DIABETES INVESTIGATION (CAPSTONE PROJECT PROPOSAL)

BACKGROUND

Diabetes mellitus is a chronic disease where the concentration of glucose in the blood of the patients is at an unusually high level (WHO, 2014). This disease could be caused by the failure of some cells in the body to respond to insulin, or the inability of the pancreas to produce insulin; a protein that stimulates blood glucose reduction (WHO, 2013). In the early stages of the condition, symptoms include frequent urination whilst at the end stage, is characterized by blindness, idiopathic foot ulcers, stroke and eventually death.

Approximately **8%** of the world population (adults only) suffer from *diabetes* IDF (2019). According to the CDC (2020), *diabetes* ranks 7th among the diseases that cause the most deaths in the worldwide. Though several efforts have been directed towards the reduction of the incidence of *diabetes*, cases continue to rise (IDF, 2019).

The risk of complications from *diabetes* can be prevented when measures aimed at controlling the blood glucose levels are applied at an early stage of the condition thus, making early detection and treatment of the condition very important. Also, *diabetes* tests in regions around the world that do not have ready access to healthcare takes days and sometimes months.

Therefore, it is very important to have a machine learning model, that can predict the diabetic status faster than the traditional methods. Due to my background in cellular biology and biochemistry, I see this field as very important because it exposes me to real world situations in my field where machine learning can be applied.

PROBLEM STATEMENT

There is the need to obtain ways of testing the diabetic status of a person quickly. This would ensure that interventions to reduce the risk of complications with regards to diabetes, are reduced. This problem will be a **binary classification task** and characteristics of patients such as blood glucose levels, skin thickness and age would be considered, and would be used to predict whether a patient is diabetic or not.

DATASET

The dataset to be used is the PIMA Indian Diabetes dataset from Kaggle. This dataset was originally collated by the National Institute of Diabetes and Digestive and Kidney Diseases and the subjects are 21+ years old females. Some of the information obtained from the females were their age, Body Mass Index (BMI) and Insulin levels. Also, the diabetes status of the females were determined. This is a freely downloadable dataset from Kaggle, and would be used to train and evaluate a machine learning model for the task at hand. It contains **768 patients (rows)** and **9 measured characteristics (features)** including their blood glucose levels, skin thickness and age. The classes are imbalanced as shown below with non-diabetics being almost twice as much as diabetics. Most scikit-learn algorithms have the **class_weight** parameter which I will set to ‘balanced’ to deal with the class imbalance. I will split the dataset into 20% test set and 80% training set using the **train_test_split** function. I will also set the **stratify** argument to True to perform stratify sampling so that the class imbalance will not drastically affect the splitting. I will then perform cross-validation to get a better estimate of my model performance.

In [1]:

```
# import pandas library
import pandas as pd


# load the dataset and view the first five rows
data = pd.read_csv(
    filepath_or_buffer='/home/biopython/Downloads/Datasets/datasets_228_482_diabetes.csv'
)


# print the number of columns and rows in the data
print('There are {} patients and {} measured characteristics of the patients'.format(
    data.shape[0], data.shape[1]))
print()
# prints the nimber of diabetics and non-diabetics
print('The labels are discrete values; 0 for non-diabetic, 1 for diabetic\n', data.Outcome.value_counts())
```

```
There are 768 patients and 9 measured characteristics of the patient
s

The labels are discrete values; 0 for non-diabetic, 1 for diabetic
 0    500
1    268
Name: Outcome, dtype: int64
```

SOLUTION STATEMENT

This problem can be solved by building highly accurate and precise machine learning models such as KNearestNeighbors, NaiveBayes and LogisitcRegression e.t.c. These models would be trained to predict whether a patient is diabetic ‘1’ or is non-diabetic ‘0’. A model for this task would be considered as successful if it has an accuracy >= 90% and also, a specifity and sensitivity values between 0.7 and 0.8.

EXISTING METHODS (BENCHMARK MODEL)

Existing methods of detecting diabetes include measuring blood glucose levels of a patient and determining whether the levels of glucose is beyond a particular threshold. This process could take days especially in regions where accessibility to healthcare is inadequate.

The bench mark model to use would be a **DummyClassifier** model with a **cross-validated accuracy** of about **0.53 %** which can be obtained from sklearn and used as shown below. The performance of future models would be compared to the cross-validated accuracy of this dummy model and will tell whether my model is performing better or worse.

In [2]:

```
# import the relevant sklearn functions to be used
from sklearn.model_selection import cross_val_score
from sklearn.dummy import DummyClassifier


# set the feature matrix as 'X' and the response vector as 'y'
X = data.drop('Outcome', axis=1)
y = data.Outcome


# instantiate a dummy classifier
clf = DummyClassifier(strategy='stratified')


# calculate 10-fold cross-validated accuracy of the model
cv_accuracy = cross_val_score(estimator=clf,
                              X=X,
                              y=y,
                              scoring='accuracy',
                              cv=10,
                             ).mean()
print('The accuracy of the base model is {}'.format(cv_accuracy))
```

```
The accuracy of the base model is 0.5455058099794943
```

EVALUATION METRICS

The performance of the target model would be measured by using model 10-fold cross-validated accuracy. Also, the specificity and sensitivity of the target model will be examined to actually see a better estimate of model performance.

WORKFLOW

- Relevant libraries such as *pandas*, *numpy*, *matplotlib* and *sci-kit* learn will be imported.
- The dataset will be downloaded from *Kaggle*
- The dataset will be preprocesed by dealing with null values and removing duplicate observations
- Data exploration will be done by checking the correlation of each feature with the target variable, and also with other feature variables.
- Feature selection and Engineering will be done based on the results of the data exploration.
- The final processed and engineered dataset will then be fed to a base model **DummyClassifier**.
- Cross-validated accuracy of the model will then be determined.
- Other models such as **KNearestNeighbors** will also be trained, evaluated, and compared to the performance of the base model.
- A final *confusion matrix* and an *roc/auc curve*, as well as *specificity* and *sensitivity* of the model, will be determined and used as a final estimate of model performance.
- The final model will be deployed as a web-app which will take an array of values as input then, will output whether the person is diabetic or not.

REFERENCES

- "IDF DIABETES ATLAS Ninth Edition 2019" (PDF). www.diabetesatlas.org. Retrieved 18 May 2020.
- "Diabetes Fact sheet N°312". WHO. October 2013. Archived from the original on 26 August 2013. Retrieved 25 March 2014.
- "What is Diabetes?". Centers for Disease Control and Prevention. 11 March 2020. Retrieved 18 May 2020.
- "The top 10 causes of death Fact sheet N°310". World Health Organization. October 2013. Archived from the original on 30 May 2017.
- "About diabetes". World Health Organization. Archived from the original on 31 March 2014. Retrieved 4 April 2014.

ACKNOWLEDGEMENTS

Smith, J.W., Everhart, J.E., Dickson, W.C., Knowler, W.C., & Johannes, R.S. (1988). Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In Proceedings of the Symposium on Computer Applications and Medical Care (pp. 261--265). IEEE Computer Society Press.