Udacity Nanodegree Capstone Project Report

2019

Predict the onset of diabetes based on diagnostic measures

Debu Sinha
debusinha2009@gmail.com

Table of Content

Definition

Domain Background

Joshi and Parikh have described India as the “capital of the world with 41 million Indians having diabetes, every fifth diabetic in the world is an Indian”.

In 2000, India (31.7 million) topped the world with the highest number of people with diabetes mellitus followed by China (20.8 million) with the United States (17.7 million) in second and third places, respectively. According to Wild et al. .[3], the prevalence of diabetes is predicted to double globally from 171 million in 2000 to 366 million in 2030, with a maximum increase in India. It is predicted that by 2030 diabetes mellitus may afflict up to 79.4 million individuals in India, while China (42.3 million) and the United States (30.3 million) will also see significant increases in those affected by the disease.[3,4] India currently faces an uncertain future about the potential burden that diabetes may impose upon the country. Many influences affect the prevalence of disease throughout the country, and the identification of those factors is necessary to facilitate change when facing health challenges.

Although the Indian urban population has access to reliable screening methods and anti-diabetic-medications, such health benefits are not often available to rural patients. There is a disproportionate allocation of health resources between urban and rural areas, and also, poverty in rural areas may be multifaceted. Aged care facilities in rural areas report disparity in diabetes management compared with their urban counterparts,[11] with these populations more likely to suffer from diabetic complications compared to their urban counterparts. More needs to be done to address the rural-urban inequality in diabetes intervention entry.

The clinical data that will be used in this project has been provided by Kaggle https://www.kaggle.com/uciml/pima-indians-diabetes-database.

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

The datasets consist of several medical predictor variables and one target variable, Outcome. Predictor variables include the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.

The dataset has 768 entries. The outcome class has two categories 0 and 1. The data is unbalanced as the number of 1’s(tested positive) is 268 and the 0's(tested negative) are 500.

Since the ratio of the **number of outcomes with value 0 / number of outcome with value 1** is 1.82 we can move forward with our model training without performing upsampling or downsampling.

The models we are proposing to work with are logistic regression and XGBoost both of which handle unbalanced data well.

Upsampling can suffer from overfitting and downsampling can suffer from loss of important information in the data points that we don’t include.

Problem Statement

The goal of this project is to build a reliable machine learning model to diagnose if a person has diabetes or not based on clinical data input provided with high precision.

This is a classification problem to determine if a patient has diabetes or not.

Evaluation Metrics

As the use case we are working on is dealing with a classification problem, we will evaluate the performance of the model based on **Accuracy** that we can get from the sklearn.metrics Apart from the accuracy score it will also be important to have **high recall and high precision** in order to correctly diagnose the diabetes condition. We can use the **F1 score** as a metric to evaluate our model. F1 score will raise a flag if either of the precision and recall value is low.

Analysis

Data Exploration

The dataset used for this project is the Pima-Indians-diabetes.csv, which was picked from the website kaggle.com and original owners are the National Institute of Diabetes and Digestive and Kidney Diseases.

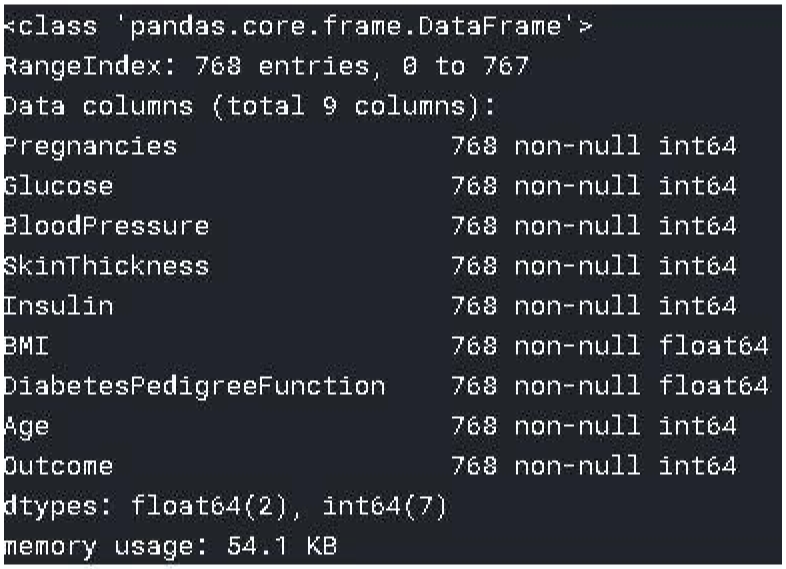
There are 768 records in the dataset and 9 features.

Feature label	**Description**
Pregnancies	Number of times pregnant
GlucosePlasma	Plasma glucose concentration 2 hours in an oral glucose tolerance test
BloodPressureDiastolic	Diastolic blood pressure (mm Hg)
SkinThicknessTriceps	Triceps skinfold thickness (mm)
Insulin	2-Hour serum insulin (mu U/ml)
BMI body	Body mass index (weight in kg/(height in m)^2)
DiabetesPedigreeFunction	Diabetes pedigree function
Age	Age (years)
Outcome	Class variable (0 or 1) 268 of 768 are 1, the others are 0

We can take a look at the data to determine if we have any null values and the data types associated with each column.

It is important to deal with null values and categorical columns as machine learning algorithms don't do well with the sparse datasets and understand only numeric values.

In case we have found any categorical values we will need to convert them to numeric and also perform one-hot encoding in order to remove bias.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
Pregnancies                 768 non-null int64
Glucose                     768 non-null int64
BloodPressure               768 non-null int64
SkinThickness               768 non-null int64
Insulin                     768 non-null int64
BMI                         768 non-null float64
DiabetesPedigreeFunction    768 non-null float64
Age                         768 non-null int64
Outcome                     768 non-null int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

The dataset we have here does not have any categorical values or null values. All the datatypes are numeric.

There are 500 non-diabetic patients (class = 0) and 268 diabetic ones (class = 1) for an incidence rate of 34.9%. Thus if you simply guess that all are non-diabetic, your accuracy rate is 65.1% (or error rate of 34.9%).

Although the database is labeled as having no missing values, some of the features have 0 value which doesn’t look correct.

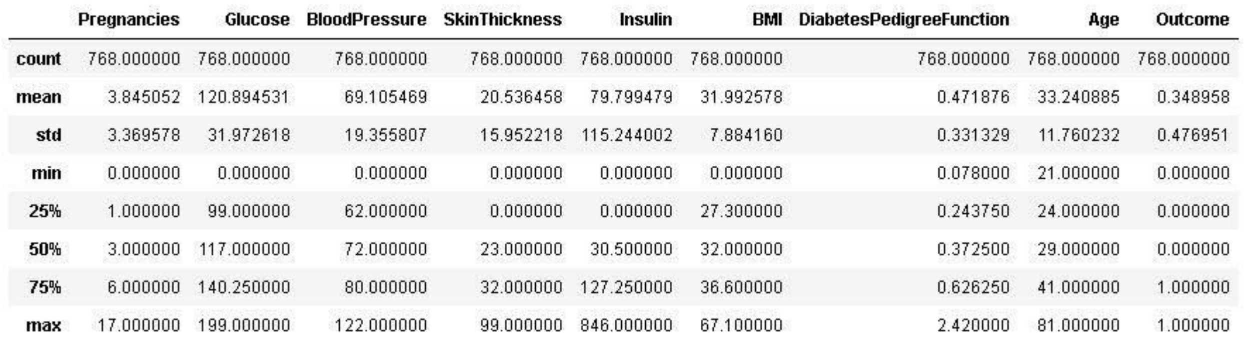
number of rows missing Glucose	5
number of rows missing BloodPressure	35
number of rows missing SkinThickness	227
number of rows missing Insulin	374

number of rows missing BMI	11
number of rows missing DiabetesPedigreeFunction	0
number of rows missing Age	0

Studies that did not realize the previous zeros were in fact missing variables essentially used a rule of substituting zero for the missing variables.

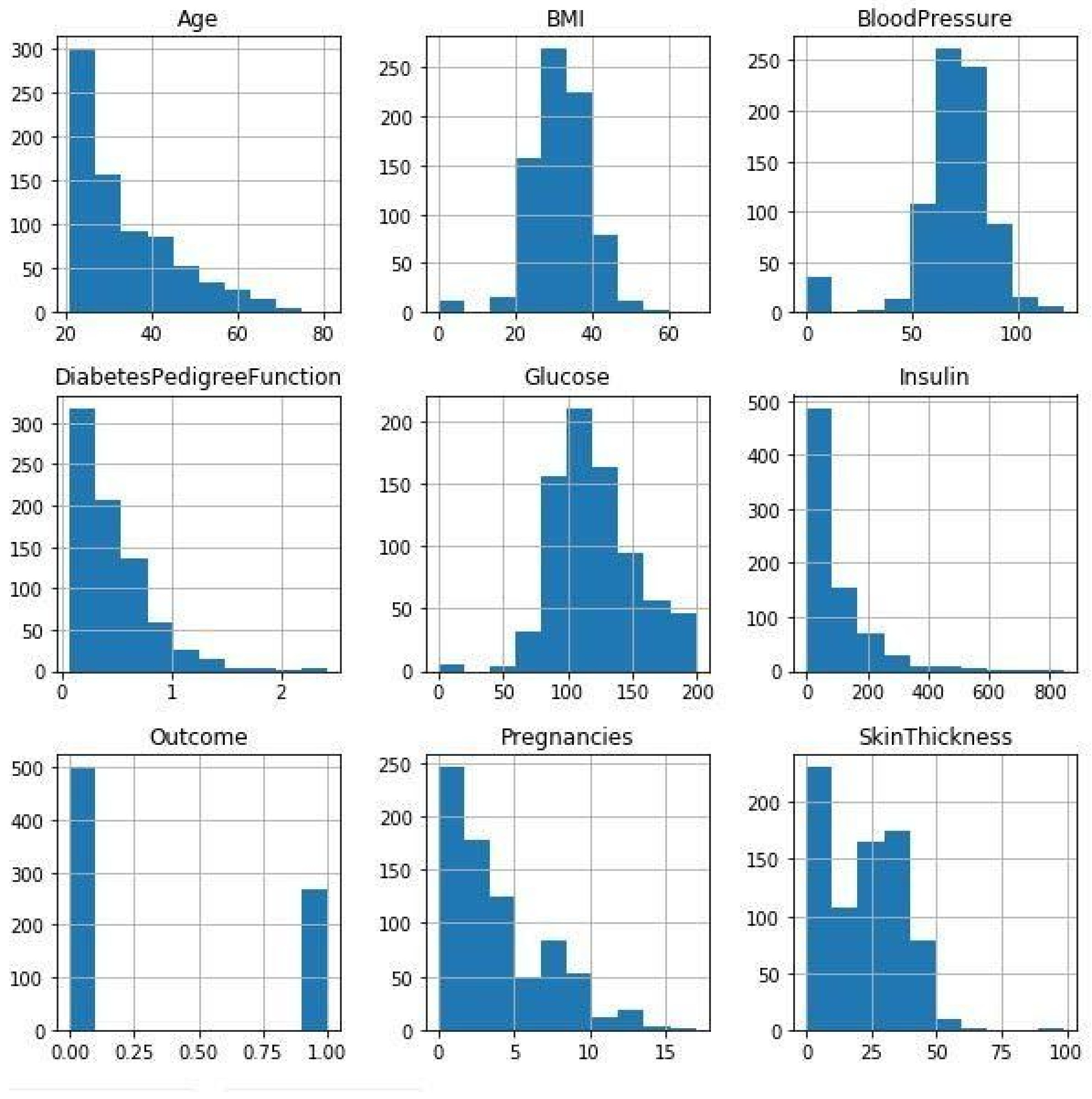
All of these features bear importance in predicting the class variable. From domain knowledge **GlucosePlasma** and **Insulin** seem to have strong relevance. **BloodPressureDiastolic** is important as diabetic patients usually have diastolic blood pressure 1-3 mm lower than normal. Older individuals seem to have a higher risk of diabetes, it is difficult to contract diabetes at a young age because of the high metabolism, and everything else is normal. Diabetes can be hereditary, hence the **DiabetesPedigreeFunction** is an important feature because it is a measure of the occurrence of diabetes in relatives and blood-related individuals to the patient.

Important: It should be noted that the above data set contains only limited features, whereas in reality, numerous features come into play.



	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

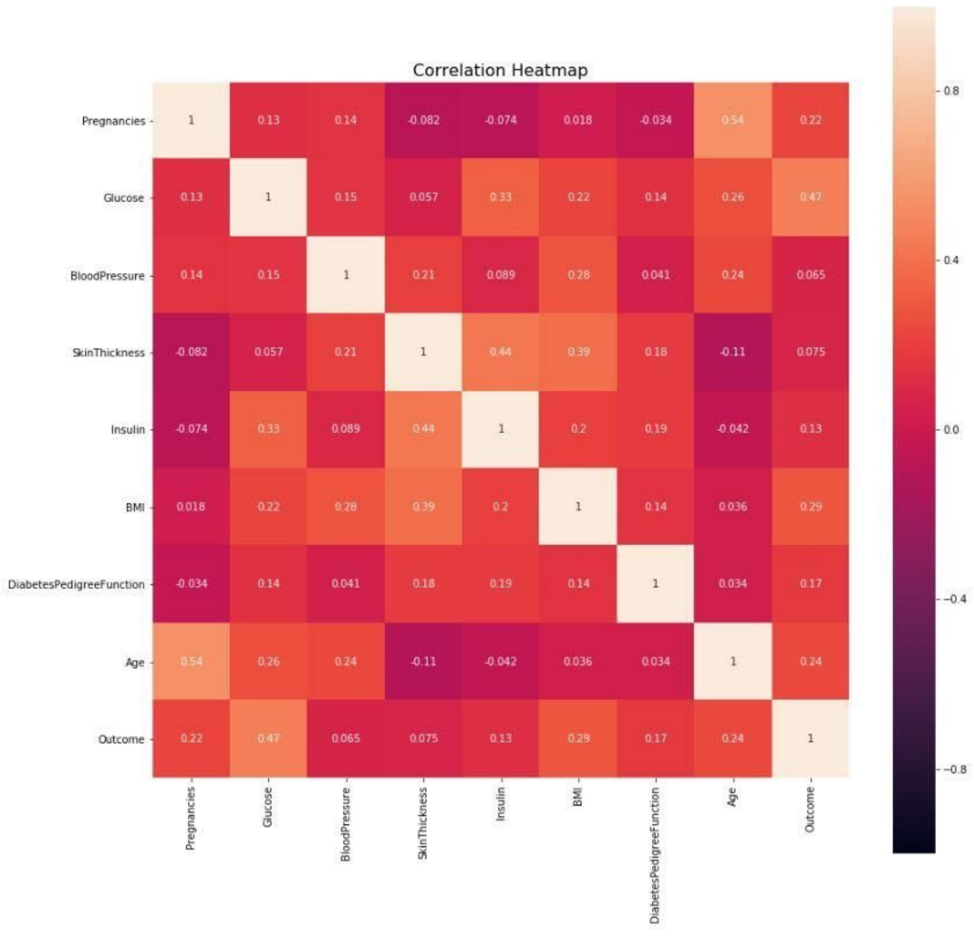
Exploratory Visualization



From the histograms, we can tell that Age, DiabetesPedigreeFunction, Insulin, Pregnancies, and SkinThickness are positively skewed. Blood pressure is negatively skewed. We can also see that we have about 500 records with the outcome of 0(diabetes negative) and 278 records with outcome of 1(diabetes positive). We have enough data points for each category of the target variable to train a classification model.

Most of the women are pregnant in the range of 0-2 and most of the women studied were in the age group of 20-25.

We can draw a correlation matrix to visualize the correlation between various variables in our dataset. The correlation value ranges between -1(perfect negative) and +1(perfect positive) and tells how strongly one variable is correlated to the other. A value of 0 means the two variables have no relation with each other.



There is some correlation between the number of pregnancies and age. BMI and tricep skin thickness also seem to be a bit correlated. Glucose has the highest correlation with the outcome. It is also correlated to Insulin.

Algorithms and Techniques

The algorithms used In this project are Implemented after the data is cross-validated after splitting it into training and testing sets, and after setting a random state. The algorithms used are imported from the scikit learn library of python. Three supervised classification algorithms were trained and tested on the dataset, and the ones which showed the best accuracies were shortlisted and stacked to form a combination classifier. The algorithms tested were SVM classifier, Decision Tree Classifier classifier and Logistic Regression.

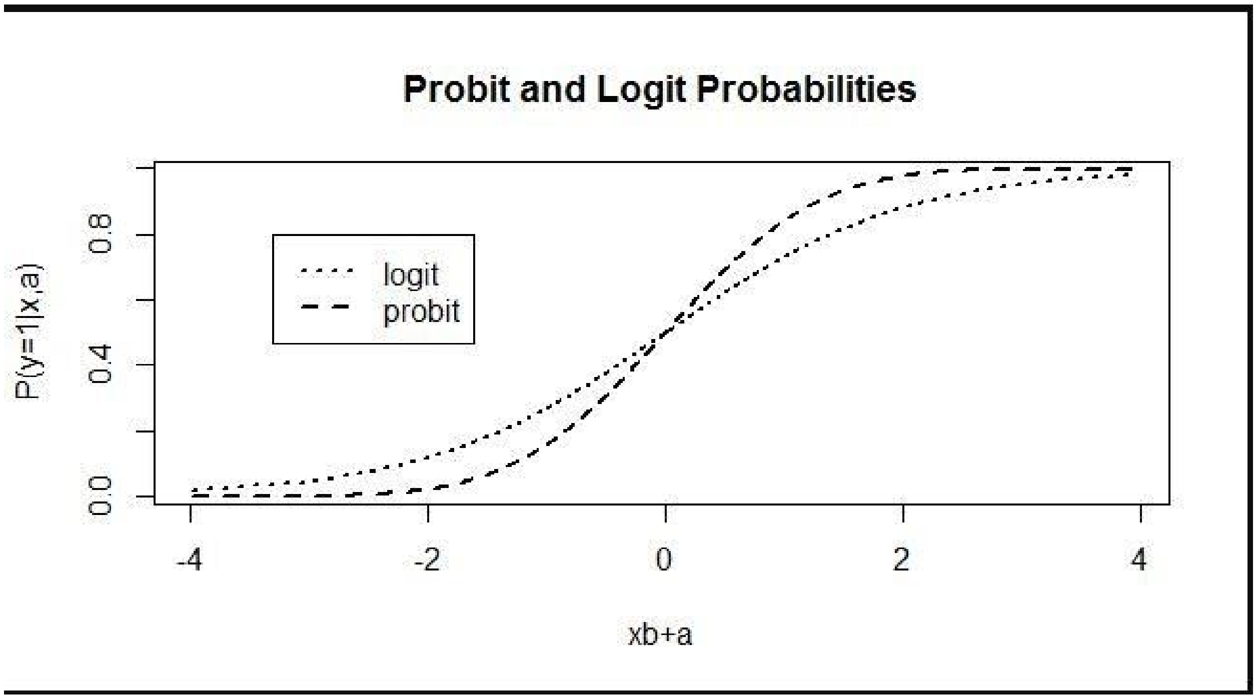
The shortlisted algorithms after setting a random state to each of them and training them separately were Linear SVM, Radial SVM and Logistic Regression classifiers. The individual classifies were stacked together using a soft VotingClassifier which combines similar or conceptually different machine learning classifiers for classification via majority or plurality voting. In soft voting, we predict the class labels based on the predicted probabilities. XGBoost algorithm was also trained to make predictions on this dataset to get the best performing model on the dataset.

A support vector machine is a binary classification algorithm. This algorithm is robust to a very large number of variables and small sample. It can also learn both simple and complex classification models and using complex mathematical principles to avoid overfitting. When using a linear kernel SVM finds a linear decision surface that separates the outcome class with the largest gap or margin. If such a linear boundary does not exist, the data is mapped into a much higher multidimensional space where the separating decision surface is found.

Decision Tree Classifier is a simple and widely used classification technique. It applies a straightforward idea to solve the classification problem. Decision Tree Classifier poses a series of carefully crafted questions about the attributes of the test record. Each time it receives an answer, a follow-up question is asked until a conclusion about the class label of the record is reached. Once the decision tree has been constructed, classifying a test record is straightforward. Starting from the root node, we apply the test condition to the record and follow the appropriate branch based on the outcome of the test. It then leads us either to another internal node, for which a new test condition is applied or to a leaf node. When we reach a leaf node, the class label associated with the leaf node is then assigned to the record, it traces the path in the decision tree to predict the class label of the test record, and the path terminates at a leaf node labeled either with 0 or 1.

Logistic regression, also known as maximum entropy, is really good when dealing with binary labels. Logistic regression assumes that a dependent variable is a stochastic event. For example, if we analyze pesticides kill rate, the outcome event is either killed or alive. Since even the most resistant bug can only be either of these two states, logistic regression thinks in likelihoods of the bug getting killed. If the likelihood of killing the bug is > 0.5 it is assumed dead, if it is < 0.5 its assumed alive. The outcome variable – which must be coded as 0 and 1 – is placed in the first box labeled Dependent, while all predictors are entered into the Covariates box (categorical variables should be appropriately dummy coded).

Sometimes instead of a logit model for logistic regression a probit model is used. The following graph shows the difference between logit and a probit model for different values (-4,4).



Both models are commonly used in logistic regression, and in most cases, the model is fitted with both functions and the function with the better fit is chosen. However, probit assumes normal distribution of the probability of the event, when logit assumes the log distribution. Thus the difference between logit and probit is typically seen in small samples.

At the center of the logistic regression analysis is the task estimating the log odds of an event. Mathematically, logistic regression estimates a multiple linear regression function defined as:

$$\log\left(\mathrm{Odds}\left(Y=1\right)\right)=\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_q x_q.$$

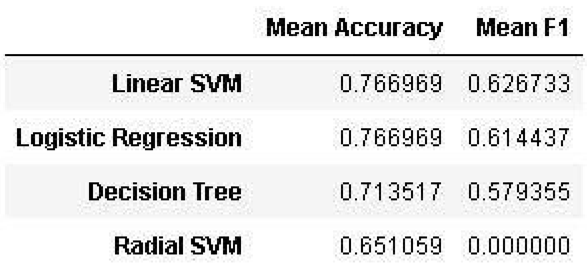
$$\mathrm{Odds}\left(Y=1\right)=\frac{p}{1-p}.$$

XGBoost is an implementation of gradient boosted trees designed for speed performance. Boosting is an ensemble technique where new models are added to correct the errors made by existing models. Models are added sequentially until no further improvements can be made. A popular example is an AdaBoost algorithm that weights data points that are hard to predict.

Gradient boosting is an approach where new models are created that predict the residuals or errors of prior models and then added together to make the final prediction. It is called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models.

Benchmark

For the benchmarking model, we have used multiple algorithms over data that have not been preprocessed. Logistic regression is still our primary benchmark.



	Mean Accuracy	Mean F1
Linear SVM	0.766969	0.626733
Logistic Regression	0.766969	0.614437
Decision Tree	0.713517	0.579355
Radial SVM	0.651059	0.000000

Methodology

Data Preprocessing

The data consisted of zeroes at inappropriate places, it is impossible for some of these features to have a value of zero. While entering the data, the missing values were put as 0 in five of these features: Glucose, blood pressure, Insulin, BMI, SkinThickness. The pandas data.describe() function was used to confirm the suspicion and carry out statistical analysis of the features. Some of these features had their first quartiles and min values as zero.

The Glucose concentration features missing values were replaced by its mean. For Blood pressure, the missing values were replaced by the median and mean were the same for this feature. Triceps skin thickness missing values were replaced by its median as the number of missing values is too high, the appropriate metric for this feature was the median, a similar replacement was done in the case of serum insulin. The body mass index was replaced by its mean. After this, the data was separated into two separate variables, features, and labels. The features were scaled together using StandardScalar function from the preprocessing module of scikit learn. This data was stored in X. X and y were later fed into the cross-validation module to be split into training and testing states, the split was decided to be 70-30.

Implementation

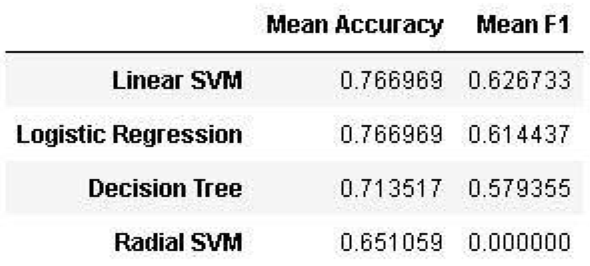
The algorithms were imported, and their random states and hyper parameters set, time was imported to evaluate the training and testing time of every algorithm. The evaluation metrics like accuracy, F1 score were imported as well to measure the performance of every algorithm. At first linear SVM, radial SVM, Logistic regression and Decision tree algorithm were trained and tested on the raw dataset and the result was recorded. Logistic regression and Linear SVM performed the best while radial SVM giving the worst F1 score of 0.

While performing exploratory data analysis we realized that the dataset was not clean and certain features like SkinThickness and Glucose had 0 value which didn’t look correct. After performing data cleaning and processing Linear SVM, Radial SVM, Logistic regression and Decision tree classifiers were trained again on the dataset and this time the best performing classifiers were Logistic regression, Linear and Radial SVM.

These were then stacked together in a voting classifier to perform the classification on the PIMA dataset along with XGBoost to record any performance boost.

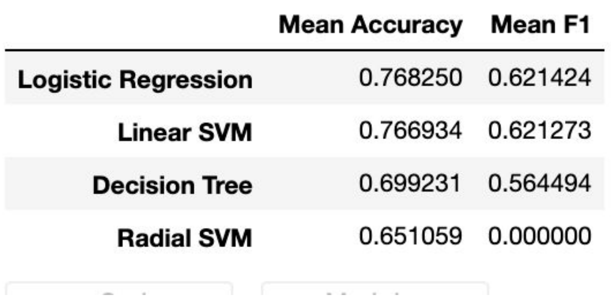
Refinement

While working with raw dataset the untuned algorithms had the following performance:



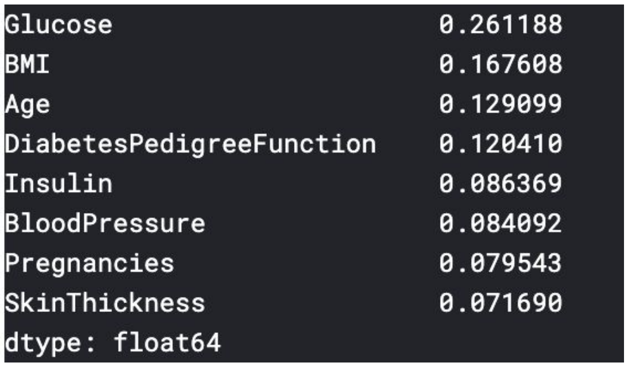
	Mean Accuracy	Mean F1
Linear SVM	0.766969	0.626733
Logistic Regression	0.766969	0.614437
Decision Tree	0.713517	0.579355
Radial SVM	0.651059	0.000000

Radial kernel SVM performed the worst. After preprocessing our data (using imputer to fill in values for 0 where-ever needed and also scaling the features). There was no significant improvement in the model's performance.



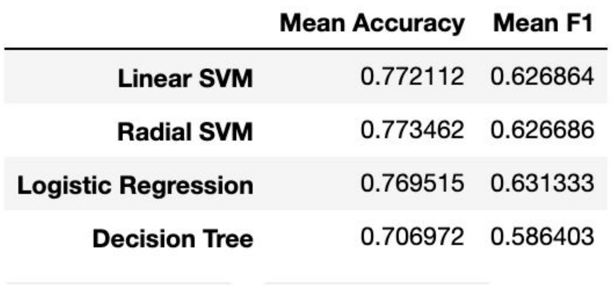
	Mean Accuracy	Mean F1
Logistic Regression	0.768250	0.621424
Linear SVM	0.766934	0.621273
Decision Tree	0.699231	0.564494
Radial SVM	0.651059	0.000000

Next we used the RandomForestClassifier to get the most important features:



```
Glucose                     0.261188
BMI                         0.167608
Age                         0.129099
DiabetesPedigreeFunction    0.120410
Insulin                     0.086369
BloodPressure               0.084092
Pregnancies                 0.079543
SkinThickness               0.071690
dtype: float64
```

After identifying the top features (Glucose, BMI, Age, and DiabetesPedigreeFunction) we trained the classification algorithms again on our dataset this time there was a significant improvement in the performance of Radial SVM and its F1 score.



	Mean Accuracy	Mean F1
Linear SVM	0.772112	0.626864
Radial SVM	0.773462	0.626686
Logistic Regression	0.769515	0.631333
Decision Tree	0.706972	0.586403

For final experiments, Linear SVM, Radial SVM and Logistic Regression were combined in a Voting classifier to make predictions on the dataset with only the identified top 4 features along with Hyperparameter tuned XGBoost.

Results

Model Evaluation and Validation

The final model I chose is an Ensemble model. Ensemble methods are techniques that create multiple models and then combine them to produce improved results. Ensemble methods usually produce more accurate solutions than a single model would. The models used to create such ensemble models are called ‘base models’.

The base models consist of:
radial_svc=svm.SVC(kernel='rbf', probability=True)
linear_svc=svm.SVC(kernel='linear', probability=True)
lr=LogisticRegression()

The final model utilizes Voting Ensemble. Voting is one of the simplest ways of combining predictions from multiple machine learning algorithms. It works by first creating two or more standalone models from your training dataset. A Voting Classifier can then be used to wrap your models and average the predictions of the sub-models when asked to make predictions for new data.

I used a weighted Voting Classifier and the weights were assigned to the classifiers according to their accuracy. So, the classifier with the best accuracy was assigned the highest weight and so on.

In our case, I use the Top 3 classifiers i.e. Linear SVM, Radial SVM, and Logistic Regression classifiers.

The final classifier looks like this:
ensemble_lin_rad_lr=VotingClassifier(estimators=[('Linear_svm', linear_svc), ('Radial_SVM', radial_svc),('Logistic Regression', lr)], voting='soft', weights=[0.2, 0.1, 0.1])

The F1 score of the final model is **0.651632** and accuracy is **0.785**.

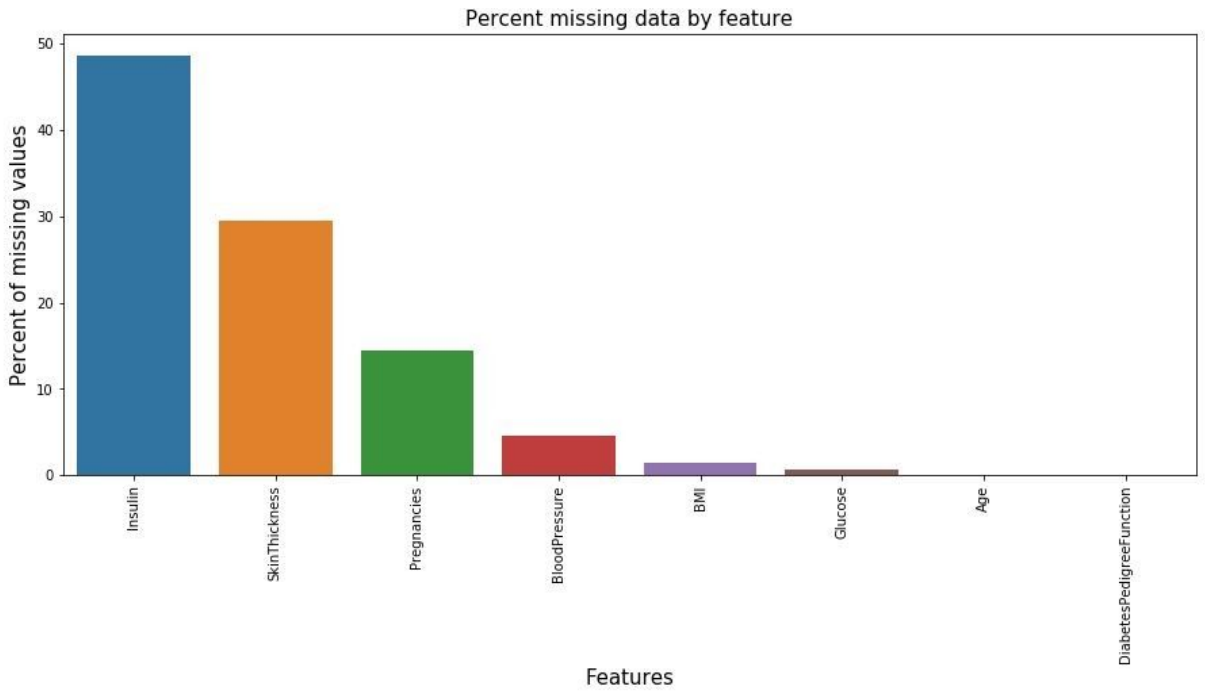
Justification

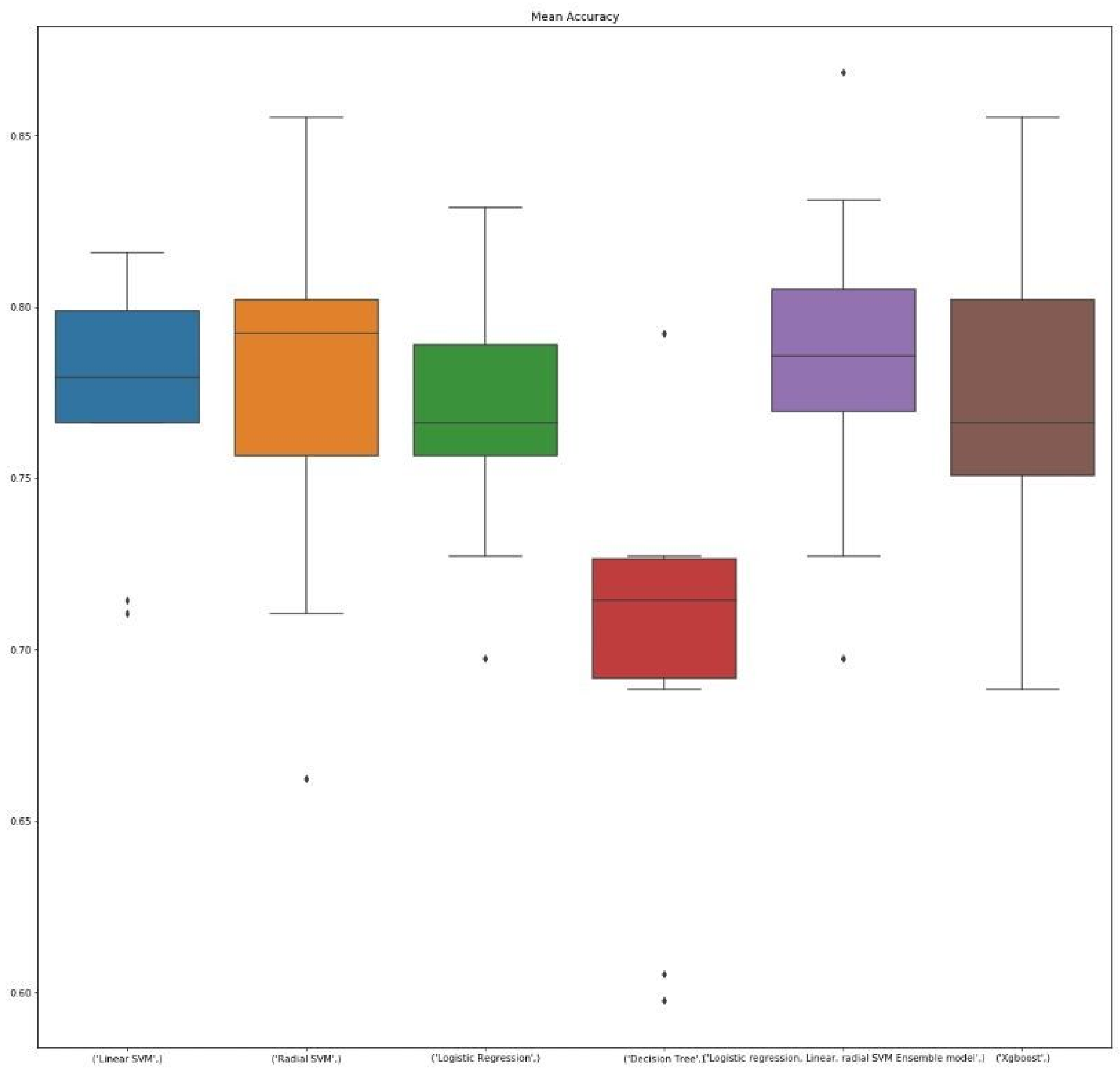
The average **F1 score** and **Accuracy** of **0.651632** and **0.785**, which is higher than the benchmark Logistic regression models score of **0.631333** and **0.769515**. But, we must take into consideration that the input data issued by Kaggle was dirty. The important features had up to 374 missing values in this case, which makes it a decent enough model. As good data is better than a good algorithm, and an accuracy score of 0.65 is considered acceptable, we consider this an acceptable model, but not a great model by any means. The kaggle issued dataset contained 374 missing values for Insulin, 227 for triceps skin thickness. If we deleted the rows with incomplete data it would result in loss of more than half of the dataset in our case, Hence it had to be preprocessed and the missing values replaced by mean and medians. Hence, the final model is not very strong. It is a decent result, when compared with the scores in kaggle, and considering that it is above 69% accuracy, it is still an acceptable model. So, the problem is considered to be solved, to the extent of the quality of the data.

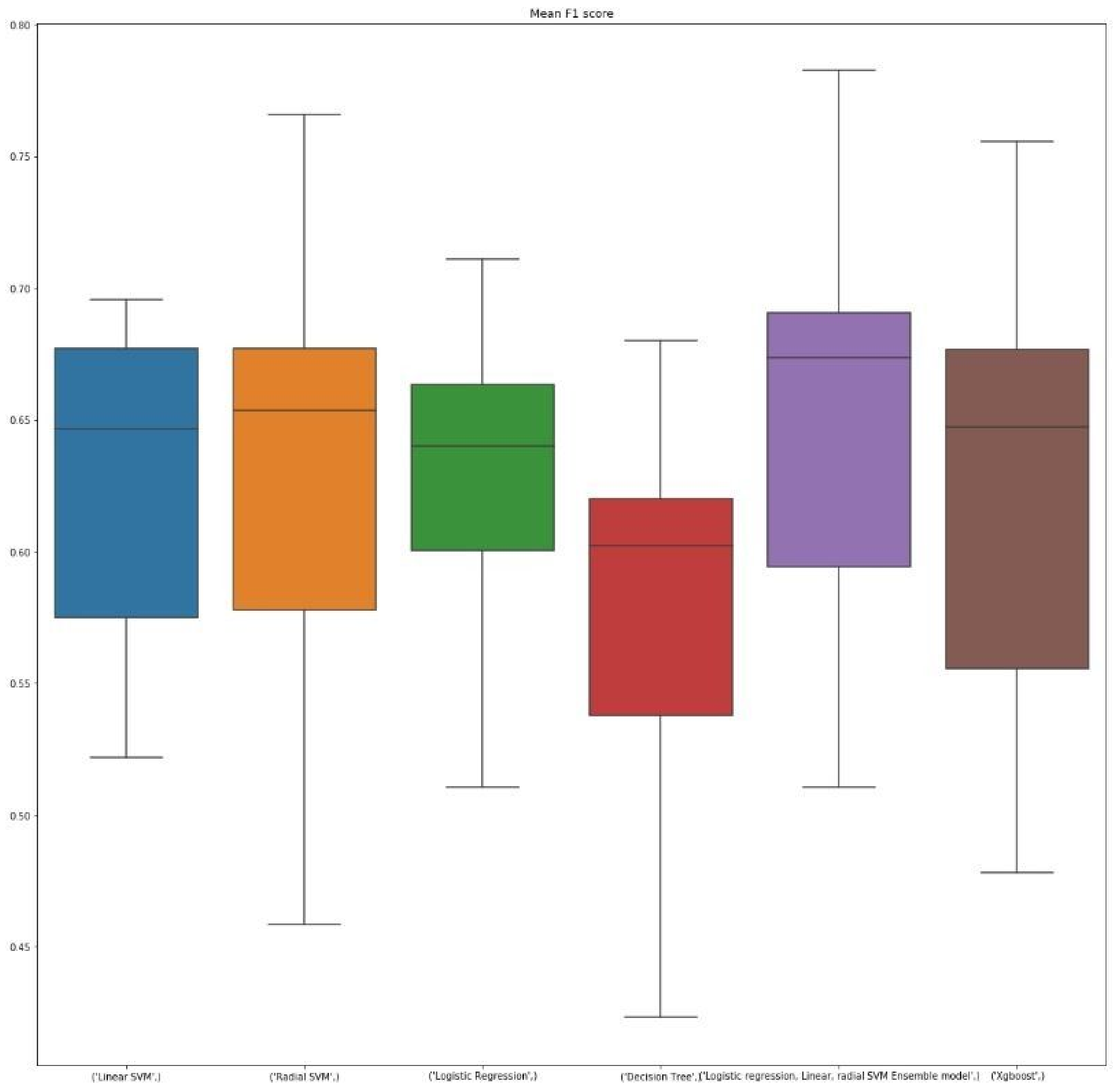
Conclusion

Free-Form Visualization

The following is the visualization of the number of missing features in the dataset in the features with missing values. These missing values deteriorated the quality of the dataset, making it a challenging task to fit a classifier that performs the classification tasks thoroughly.




Mean Accuracy
0.85
0.80
0.75
0.70
0.65
0.60
('Linear SVM',)
('Radial SVM',)
('Logistic Regression',)
('Decision Tree',)
('Logistic regression, Linear, radial SVM Ensemble model',)
('Xgboost',)



Above box plots compare the mean F1 and Accuracy scores of all the chosen classifiers in the project.

Reflection

The problem solution is summarized as follows, importing the necessary libraries, reading the dataset, looking at the statistics of the dataset, identifying the abnormality with missing values and dataset. Discarding them would shorten the dataset considerably. Therefore, replacing them with their mean and median values accordingly. Separating the feature variables and class labels, then scaling the feature variables to be fitted to the classifier, separating the features and labels into training and testing states, setting a random state. Creating the classifier, fitting the data, noting down the F1 scores and Accuracy. Trying different classifiers. Importing the library Voting Classifier and fitting the classifiers, trying different base and blender combinations, trying different no of folds. Checking the F1 scores over different random states to verify that the model has generalized well. Dealing with the dirty data and preprocessing was a difficult task, also tuning the various classifiers and choosing the right classifiers after checking their scores and performance was a tedious task. XGBoost surprisingly resulted in a lower score than Linear SVM and Radial SVM.

Improvement

There could be a better combination of algorithms or tuning, which I haven't come across, it's always a probability, that better-cleansed data, would have yielded better results. Also, if the intuition was applied on Glucose and Insulin to cleanse the data, by shifting missing values to 3rd quartiles in case diabetes was present, and kept its median in absence of diabetes, maybe the classifiers would have responded well. It will also be interesting to see how Neural Nets perform on this dataset. I have set the current solution as my benchmark as this is the best result I was able to get to.

References

1. Joshi SR, Parikh RM. India - diabetes capital of the world: now heading towards hypertension. J Assoc Physicians India. 2007;55:323–4. [PubMed] [Google Scholar]

2. Kumar A, Goel MK, Jain RB, Khanna P, Chaudhary V. India towards diabetes control: Key issues. Australas Med J. 2013;6(10):524–31. [PMC free article] [PubMed] [Google Scholar]

3. Wild S, Roglic G, Green A, Sicree R, King H. Global prevalence of diabetes-estimates for the year 2000 and projections for 2030. Diabetes Care. 2004;27(3):1047–53.[PubMed] [Google Scholar]

4. Whiting Dr, Guariguata L, Weil C, Shawj. IDF Diabetes atlas: Global estimates of the prevalence of diabetes for 2011 and 2030. Diabetes Res Clin Pract. 2011;94:311–21.[PubMed] [Google Scholar]

5. Anjana RM, Ali MK, Pradeepa R, Deepa M, Datta M, Unnikrishnan R, Rema M, Mohan V. The need for obtaining accurate nationwide estimates of diabetes prevalence in India - rationale for a national study on diabetes. Indian J Med Res. 2011;133:369–80.[PMC free article] [PubMed] [Google Scholar]

6. Zargar AH, Khan AK, Masoodi SR, Laway BA, Wani AI, Bashir MI, Dar FA. Prevalence of type 2 diabetes mellitus and impaired glucose tolerance in the Kashmir Valley of the Indian subcontinent. Diabetes Res Clin Pract. 2000;47(2):135–46. [PubMed] [Google Scholar]

7. Ramachandran A, Snehalatha C, Kapur A, Vijay V, Mohan V, Das AK, Rao PV, Yajnik CS, Prasanna Kumar KM, Nair JD. Diabetes Epidemiology Study Group in India (DESI). High prevalence of diabetes and impaired glucose tolerance in India: National Urban Diabetes Survey. Diabetologia. 2001;44(9):1094–101. [PubMed] [Google Scholar]

8. Arora V, Malik JS, Khanna P, Goyal N, Kumar N, Singh M. Prevalence of diabetes in urban Haryana. Australas Med J. 2010;3(8):488–94. [Google Scholar]

9. Bramley D, Hebert P, Jackson R, Chassin M. Indigenous disparities in disease-specific mortality, a cross-country comparison: New Zealand, Australia, Canada, and theUnited States. N Z Med J. 2004;117(1207):U1215. [PubMed] [Google Scholar]

10. Sukala WR, Page RA, Rowlands DS, Lys I, Krebs JD, Leikis MJ, Cheema BS. Exercise intervention in New Zealand Polynesian peoples with type 2 diabetes: Cultural considerations and clinical trial recommendations. Australas Med J. 2012;5(8):429–35. [PMC free article][PubMed] [Google Scholar]

11. Khalil H, George J. Diabetes management in Australian rural aged care facilities: A cross-sectional audit. Australas Med J. 2012;5(11):575–80. [PMC free article] [PubMed] [Google Scholar]