GROUP 1

Utilizing Bioinformatics to Predict Diabetes

A Computational Biology Presentation

By:
Brandon Steven - 2501974031
Jonathan Ivan - 2501963980
Julius Ferdinand - 2501962542
Rayes Jordan Pradana - 2502033102

Biology

What is "Bioinformatics"?

- From the combination of "Bio" and "Informatics"
- Utilizing computational features of technology in the field of biology

Biology

What is "Diabetes"?

Diabetes is one of the biggest and fatal diseases all over the world, there are 2 types of diabetes, one is autoimmune diseases, when our own immune system attack ourself, in this case the body wont produce any insulin.
The other one is when, the body cant produce enough insulin.

Causes of Diabetes

The causes of diabetes is high levels of bloodsugar, this is caused by the body can't use glucose to get energy to the cell.

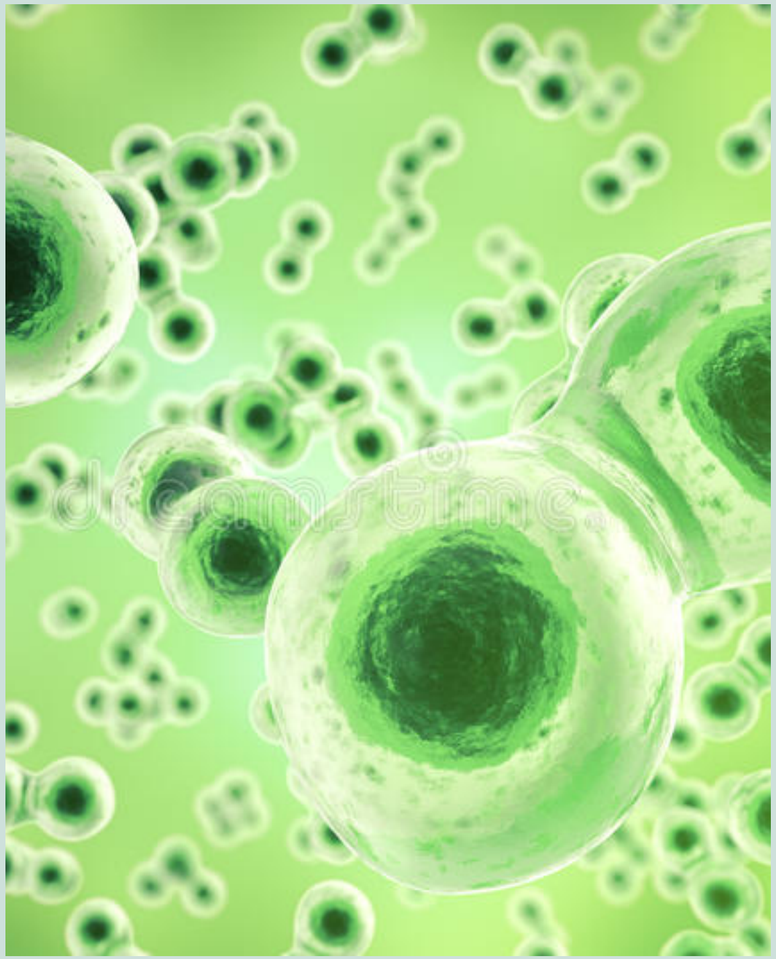
So, sugar level on the blood is too high, and that will cause diabetes, there are organs that are often attacked by the diabetes, such as heart, kidney, eyes, and the nerves.

Factors.

- Overweight
- Too much fat on the belly parts
- Unhealthy lifestyle
- Diabetes type 2 run in the family
- Certain races could get diabetes more than other races
- Age over 45 years old

symptoms

- Thirsty all the time.
- Frequent peeing
- Exhausted easily , loss of vision.
- Weight loss with unknown factors
- Infection on the body. are tools that can be used as lectures.



What's the use of Bioinformatics in this topic?

- Predicting the probability of diabetes for a patient using various vactors.
- Creating a visualization of the result for easier understanding.
- Advantages & disadvantages

Related Works

Research conducted on the same topic while implementing similar methods

A Comparative Analysis of Early Stage Diabetes Prediction using Machine Learning and Deep Learning Approach

Model(s) : XGB, Random Forest, Decision Tree, KNN, SVM, ANN, Multi-layer Perceptron, and LSTM.

Metric(s) : Accuracy, Precision, Recall, F1-Score, Roc-Auc Score

Best result : XGB Classification and Random Forest

Analysis and Prediction of Diabetes Using Machine Learning

Model(s) : Decision Tree, Naïve Bayes , KNN (K = 1), and KNN (K = 3).

Metric(s) : Accuracy

Best result : Decision Tree (After bootstrapping)

Biology

Diabetes Prediction

Direct Experimentation with Various Models

Training various different models using a dataset to experiment with the utility of the bioinformatic

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

Python Code Biology Diabetes Predicition Using Machine Learning Classification

```python
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn.metrics import accuracy_score, f1_score, log_loss, recall_score, precision_score
import seaborn as sns
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report, confusion_matrix
```

Python Code Biology Diabetes Predicition Using Machine Learning Classification

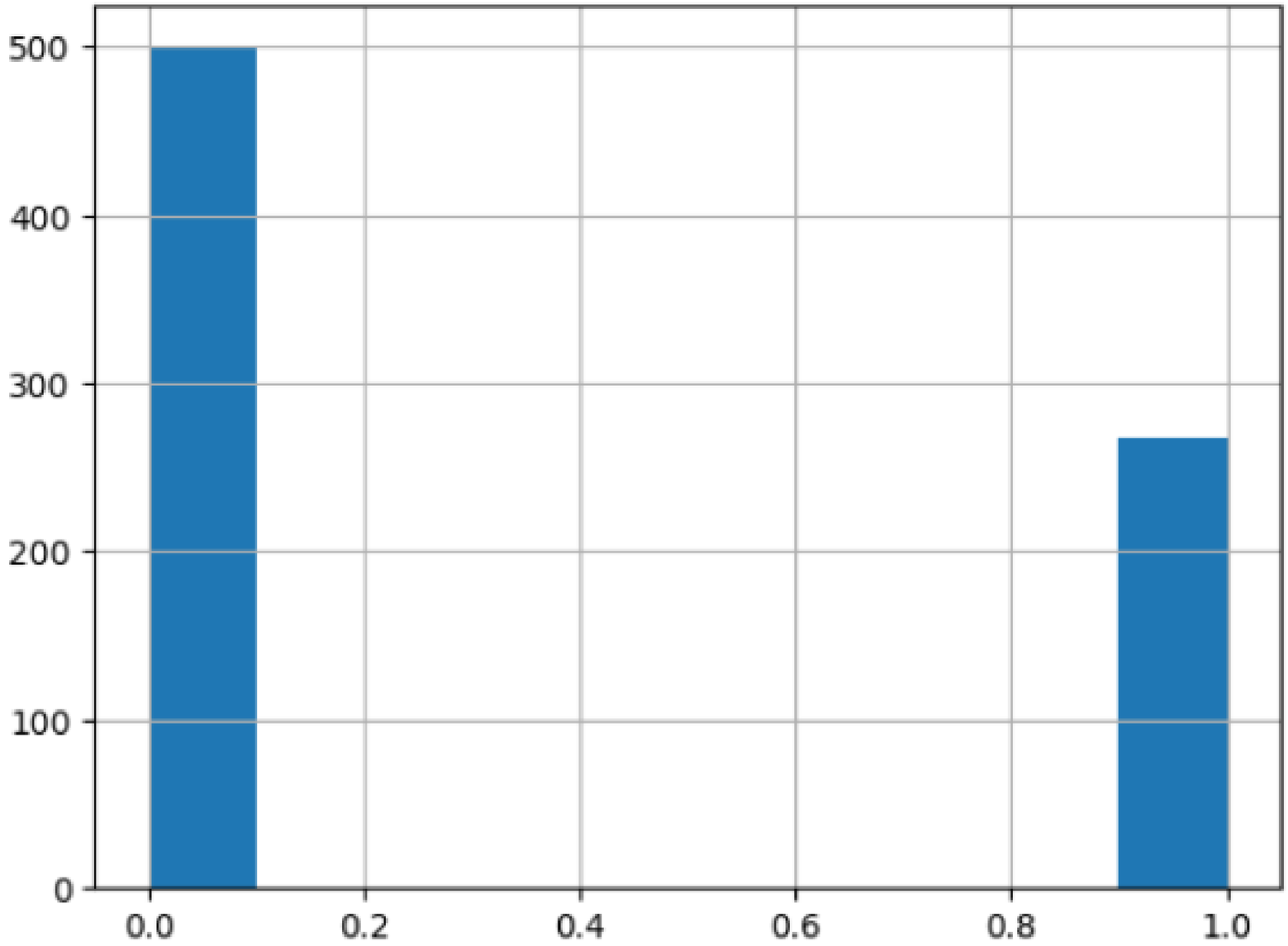
```
data = pd.read_csv('diabetes.csv')
data.info()
```

Python Code Biology Diabetes Predicition Using Machine Learning Classification

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Pregnancies               768 non-null    int64  
 1   Glucose                   768 non-null    int64  
 2   BloodPressure             768 non-null    int64  
 3   SkinThickness             768 non-null    int64  
 4   Insulin                   768 non-null    int64  
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64  
 8   Outcome                   768 non-null    int64  
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

Python Code Biology Diabetes Predicition Using Machine Learning Classification

Histogram



Heatmap


Pregnancies 1 0.13 0.14 -0.082 -0.074 0.018 -0.034 0.54 0.22
Glucose 0.13 1 0.15 0.057 0.33 0.22 0.14 0.26 0.47
BloodPressure 0.14 0.15 1 0.21 0.089 0.28 0.041 0.24 0.065
SkinThickness -0.082 0.057 0.21 1 0.44 0.39 0.18 -0.11 0.075
Insulin -0.074 0.33 0.089 0.44 1 0.2 0.19 -0.042 0.13
BMI 0.018 0.22 0.28 0.39 0.2 1 0.14 0.036 0.29
DiabetesPedigreeFunction -0.034 0.14 0.041 0.18 0.19 0.14 1 0.034 0.17
Age 0.54 0.26 0.24 -0.11 -0.042 0.036 0.034 1 0.24
Outcome 0.22 0.47 0.065 0.075 0.13 0.29 0.17 0.24 1
Pregnancies
Glucose
BloodPressure
SkinThickness
Insulin
BMI
DiabetesPedigreeFunction
Age
Outcome
1.0
0.8
0.6
0.4
0.2
0.0

```
from imblearn.over_sampling import SMOTE

sm = SMOTE(random_state=42)

features = data.drop(columns=['Outcome'])
label = data['Outcome']

features, label = sm.fit_resample(features, label)

scaler = StandardScaler()
scaled_features = scaler.fit_transform(features)
```

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(scaled_features,
                                                    label,
                                                    test_size=0.2,
                                                    random_state=3)
```

Splitting 20% of the dataset for testing and 80% for training

```python
from sklearn.ensemble import RandomForestClassifier

clf_rf = RandomForestClassifier()
clf_rf.fit(X_train, y_train)

y_pred_rf = clf_rf.predict(X_test)
clf_rf_acc = get_model_accuracy(clf_rf, X_test, y_test)
clf_rf_prec = get_model_precision(clf_rf, y_pred_rf, y_test)
clf_rf_recall = get_model_recall(clf_rf, y_pred_rf, y_test)
clf_rf_f1 = get_model_f1(clf_rf, y_pred_rf, y_test)

print(f'Random Forest Testing Accuracy : {clf_rf_acc:.4}')
print(f'Random Forest Testing Precision : {clf_rf_prec:.4}')
print(f'Random Forest Testing Recall : {clf_rf_recall:.4}')
print(f'Random Forest Testing F1-score : {clf_rf_f1:.4}')
```

```
Testing Accuracy : 0.84
Testing Precision : 0.8505
Testing Recall : 0.8505
Testing F1-score : 0.8505
```

```python
from sklearn.ensemble import VotingClassifier

estimators = [('Random Forest', clf_rf),
              ('Gradient Boost', clf_gb),
              ('Xgradient boost', clf_xgb),
              ('Logistic Regression', clf_lr)]
clf_vote = VotingClassifier(estimators, voting='hard')
clf_vote.fit(X_train, y_train)
y_pred_vote = clf_vote.predict(X_test)

clf_vote_acc = get_model_accuracy(clf_vote, X_test, y_test)
clf_vote_prec = get_model_precision(clf_vote, y_pred_vote, y_test)
clf_vote_recall = get_model_recall(clf_vote, y_pred_vote, y_test)
clf_vote_f1 = get_model_f1(clf_vote, y_pred_vote, y_test)

print(f'Ensemble Voting Testing Accuracy : {clf_vote_acc:.4}')
print(f'Ensemble Voting Testing Precision : {clf_vote_prec:.4}')
print(f'Ensemble Voting Testing Recall : {clf_vote_recall:.4}')
print(f'Ensemble Voting Testing F1-score : {clf_vote_f1:.4}')
```

```
Voting Testing Accuracy : 0.81
Voting Testing Precision : 0.7944
Voting Testing Recall : 0.8416
Voting Testing F1-score : 0.8173
```

```
params_rand = {'n_estimators': [50, 100, 200, 300, 400],
     'max_depth': [None, 5, 10, 15, 20],
     'min_samples_split': [2, 5, 10, 20],
     'criterion': ['gini', 'entropy']}

grid_search_rf = GridSearchCV(RandomForestClassifier(),
                              param_grid=params_rand,
                              refit=True,
                              verbose=3,
                              scoring="accuracy",
                              n_jobs=-1)
```

```
grid_search_rf.fit(X_train, y_train)

grid_predicitons_rf = grid_search_rf.predict(X_test)

grid_search_rf_acc = get_model_accuracy(grid_search_rf, X_test, y_test)
grid_search_rf_prec = get_model_precision(grid_search_rf, grid_predicitons_rf, y_test)
grid_search_rf_recall = get_model_recall(grid_search_rf, grid_predicitons_rf, y_test)
grid_search_rf_f1 = get_model_f1(grid_search_rf, grid_predicitons_rf, y_test)

print(f'RF with GridSerchCV Accuracy: {grid_search_rf_acc:.4}')
print(f'Random Forest Testing Precision : {grid_search_rf_prec:.4}')
print(f'Random Forest Testing Recall : {grid_search_rf_recall:.4}')
print(f'Random Forest Testing F1-score : {grid_search_rf_f1:.4}')
print(classification_report(grid_predicitons_rf, y_test))

title = 'Confution Matrix of Random Forest with GridSearchCV'
plot_confusion_matrix(y_test, grid_predicitons_rf, 'Greens', title)
```

```
Fitting 5 folds for each of 200 candidates, totalling 1000 fits
RF with GridSerchCV Accuracy: 0.86
Random Forest Testing Precision : 0.8879
Random Forest Testing Recall : 0.8559
Random Forest Testing F1-score : 0.8716
              precision    recall  f1-score   support

           0       0.83      0.87      0.85        89
           1       0.89      0.86      0.87       111

    accuracy                           0.86       200
   macro avg       0.86      0.86      0.86       200
weighted avg       0.86      0.86      0.86       200
```

Comparasion of Random Forest before and after Hyperparameter Tuning

Before

Accuracy :84%
Precission : 85.05%
Recall : 85.05%
F1-Score : 85.05%

After

Accuracy :86%
Precission : 88.79%
Recall : 85.59%
F1-Score : 87.16%

Link to the experiment and the dataset used

Google Colab :
https://colab.research.google.com/drive/15TgaFRuHwJFAwtRXYeAJaEl32Zh3PM4r#scrollTo=FK2DwIj4FmMm

Dataset :
https://www.kaggle.com/datasets/whenamancodes/predict-diabities

Experiment

References

- Refat, M. A. R., Al Amin, M., Kaushal, C., Yeasmin, M. N., & Islam, M. K. (2021, October). A Comparative Analysis of Early Stage Diabetes Prediction Using Machine Learning and Deep Learning Approach. *2021 6th International Conference on Signal Processing, Computing and Control (ISPCC)* (pp. 654-659).
- Saru, S., & Subashree, S. (2019). Analysis and Prediction of Diabetes Using Machine Learning. *International Journal of Emerging Technology and Innovative Engineering*, 5(4).