Diabetes Classification using ML

Ashish Kumar

Date: 30/01/2022

1. Abstract:

Diabetes is one of the major health challenges faced by world population. In today's world every 1 person out of 14 is suffering from diabetes. Diabetes is a metabolic disease characterized by high blood sugar. The main clinical types are type 1 diabetes and type 2 diabetes. Now, the proportion of young people suffering from type 1 diabetes has increased significantly. Type 1 diabetes is chronic when it occurs in childhood and adolescence, and has a long incubation period. The early symptoms of the onset are not obvious, which may lead to failure to detect in time and delay treatment. Long-term high blood sugar can cause chronic damage and dysfunction of various tissues, especially eyes, kidneys, heart, blood vessels and nerves. Therefore, the early prediction of diabetes is particularly important. In this paper, we use supervised machine-learning algorithms to train on the actual data of diabetic patients. By using different machine learning model on the data set, we can choose the best model to accurately classify the person at potential risk.

2. Problem statement:

The incidence of diabetes increased from 11.3 million (95% UI 10.6–12.1) in 1990 to 22.9 million (21.1–25.4) in 2017, with a 102.9% increase. People losing their life due to diabetes is also increasing. India stands at 1st place in case of people dying due diabetes around the world. If we are able to know diabetes type and severity at early stage then it is possible to guide people to take proper diet and maintain lifestyle to avoid complex process like medical and surgical and live a less stress and happy life. To attain this goal, we will be using machine learning algorithm to predicts diabetes in patient. Machine learning can make such prediction easily and more accurately which helps doctor to guide patient more easily and correctly.

3. Market / Customer/ Business Need Assessment:

In India only, we have more than 61 million people living with diabetes. There are numbers of issues which are faced by patient and doctor. Treatment of diabetes and its complications is a major challenge in India owing to several issues, including sociocultural factors, lack of appropriate facilities for diabetes care, an inadequate health system, poor monitoring and follow-up of patients, and problems in

implementing effective management and educational strategies. These are some common challenges faced by diabetes patient such as:

- Regular glucose testing
- Difficult remember medication
- Managing time for exercise
- Not seeing result even after following all diet and exercise plan
- Stress and Depression

In this paper, I will focus on identifying the diabetes and the stage to determine whether the patient condition. So that accordingly we will able to help doctor to provide best treatment process for patient.

4. Target Specifications and Characterization

The proposed system/service is built to provide better prescription, routine checkup and efficient monitoring system for both doctors and patient. Patient will be able to view information related to their glucose level, diet and health very easily. For doctor its very hassle task to maintain record and follow-up with patient. We want to provide one-point solution for both doctors and patient. So, that doctors can easily check their patient and help them more closely.

5. External Search (information sources/references)

The dataset can be found on the Kaggle. his dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective is to predict based on diagnostic measurements whether a patient has diabetes.

Dataset Origin:

https://www.kaggle.com/mathchi/diabetes-data-set

Sources:

(a) Original owners: National Institute of Diabetes and Digestive and Kidney Diseases
(b) Donor of database: Vincent Sigillito (vgs@aplcen.apl.jhu.edu)

Let’s view our dataset

```
diabetes_df.head(5)
```

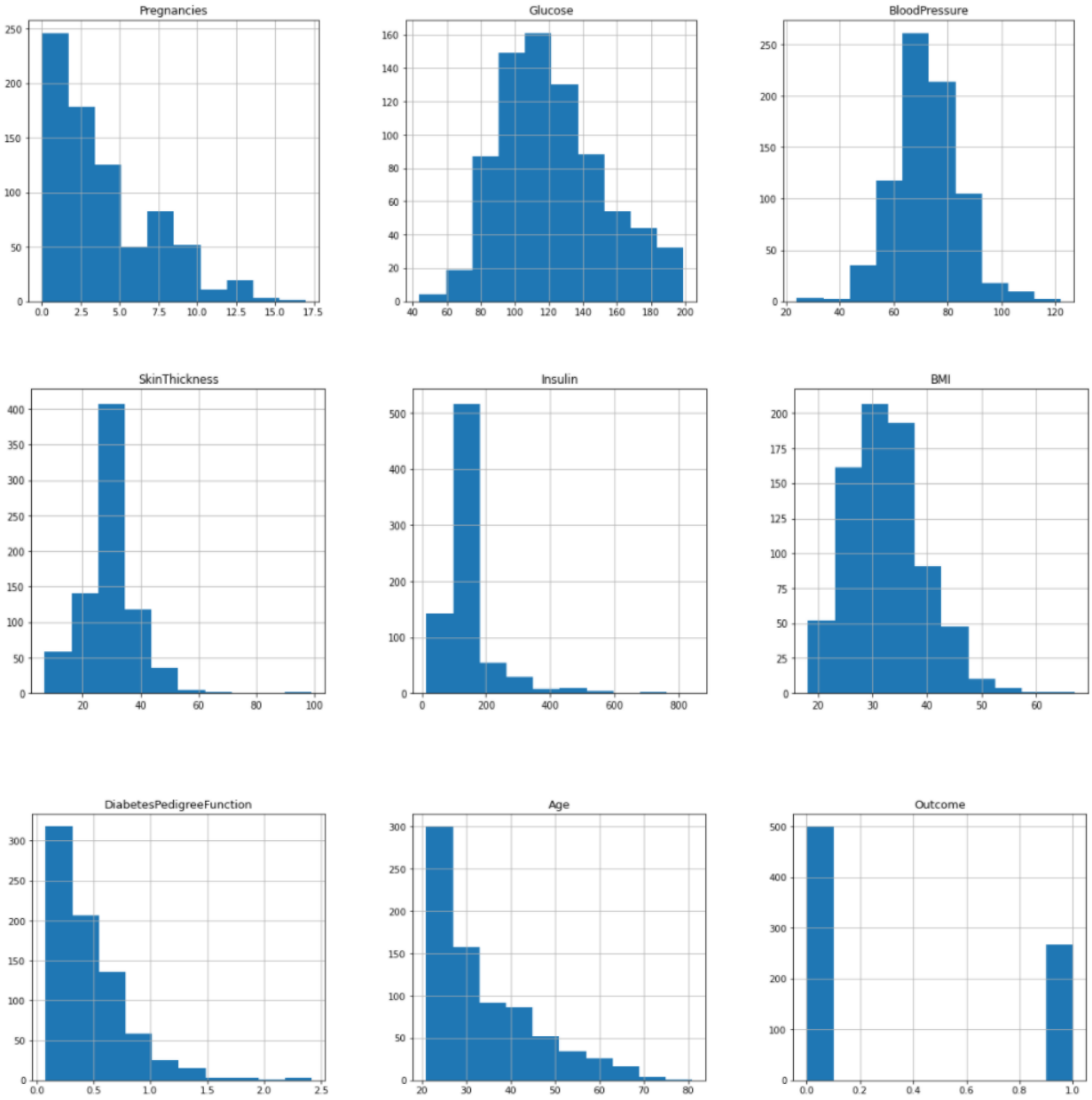
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

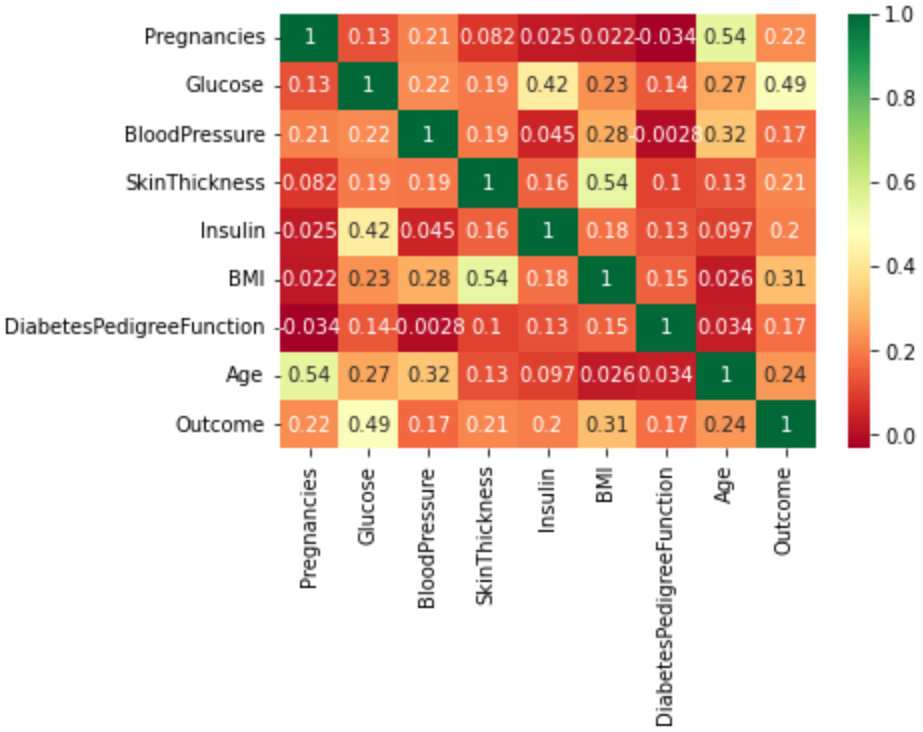
Information about our dataset

```
# gives information about the data types,columns, null value counts, memory usage etc
diabetes_df.info()
```

```
<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
```

6. Benchmarking





From above image, we are able to find the relationship among the columns. ML libraries seaborn is used to plot heatmap which help us to find correlation among columns.

7. Applicable Regulations (Government and Environmental)

- Patents on ML algorithms developed
- Laws related to privacy for collecting data from users
- Protection/ownership regulations
- Creating an e-mail service to mail the report to the patient and doctor.
- Being responsible by design.
- Ensuring open-source, academic and research community for an audit of Algorithms.
- Review of existing work authority regulations.

8. Applicable Constraints

Expertise:

- Requires a lot of research to obtain universal dataset of diabetes patients in-order to provide more sophisticated and accurate results.
- Establishing e-mail service in the product which have to send the report after the machine learning model is deployed in any server.
- Confidential health data to be obtained to train the model.

- Thorough understanding of dataset and verification of the results must be performed by the medical staff from the machine learning model to provide a great health prescription and service to the user.

9. Business Opportunity:

It's very common that in initial stage diabetes go undetected unless you experience some noticeable signs of disease. To overcome situation, we are going to use Machine Learning, which will provide faster and more accurate result to help peoples to detect at early stage.

10. Concept Generation:

Diabetes is a group of diseases in which the body doesn't produce enough or any insulin, doesn't properly use the insulin that is produced, or exhibits a combination of both. When any of these things happens, the body is unable to get sugar from the blood into the cells. That leads to high blood sugar levels. Glucose, the form of sugar found in your blood, is one of your main energy sources. A lack of insulin or resistance to insulin causes sugar to build up in your blood. This can lead to many health problems.

The three main types of diabetes are:

Type 1 diabetes: Type 1 diabetes is believed to be an autoimmune condition. This means your immune system mistakenly attacks and destroys the beta cells in your pancreas that produce insulin. The damage is permanent.

What prompts the attacks isn't clear. There may be both genetic and environmental reasons. Lifestyle factors aren't thought to play a role.

Type 2 diabetes: Type 2 diabetes starts as insulin resistance. This means your body can't use insulin efficiently. That stimulates your pancreas to produce more insulin until it can no longer keep up with demand. Insulin production decreases, which leads to high blood sugar.

The exact cause of type 2 diabetes is unknown. Contributing factors may include:

- genetics
- lack of exercise
- being overweight

There may also be other health factors and environmental reasons.

Gestational diabetes: Gestational diabetes is due to insulin-blocking hormones produced during pregnancy. This type of diabetes only occurs during pregnancy.

So in order to generate the model based on the problem stated above, we need to use Machine learning. Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so.

1. First we clean the data and scale the data.
2. Splite the data in x,y variable

```
diabetes_df.columns
```

```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

```
sc_X= StandardScaler()
X = pd.DataFrame(sc_X.fit_transform(diabetes1_df.drop(['Outcome'],axis = 1)),columns=['Pregnancies', 'Glucose', 'BloodPressure',
       'BMI', 'DiabetesPedigreeFunction', 'Age'])
X.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	0.639947	0.865108	-0.033518	0.670643	-0.181541	0.166619	0.468492	1.425995
1	-0.844885	-1.206162	-0.529859	-0.012301	-0.181541	-0.852200	-0.365061	-0.190672
2	1.233880	2.015813	-0.695306	-0.012301	-0.181541	-1.332500	0.604397	-0.105584
3	-0.844885	-1.074652	-0.529859	-0.695245	-0.540642	-0.633881	-0.920763	-1.041549
4	-1.141852	0.503458	-2.680669	0.670643	0.316566	1.549303	5.484909	-0.020496

```
y = diabetes1_df.Outcome
```

```
# train_test_split
X_train, X_test,y_train,y_test = train_test_split(X,y,test_size=1/3,random_state=42,stratify=y)
```

3. We will use different models and we will finalize the model which will give good accuracy

1. Bagging - Decision Tree

Bagging - Decision Tree

```
bagging = BaggingClassifier(DecisionTreeClassifier(random_state=42),n_estimators=500,
                            max_samples=100,bootstrap=True,n_jobs=-1,random_state=42,
                            oob_score=True)
bagging.fit(X_train,y_train)
```

```
BaggingClassifier(base_estimator=DecisionTreeClassifier(random_state=42),
                  max_samples=100, n_estimators=500, n_jobs=-1, oob_score=True,
                  random_state=42)
```

```
print(classification_report(bagging.predict(X_test),y_test))
```

```
              precision    recall  f1-score   support

           0       0.96      0.91      0.94       104
           1       0.84      0.92      0.88        50

    accuracy                           0.92       154
   macro avg       0.90      0.92      0.91       154
weighted avg       0.92      0.92      0.92       154
```

2. RandomForest Classifier with GridSearchCV

```
rf = RandomForestClassifier(n_jobs=-1,max_features= 'sqrt' ,n_estimators=100, oob_score = True)
param_grid = {
    'n_estimators': [200, 700],
    'max_features': ['auto', 'sqrt', 'log2']
}

rfc = GridSearchCV(estimator=rf, param_grid=param_grid, cv= 5)
rfc.fit(X_train,y_train)
print(classification_report(y_test,rfc.predict(X_test)))
```

```
              precision    recall  f1-score   support

           0       0.91      0.91      0.91        99
           1       0.84      0.84      0.84        55

    accuracy                           0.88       154
   macro avg       0.87      0.87      0.87       154
weighted avg       0.88      0.88      0.88       154
```

3. XGBoost Classifier

XGBoost Classifier

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.2, random_state=42)
xgbc = xgb.XGBClassifier(
 learning_rate = 0.01,
 n_estimators= 2000,
 max_depth= 9,
 min_child_weight= 2,
 #gamma=1,
 gamma=0.4,
 subsample=0.8,
 colsample_bytree=0.8,
 objective= 'binary:logistic',
 nthread= -1,
 scale_pos_weight=1).fit(X_train, y_train)
```

```
print(classification_report(y_test,xgbc.predict(X_test)))
```

```
              precision    recall  f1-score   support

           0       0.92      0.93      0.92        99
           1       0.87      0.85      0.86        55

    accuracy                           0.90       154
   macro avg       0.90      0.89      0.89       154
weighted avg       0.90      0.90      0.90       154
```

4. LightGBM Classifier

LightGBM Classifier

```
random_state=42

fit_params = {"early_stopping_rounds" : 100,
              "eval_metric" : 'auc',
              "eval_set" : [(X,y)],
              'eval_names': ['valid'],
              'verbose': 0,
              'categorical_feature': 'auto'}

param_test = {'learning_rate' : [0.01, 0.02, 0.03, 0.04, 0.05, 0.08, 0.1, 0.2, 0.3, 0.4],
              'n_estimators' : [100, 200, 300, 400, 500, 600, 800, 1000, 1500, 2000],
              'num_leaves': sp_randint(6, 50),
              'min_child_samples': sp_randint(100, 500),
              'min_child_weight': [1e-5, 1e-3, 1e-2, 1e-1, 1, 1e1, 1e2, 1e3, 1e4],
              'subsample': sp_uniform(loc=0.2, scale=0.8),
              'max_depth': [3,4,5,6,7],
              'colsample_bytree': sp_uniform(loc=0.4, scale=0.6),
              'reg_alpha': [0, 1e-1, 1, 2, 5, 7, 10, 50, 100],
              'reg_lambda': [0, 1e-1, 1, 5, 10, 20, 50, 100]}
```

```
#number of combinations
n_iter = 300

#intialize lgbm and lunch the search
lgbm_clf = lgbm.LGBMClassifier(random_state=random_state, silent=True, metric='None', n_jobs=4)
grid_search = RandomizedSearchCV(
    estimator=lgbm_clf, param_distributions=param_test,
    n_iter=n_iter,
    scoring='accuracy',
    cv=5,
    refit=True,
    random_state=random_state,
    verbose=True)

grid_search.fit(X, y, **fit_params)
opt_parameters =  grid_search.best_params_
lgbm_clf = lgbm.LGBMClassifier(**opt_parameters)
```

```
Fitting 5 folds for each of 300 candidates, totalling 1500 fits
```

```
model = lgbm_clf
model.fit(X_train,y_train)
print(classification_report(y_test,model.predict(X_test)))
```

```
              precision    recall  f1-score   support

           0       0.94      0.92      0.93        99
           1       0.86      0.89      0.88        55

    accuracy                           0.91       154
   macro avg       0.90      0.91      0.90       154
weighted avg       0.91      0.91      0.91       154
```

By analyzing all the models, we can say that Bagging - Decision Tree is giving good results.

Bagging - Decision Tree

- Accuracy is 92%

RandomForest Classifier with GridSearchCV

- Accuracy is 88%

XGBoost Classifier

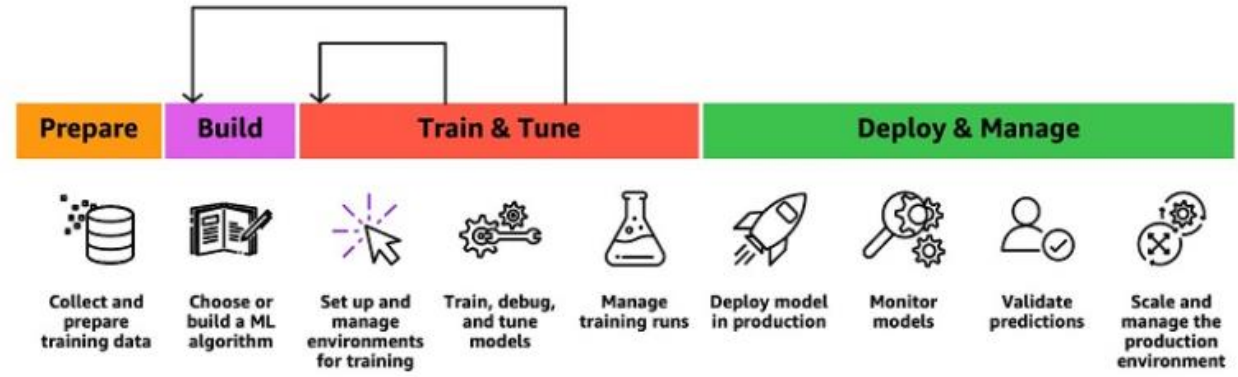
- Accuracy is 90%

LightGBM Classifier

- Accuracy is 91%

11.Concept Development

The concept can be developed by using the appropriate API (flask in this case) and using Django as framework for the same and for its deployment, the cloud services has to be chosen accordingly to the need.



12. Final Report Prototype

The product takes the following functions to perfect and provide a good result.

Back-end

Model Development: This must be done before releasing the service. A lot of manual supervised machine learning must be performed to optimize the automated tasks.

1. Performing EDA to realize the dependent and independent features.
2. Algorithm training and optimization must be done to minimize overfitting of the model and hyperparameter tuning.

Front End

1. Different user interface: The user must be given many options to choose form in terms of parameters. This can only be optimized after a lot of testing and analysis all the edge cases.
2. Interactive visualization the data extracted from the trained models will return raw and inscrutable data. This must be present in an aesthetic and an “easy to read” style.
3. Feedback system: A valuable feedback system must be developed to understand the patients and doctors needs that have not been met. This will help us train the models constantly

13. Product details - How does it work?

An interactive user system will take inputs regarding the patient health from the user and the user will get to know about the diabetes condition and guidance needed from the doctors. Doctors will be able to monitor their patient more efficiently and able to communicate with them easily.

14. Code Implementation

This is a github link :- https://github.com/ashishk831/Project3_Diabetes

15. References/Source of Information

https://www.healthline.com/health/diabetes/types-of-diabetes

https://www.openaccessjournals.com/articles/problems-associated-with-diabetes-care-in-india.pdf

https://www.kaggle.com/mathchi/diabetes-data-set

https://www.nature.com/articles/s41598-020-71908-9