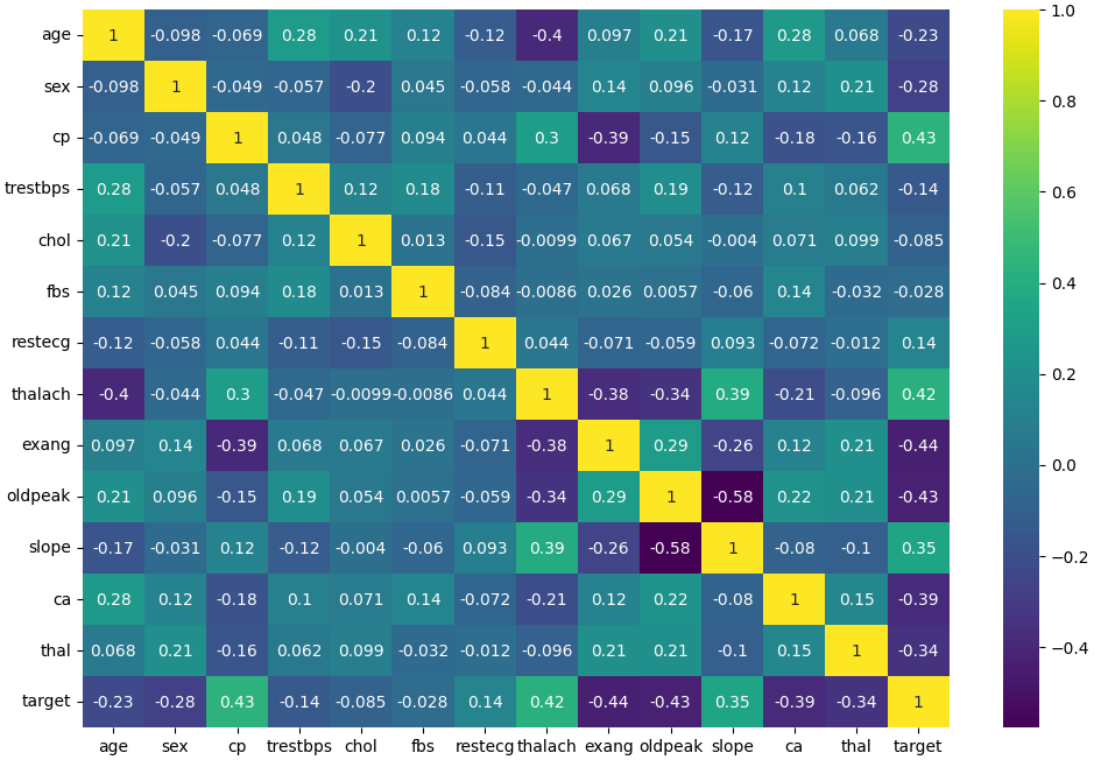


Figure 3- relationship between Ages and diabetes

This graph shows us that diabetes diseases is more shown in people who are more than 30 years old. People between 21 and 30 are less affected. That means that the more the age increase, the more people have risk to be more diabetics.

In heart diseases dataset, the total shape is 303 rows × 14 columns. The output label is a set of 0 and 1 values.



cp is highly overall correlated with target, thal is highly overall correlated with target

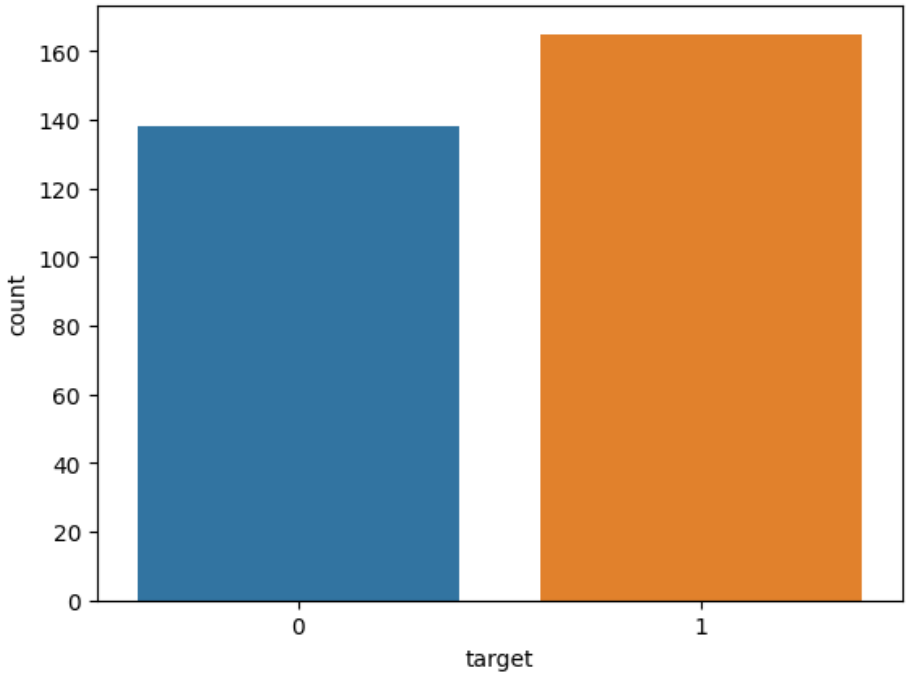


Figure 4- The output label insight of heart disease

The person affected by heart diseases are more than the persons non affected.

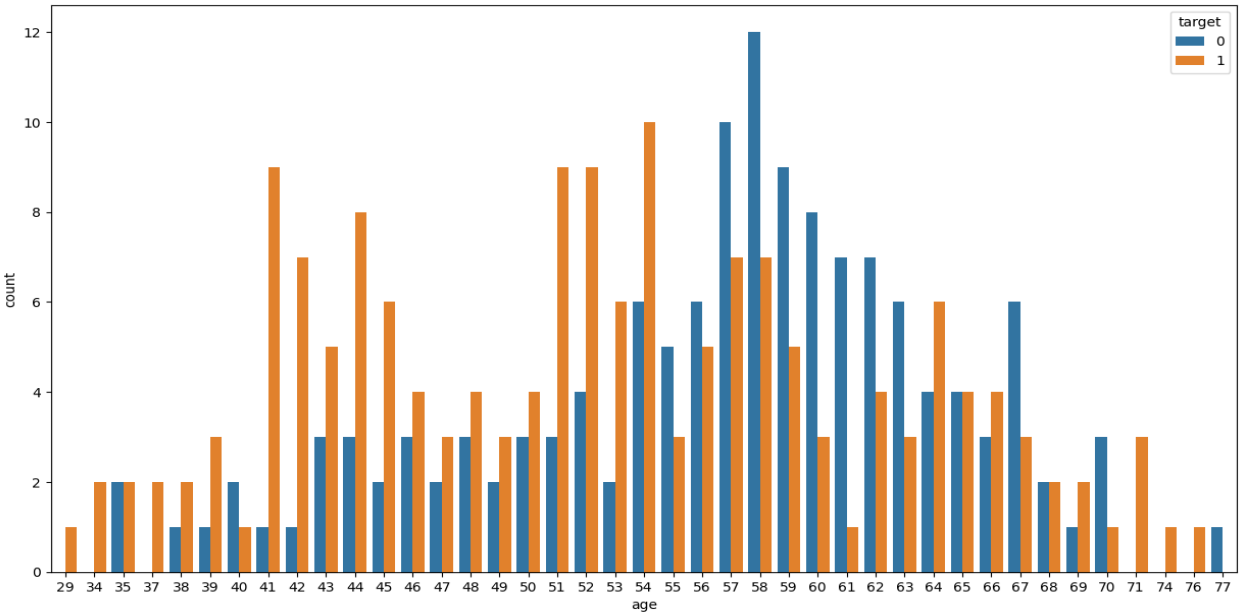


Figure 5-relationship between Ages and heart disease

The insight shows that people between 40 and 60 years old are more likely to be affected by the heart disease and also people who are greater than 65 years.

Parkinson disease is also one of chronical disease that need to be analyzed. In our study, the total entries are 195 and the number of features is 24, these features are more medical features.

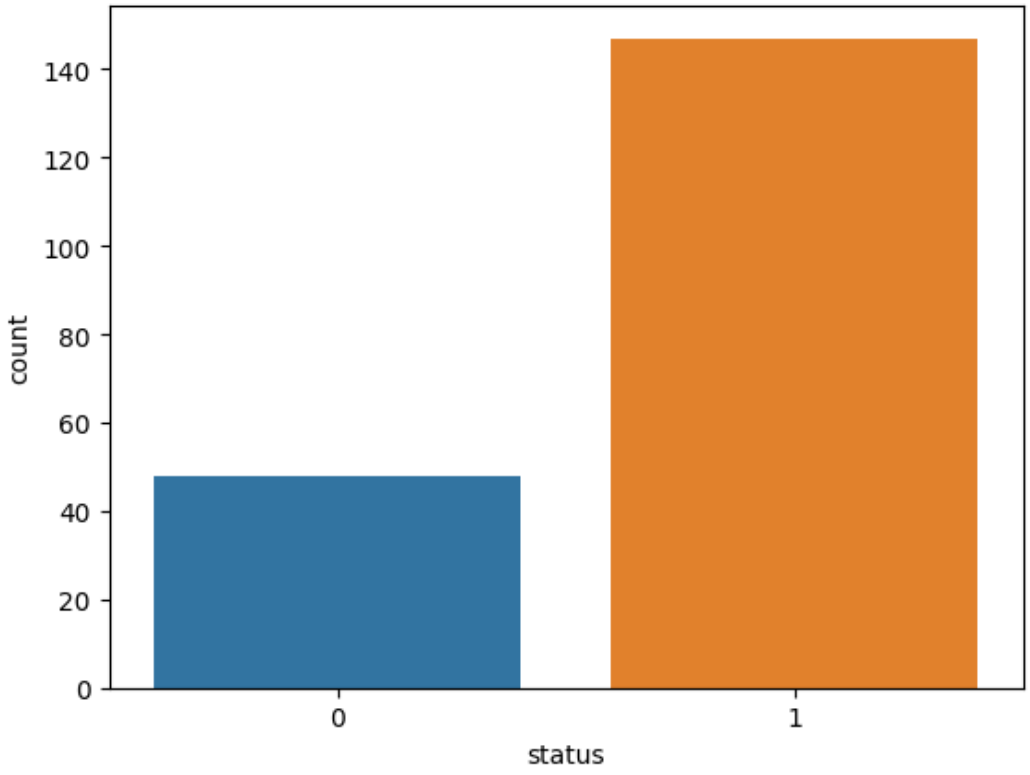


Figure 6-The output label insight of Parkinson disease

This is the count plot of the output label. 1 represents the person affected by Parkinson disease and 0 represents the persons non affected. The number of people affected by the disease is very high according to the result.

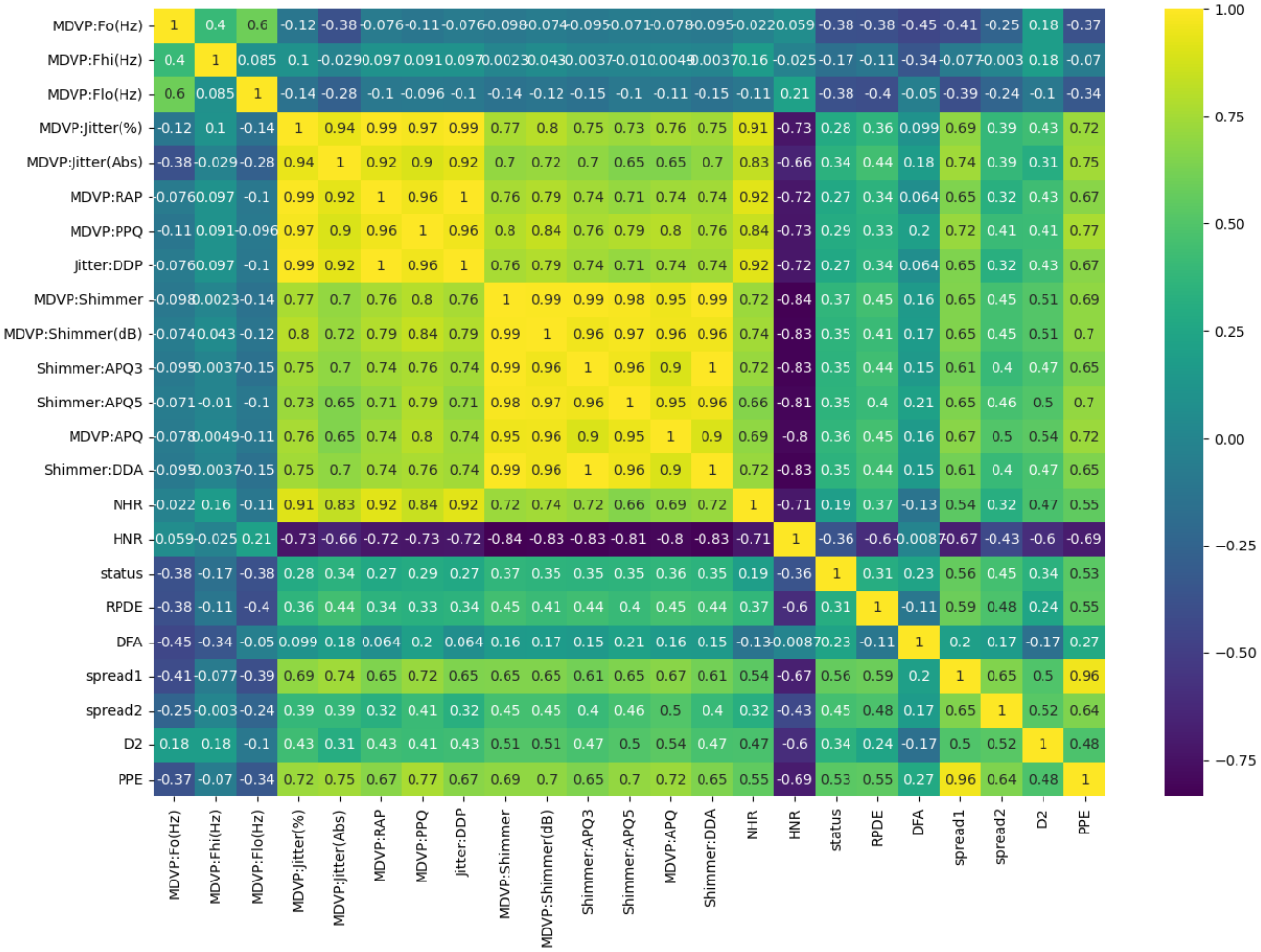


Figure 7- Parkinson disease features correlation

The picture above shows us the correlations of the features used to predict the Parkinson disease. Looking at this graph we figure out a high number of features hihly correlated. For example MDVP:Fo(Hz) is highly overall correlated with MDVP:Fhi(Hz), MDVP:Flo(Hz) is highly overall correlated with status, MDVP:Jitter(%) is highly overall correlated with MDVP:Jitter(Abs) and 13 other fields, NHR is highly overall correlated with MDVP:Jitter(%) and 15 other fields etc. By using Pandas Profiling, we got also an alert of these high features correlation.

Figure 8- Pandas Profiling features correlation alert

The high correlation between the features is too much in this data, so we need to do some feature engineering for that.

5. Feature Engineering and Data Transformation

Feature Engineering involves creating meaningful variables (or "features") that will improve the performance of the machine learning model. In chronic disease prediction, we used some feature engineering specially in diabetes dataset and Parkinson dataset.

In diabetes dataset, we used Standard Scaler to standardize or normalize the features of a dataset. Standardization is the process of rescaling the features so that they have the properties of a standard normal distribution with a mean of 0 and a standard deviation of 1. This transformation is particularly useful when dealing with algorithms that are sensitive to the scale of the input features. Here are a few reasons why Standard Scaler is used in machine learning to make diseases prediction:

- Some machine learning algorithms are sensitive to the scale of the input features. For example, distance-based algorithms (like k-nearest neighbors) and gradient descent-based optimization algorithms like support vector machines and linear regression can be influenced by the magnitude of the features.
- Machine learning algorithms such as Logistic Regression and SVM make assumptions about the distribution of the features. Standardizing features to have a mean of 0 and a standard deviation of 1 makes them more closely approximate a standard normal distribution, which can be beneficial for certain algorithms.

In Parkinson dataset, we have too much features and most of them are highly correlated, so we need to do some features selection here. In this case we used Principal Component Analysis (PCA) with Standard Scaler features normalization technique. PCA is a dimensionality reduction technique that involves finding the principal components of the data. Standardizing features is important in PCA because it is based on covariance matrices, and having features on different scales can result in biased principal components.

Model Exploration

1. Model Selection

Predicting diseases is recognized as a classification problem. In our study, the outputs are 0 and 1. 0 represents the person not affected by the disease and 1 is the result of the affected disease. It's the case for all of our three different datasets. In this case, LogisticRegression seems to be the first choice to build our models. But we need also to use others classifications algorithms such as SVM or KNN or also ensemble techniques such as RandomForest Classifier or XGBoost algorithms to choose the best accuracy. The models that perform well are the model that have the highest accuracy.

Logistic Regression is simple and computationally efficient that provides probabilities for outcomes. One of the strengths of this algorithms is that it's robust to overfitting, especially with a small number of features. However, Logistic Regression assumes a linear relationship between features and the log-odds of the outcome and it may not perform well with highly non-linear data. Logistic Regression is highly interpretable as it directly models the probability of the output.

Support Vector Machines (SVM) is effective in high-dimensional spaces and versatile due to different kernel functions. It's also resistant to overfitting. But one of the weaknesses is that the computationally is height, especially with large datasets and the model parameters (e.g., the choice of the kernel) can be challenging to interpret.

k-Nearest Neighbors (KNN) is a non-parametric and flexible supervised learning algorithm that is intuitive and easy to understand. It can capture complex decision boundaries. However, it's computationally expensive during prediction, especially with large datasets and also sensitive to irrelevant or redundant features.

Random Forest: The strength of RandomForest is that this algorithm is Robust and less prone to overfitting compared to individual decision trees. It can handle a large number of features and provides feature importance scores. The Weaknesses are Lack of interpretability compared to decision trees and the computation is sometimes more expensive.

XGBoost is a Gradient boosting framework used for a better predictive performance. It has the capacity to handle missing data well and it is robust to outliers. It's also an efficient and scalable algorithm. The weaknesses are that it can prone to overfitting, especially with insufficient regularization and requires tuning of hyperparameters.

2. Model Training

Before training any model, we analyzed each dataset, use visualization techniques such as Matplotlib, Pandas Profiling and Seaborn to visualize our data and have a deep understanding about the different diseases. After doing Exploratory data analysis, to classify these diseases, Machine Learning classification algorithms such as Logistic Regression, Support Vector

Machine, KNeareastNeighbors Classifiers and Bagging ensemble technique such as Random Forest and Boosting technique such as XGBoost Classifier are used to build three different models.

For heart disease dataset, the best model was found with Logistic Regression, the score obtained is 86% on the training data and 89% on the test dataset. 20% of our dataset is considered for evaluating and testing the model. These are the results obtained after using Hyper Parameter Tuning to optimize our model. The hyperparamter Tuning technique used is GridSearch. With this technique, we combined different algorithms together with their different hyperparameters and the best algorithms was Logistic Regression, RandomForest and Support Vector Machine (SVM).

	model	Best score	best_score
0	logistic_regression	0.834609	C= 0.23, max_iter= 100
1	random_forest	0.834609	max_depth=2, max_leaf_nodes= 6
3	SVM	0.822194	C= 100, kernel= 'linear'
4	xg_boost	0.797619	max_depth=9, max_leaf_nodes=3
5	KNN	0.702381	metric= 'manhattan', 'n_neighbors': 7

Table 1- sorted models results after hyper parameter tuning

For diabetes disease dataset, the algorithms used are Logistic Regression, SVM, RandomForest and XGboost. XGBoost gave the best score after Hyper Parameter Tuning. The score obtained by this algorithm is 96% on the training dataset and 77% on the test dataset. 20% of the dataset was used to test the model and 80% was used for training the data. We then used StandardScaler normalization technique to normalize our data. The training score with RandomForest gave us 91% and the test score with the same algorithm gave 71% as a result.

	model	best_score	best_params
0	svm	0.778529	{'C': 0.1, 'kernel': 'linear'}
1	knn	0.750860	{'n_neighbors': 7}
2	random_forest	0.780195	{'max_depth': 6, 'max_features': 'log2', 'n_es...
3	xg_boost	0.747608	{'n_estimators': 5}
4	logistic_regression	0.778529	{'C': 1, 'n_jobs': -1}

Table 2- models's results after hyper parameter tuning for diabetes disease

For Parkinson disease dataset, Dimensionality Reduction Technique known as (Principal Component analysis (PCA) was used to reduce the dimension of our features by keeping other's features importance.

For Parkinson disease dataset, the Machine Learning techniques used are: Logistic Regression, SVM, KNN, RandomForest and XGBoost. All these algorithms gave a good result but the best one obtained is with KNN. The scores obtained using KNN are 92 % on the training dataset and 90% on the test data. The second best algorithm for this data is SVM. The training score with SVM is: 85% and the test score with SVM is 87%. 80% of the data is used for training and 20% was used for testing and evaluation. GridSearch hyper Parameter tuning technique was used to perform the result and get this final score.

	model	best_score	best_params
0	svm	0.910081	{'C': 1, 'gamma': 1, 'kernel': 'rbf'}
1	knn	0.897581	{'metric': 'manhattan', 'n_neighbors': 11, 'we...
2	random_forest	0.878024	{'max_depth': 10, 'max_leaf_nodes': 9, 'n_esti...
3	xg_boost	0.884476	{'max_depth': 6, 'max_leaf_nodes': 3, 'n_estim...
4	logistic_regression	0.833266	{'C': 0.012742749857031334, 'max_iter': 100, '...

Table 3-Parkinson disease models results after hyper parameter tuning

3. Model Evaluation

To analyze the performance of our models, there are some metrics used such as confusion matrix, accuracy score and classification reports to help us evaluate our models.

Looking at the model performances of diabetes disease dataset, the accuracy is 0.77 for the test data, and the recall and f1-score are not really high. They are 0.61 and 0.65 respectively for the class 1. This is most probability due to the imbalanced dataset that we have between 0 and 1 classes. This may lead to some wrongs predictions especially False negative results would need to be decreased in this case.

	precision	recall	f1-score	support
0	0.80	0.86	0.83	100
1	0.70	0.61	0.65	54
accuracy			0.77	154
macro avg	0.75	0.74	0.74	154
weighted avg	0.77	0.77	0.77	154

Table 4-Classification_report for Diabetes data with XGBoost

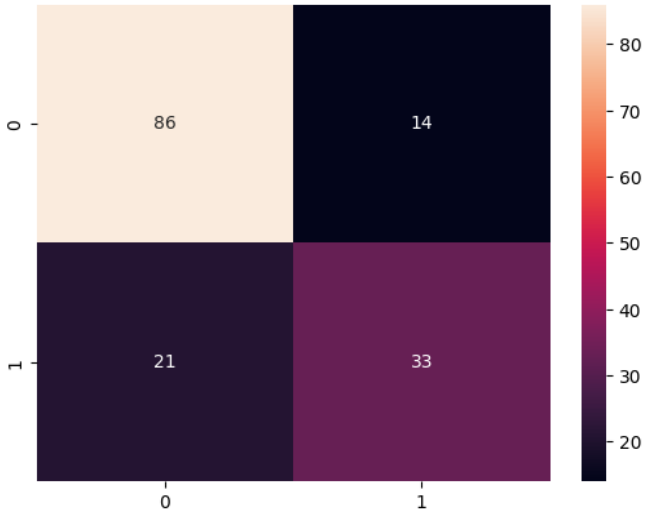


Figure 9-Confusion matrix for Diabetes data with XGBoost

The result of the model performance of heart disease dataset looks good when using Logistic Regression and we see the Precision, recall, f1-score and accuracy results for both classes 0 and 1.

	precision	recall	f1-score	support
0	0.89	0.86	0.88	29
1	0.88	0.91	0.89	32
accuracy			0.89	61
macro avg	0.89	0.88	0.88	61
weighted avg	0.89	0.89	0.89	61

Table 5- Classification_report for heart disease with LogistciRegression

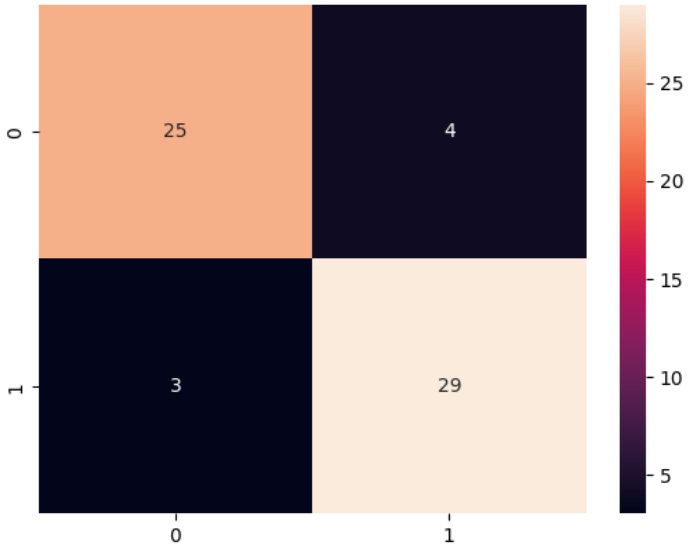


Figure 10-Confusion matrix for heart disease with LogistciRegression

When we look at the Parkinson model built, we see that the model performed well, especially for the class 1 but we might get some wrongs results because of the result of the performances the class 0. That might not really affect our model result. The accuracy also is a good accuracy 0.90 on the test data.

	precision	recall	f1-score	support
0	0.71	0.71	0.71	7
1	0.94	0.94	0.94	32
accuracy			0.90	39
macro avg	0.83	0.88	0.83	39
weighted avg	0.90	0.89	0.90	39

Table 6- Classification_report for Parkinson disease with KNN

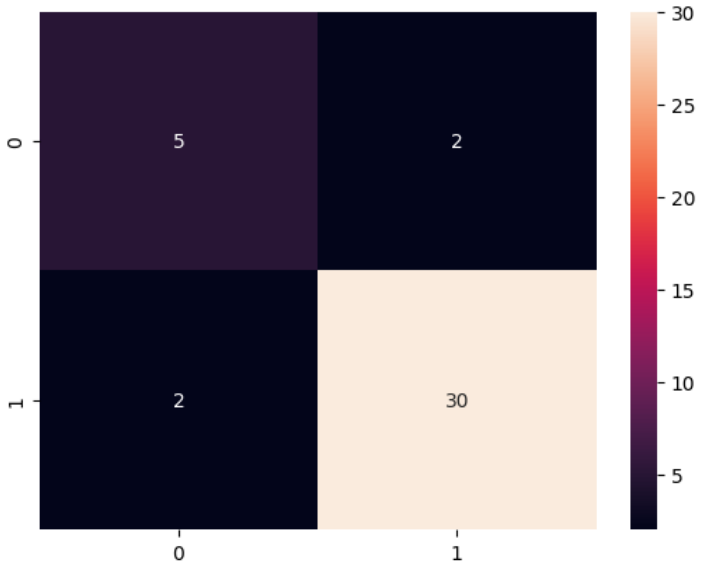


Figure 11- Classification_report for Parkinson disease with KNN

4. Code Implementation

I uploaded all codes in github as files