RUTGERS

THE STATE UNIVERSITY
OF NEW JERSEY

FINAL PROJECT REPORT

Bayesian Classification on Diabetes dataset

Jayanth Dasamantharao

Table of Contents

ABSTRACT

The objective of this project is to predict whether a patient tested positive or negative for diabetes based on several input variables including insulin level and age. To achieve this goal, Bayesian logistic regression is used, which involves applying prior distributions to the data before modeling. The PyMC3 package in Python is utilized to perform Markov Chain Monte Carlo simulations and Maximum a posteriori estimation to approximate the posterior distribution and calculate beta coefficients, respectively. I experimented with various prior distributions, including uniform and normal distributions, and compared the results with those obtained from standard logistic regression techniques. The findings of this study may have important implications for the accurate prediction of diabetes outcomes in patients.

INTRODUCTION

Diabetes is a chronic illness that affects millions of people worldwide and is associated with several health complications. The ability to accurately predict and detect diabetes in its early stages can significantly improve patient outcomes. However, creating an accurate classification model can be challenging, especially with small datasets. Traditional logistic regression techniques may not be suitable for such cases, and Bayesian classification provides a more robust alternative by incorporating prior probabilities into the model.

This project aims to predict the outcome of patients with diabetes or not, utilizing a diabetes dataset. The main objective is to develop a robust model using Bayesian classification with different prior distributions and compare their results to identify the best-fit model. The PyMC3 package in Python is used to perform Markov Chain Monte Carlo simulations and Maximum a posteriori estimation to approximate the posterior distribution and calculate beta coefficients.

The prior distribution chosen has a significant impact on the model's performance and predictive power. In this study, I explored different prior distributions, including uniform and normal distributions, and compare their results to standard logistic regression techniques. The findings of this study may help to develop a more accurate and robust classification model for diabetes prediction.

Early detection and management of diabetes are essential for reducing its adverse health effects. The development of an accurate prediction model can assist in identifying patients at high risk of developing diabetes, enabling timely interventions, and improving patient outcomes. By utilizing Bayesian classification with various prior distributions, this project aims to develop a reliable model for diabetes prediction, which can be a valuable tool in clinical practice.

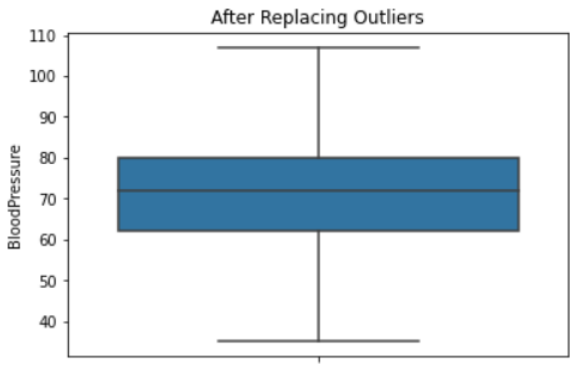
MATERIAL AND METHODS

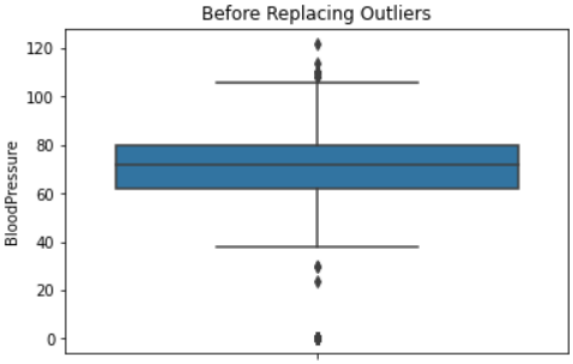
- **Dataset**

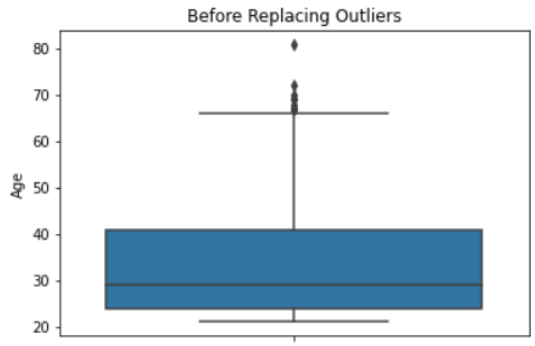
  The dataset used in this study is obtained from Kaggle and comprises 768 observations of various medical predictor variables along with one outcome variable with no missing values. The predictor variables consist of factors such as the patient's number of pregnancies, BMI, insulin levels, age, blood pressure, among others. The outcome variable indicates whether the patient has been diagnosed with diabetes or not, with a value of 1 representing "yes" and a value of 0 representing "no".

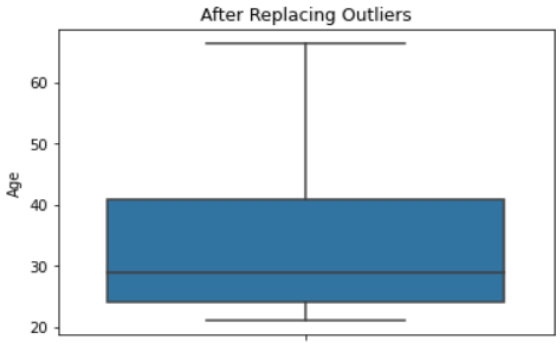
Exploratory Data Analysis

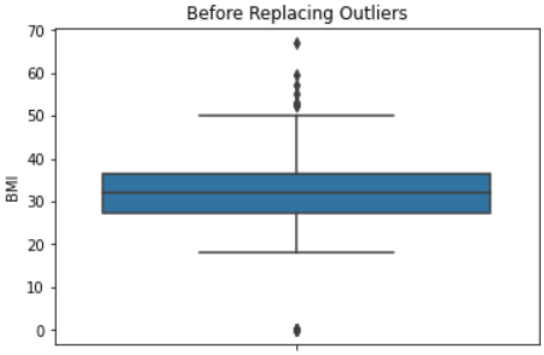
To begin with, I employed techniques for data visualization to gain further insights into the data. During this process, I discovered that certain features such as Insulin, Pregnancies, Glucose, among others, exhibited outliers. Therefore, I proceeded to cap these outliers using suitable quantile values by replacing their values with the whiskers. Whenever a value exceeds the whisker value, it is replaced with the whisker value. Before are some boxplots for a few features before and after replacing the outliers.

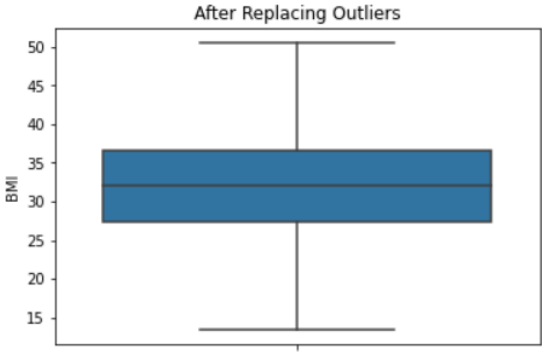




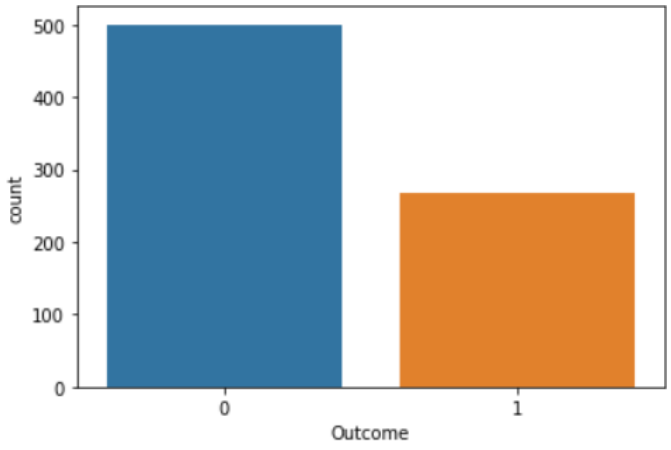




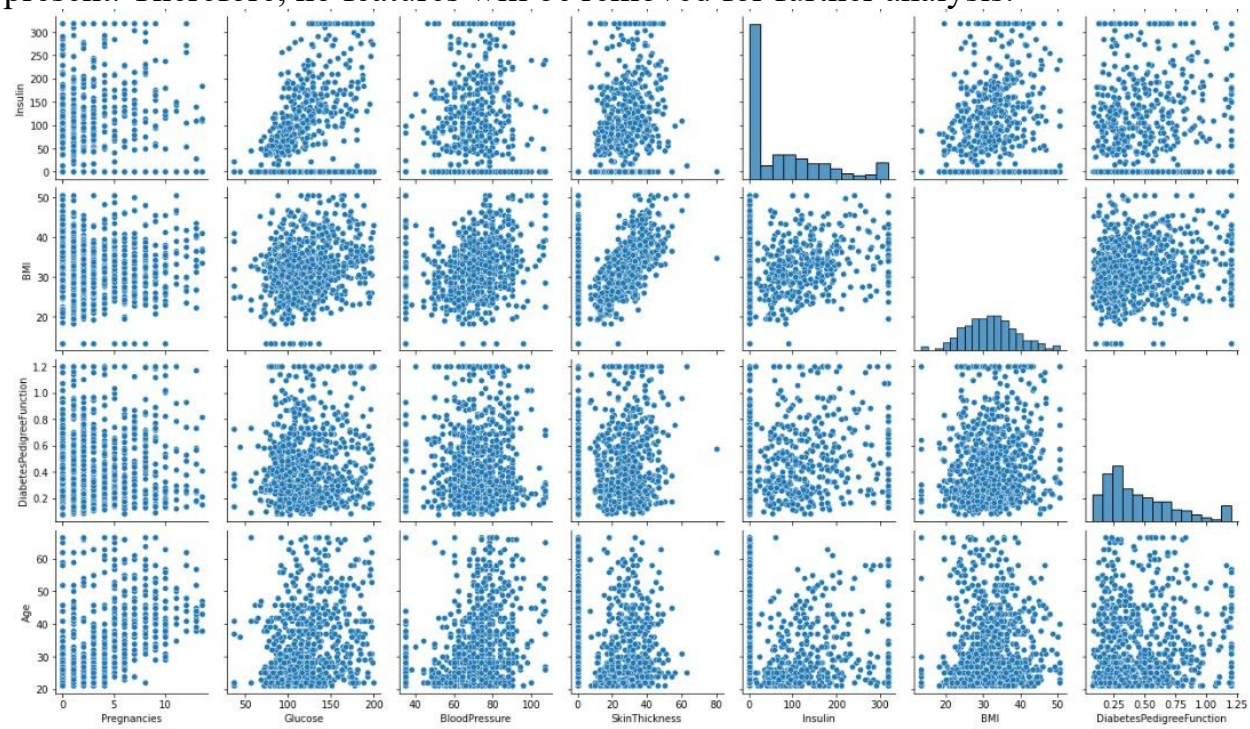




Given that imbalanced data is a common issue in classification problems, it is noteworthy that upon examining the diabetes data shown in the image below, I can see that there is no imbalance between the various target classes. Therefore, there is no need for resampling this data. The image below illustrates that there are approximately 500 instances where patients are not affected by diabetes (labeled as 0) and around 300 instances where patients are affected by diabetes (labeled as 1).



To ensure there are no correlated input features, I utilized the pairplot function from the Seaborn package. Upon examination of the resulting image below (showing only for few features), I have confirmed that there are no correlated variables present. Therefore, no features will be removed for further analysis.



Bayesian classification

Bayesian classification and frequentist classification are two different approaches to classification problems in machine learning.

In Bayesian classification, prior knowledge is used to compute the probability of an event occurring. This is done by combining the prior knowledge with new evidence to obtain a posterior probability. In this approach, probabilities are used to represent uncertain or incomplete information about a problem.

On the other hand, frequentist classification is based on statistical inference using a sample of data. The probability of an event occurring is estimated by the relative frequency of the event in the sample data. This approach assumes that the sample data accurately represents the underlying population.

Bayesian classification is generally used when there is limited or uncertain data available, and prior knowledge is important to make decisions. Frequentist classification, on the other hand, is generally used when there is a large amount of data available, and the emphasis is on using the data to make decisions.

In summary, Bayesian classification is preferred when prior knowledge is available and there is limited or uncertain data available, while frequentist classification is preferred when there is a large amount of data available, and the emphasis is on using the data to make decisions.

Steps in Bayesian classification

High-level steps involved in Bayesian classification:

1. To build a Bayesian logistic regression model, we must first put a prior distribution on each parameter. The choice of these priors will affect the outcome.
2. Once our priors are specified, PyMC3 will numerically approximate the posterior distributions using Markov Chain Monte Carlo simulations and its generalizations.
3. The posterior sample can be visualized by two commonly used plots: a trace plot, showing samples across iterations, or a plot of the empirical density of the posterior sample.
4. Compute the beta coefficients for the MAP solution.

PERFORMANCE METRICS

Following are some performance metrics used in the classification to evaluate model performance:

Accuracy

- Accuracy is the proportion of true results among the total number of cases examined.
- Accuracy = (TP+TN)/(TP+FP+FN+TN)
- Accuracy is a valid choice of evaluation for classification problems which are well balanced and not skewed or No class imbalance.

Precision

- Precision - what proportion of predicted Positives is truly Positive
- Precision = (TP)/(TP+FP)
- Precision is a valid choice of evaluation metric when we want to be very sure of our prediction.

Recall

- Recall - what proportion of actual Positives is correctly classified
- Recall = (TP)/(TP+FN)
- Recall is a valid choice of evaluation metric when we want to capture as many positives as possible.
- Recall is 1 if we predict 1 for all examples.

F1 Score

- F1 score - The F1 score is a number between 0 and 1 and is the harmonic mean of precision and recall.
- F1 = 2*(P*R)/(P+R)
- When to use? - We want to have a model with both good precision and recall.

- Simply stated the F1 score sort of maintains a balance between the precision and recall for your classifier. If your precision is low, the F1 is low and if the recall is low again your F1 score is low.
- The main problem with the F1 score is that it gives equal weight to precision and recall. We might sometimes need to include domain knowledge in our evaluation where we want to have more recall or more precision. To solve this, we can do this by creating a weighted F1 metric as below where beta manages the tradeoff between precision and recall.

RESULTS

Logistic Regression

By employing a conventional logistic regression model, the results obtained indicate a recall of 88% for class 0 and 59% for class 1. The accuracy is 78% for the model. Since the dataset is very small, the accuracy is not very high.

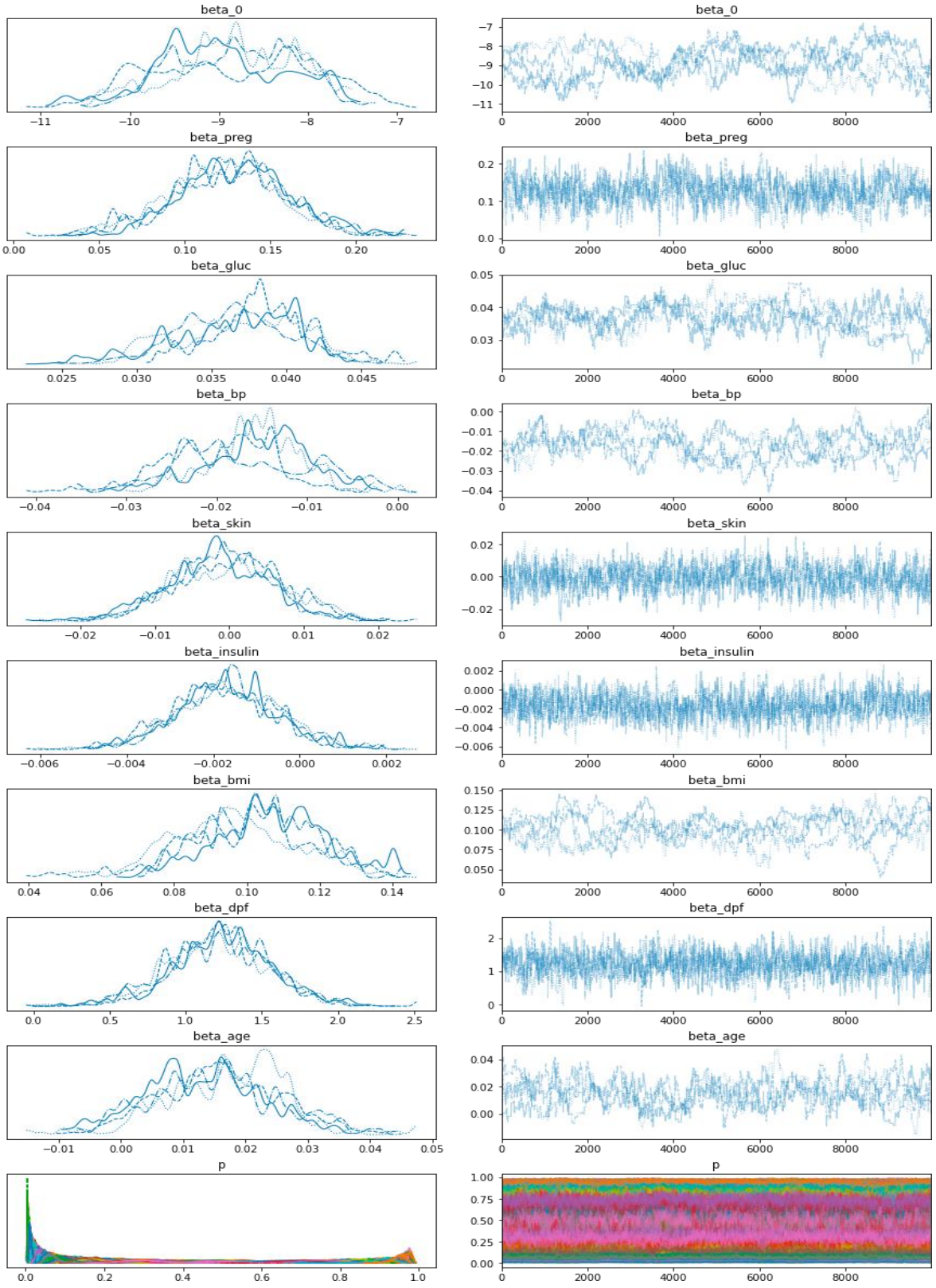
	precision	recall	f1-score	support
0	0.80	0.88	0.84	500
1	0.73	0.59	0.66	268
accuracy			0.78	768
macro avg	0.77	0.74	0.75	768
weighted avg	0.78	0.78	0.78	768

The following are the coefficients obtained from the logistic regression model for each input feature.

	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	0.035453	-0.014271	-0.000881	-0.001612	0.09763	1.04803	0.033413

Bayesian Models - Uniform Prior

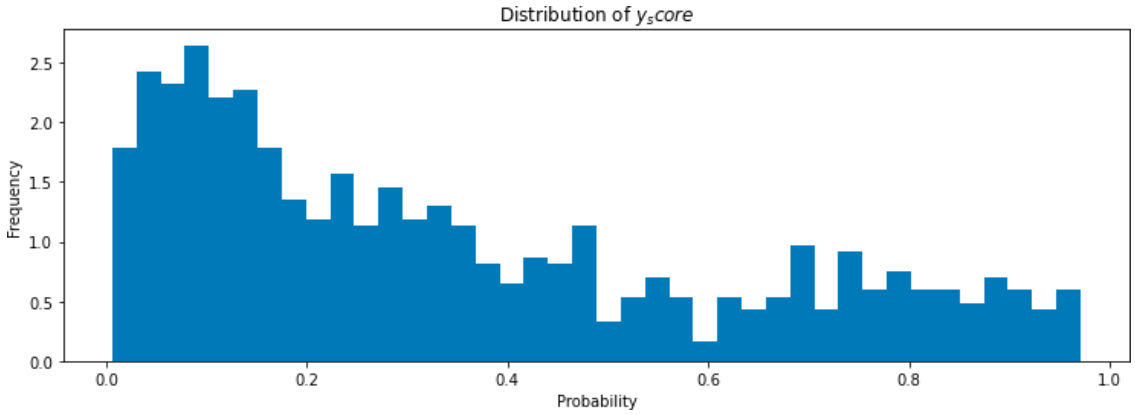
The image below displays 4 distinct chains running with uniform distribution as its Prior, which is a prevalent technique in Bayesian inference to assess convergence and detect any mixing or convergence issues. PyMC3 typically runs 4 chains by default, and the Gelman-Rubin diagnostic is used to ensure they have reached the same stationary distribution. Combining the chains in the context of Bayesian inference refers to the process of merging the samples obtained from multiple Markov chains that were run in parallel during the MCMC sampling procedure. The goal of combining the chains is to obtain a better estimate of the posterior distribution by reducing the effects of chain-specific fluctuations and increasing the effective sample size.



Trace plots when Uniform Priors are used for Bayesian Classification

From the above image, I can conclude that our algorithm does converge and shown beloware the mean of these posterior distributions.

	beta_0	beta_preg	beta_gluc	beta_bp	beta_skin	beta_insulin	beta_bmi	beta_dpf	beta_age
0	-9.188321	0.125715	0.037784	-0.015136	-0.001734	-0.001592	0.104774	1.288009	0.014999

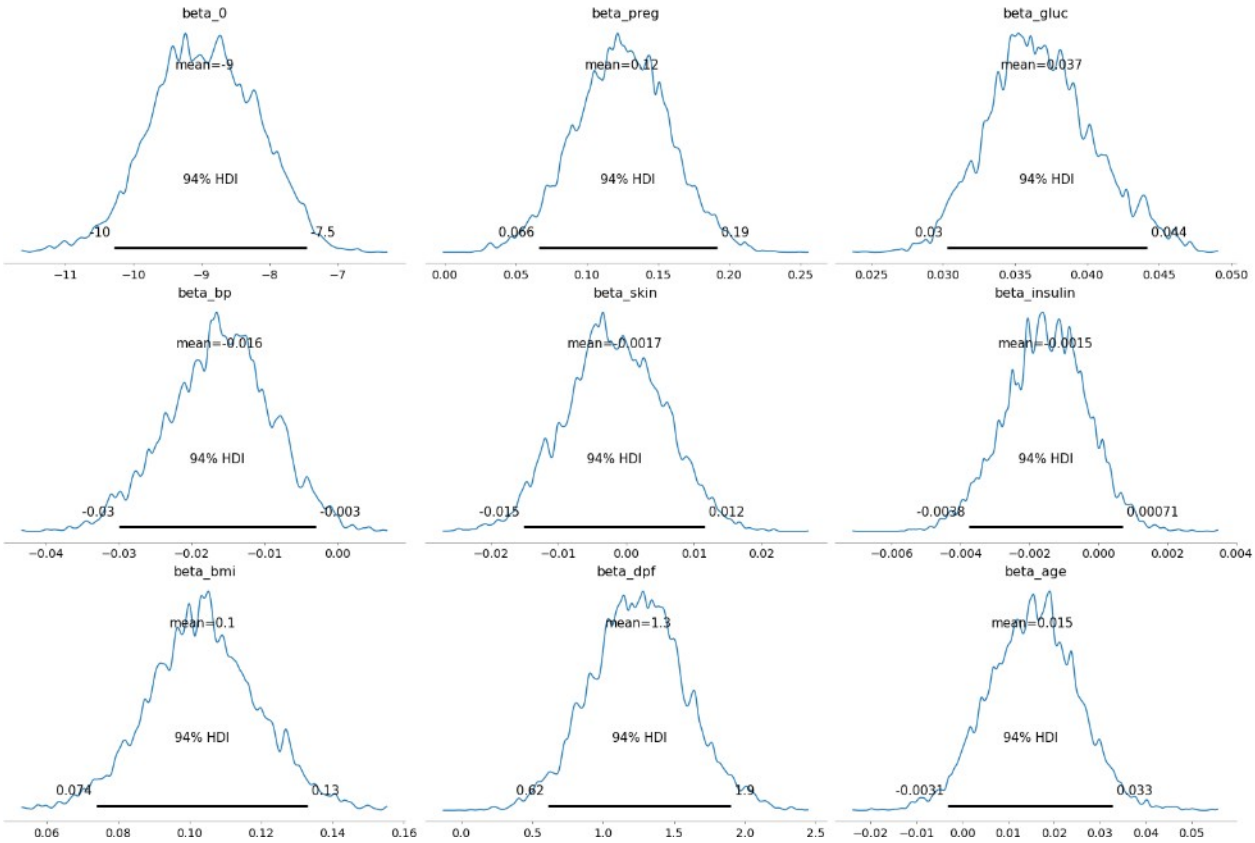


Computing the y_score by averaging over our sample values

```
#ROC-AUC
pred_scores = dict(y_true=df1['Outcome'],y_score=y_score)
roc_auc_score(**pred_scores)
```

```
0.8426940298507462
```

ROC-AUC score using Uniform priors for Bayesian classification.



Plot of posteriors using Uniform Priors

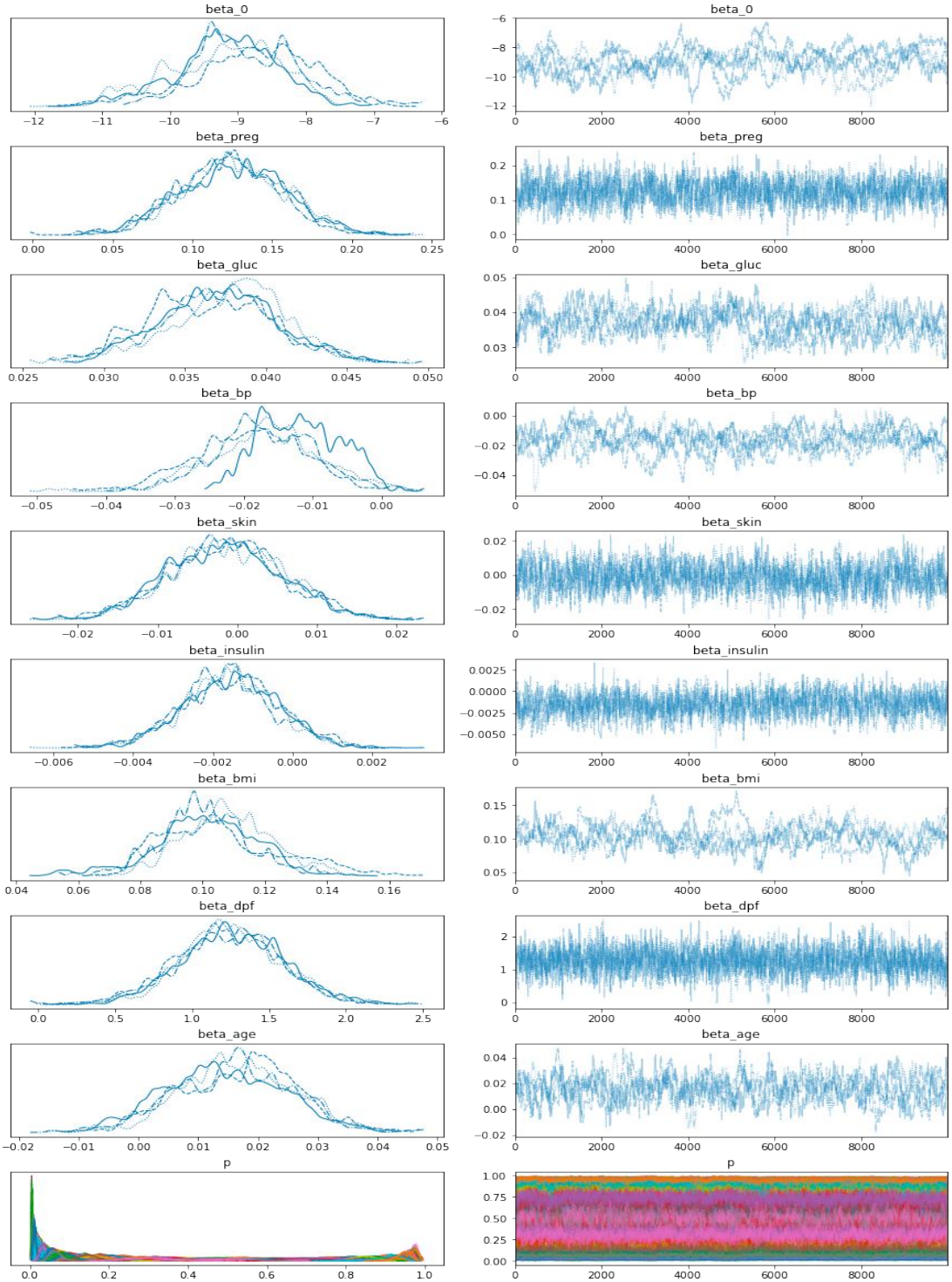
By employing a Bayesian logistic regression model with Uniform distribution as its Prior, the results obtained indicate a recall of 89% for class 0 and 59% for class 1.

	precision	recall	f1-score	support
0	0.79	0.89	0.84	500
1	0.74	0.57	0.64	268
accuracy			0.78	768
macro avg	0.77	0.73	0.74	768
weighted avg	0.77	0.78	0.77	768

Performance metrics using Uniform priors for Bayesian classification.

Bayesian Models - Normal Prior

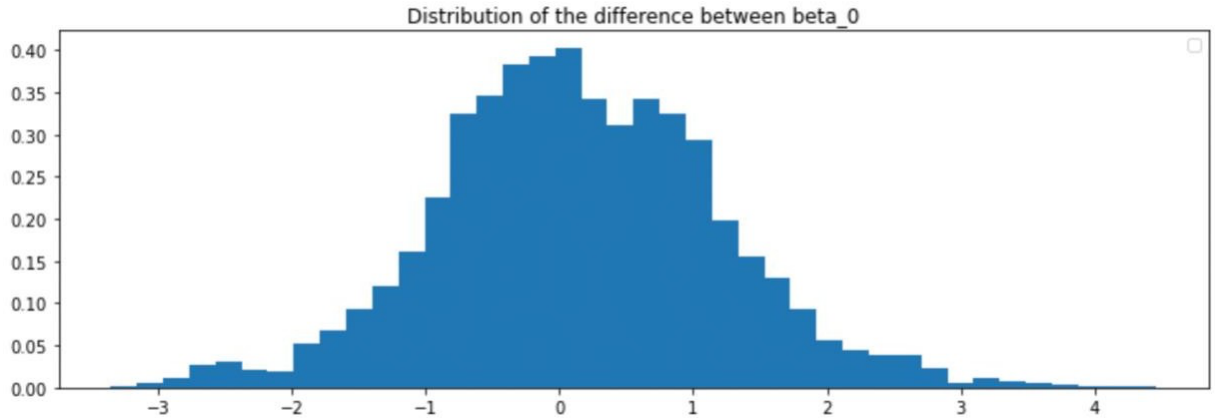
Now training our model using normal distribution as its Prior. The image below displays 4 distinct chains running if the coefficients follow normal distributions.



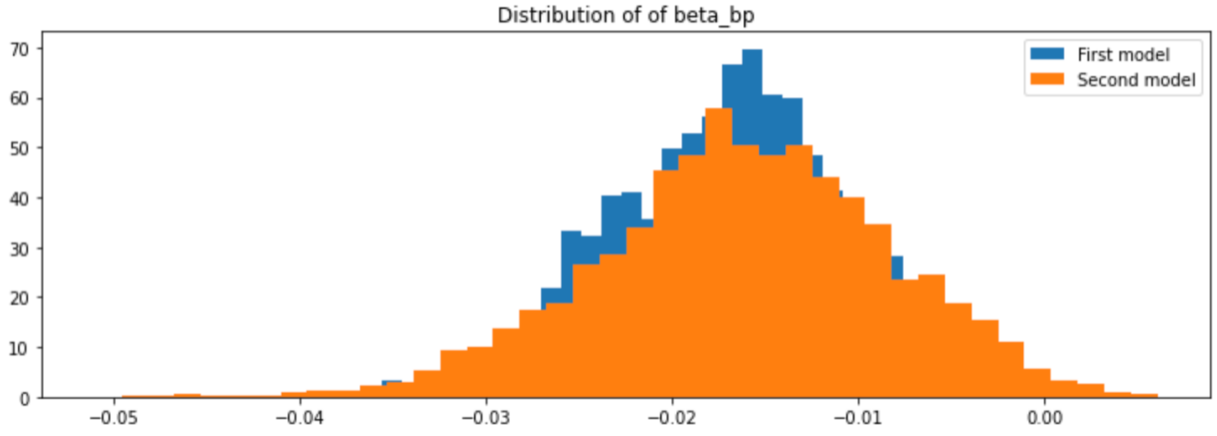
Trace plots when Normal Priors are used for Bayesian Classification

The above image, I can conclude that our algorithm does converge and shown beloware the mean of these posterior distributions.

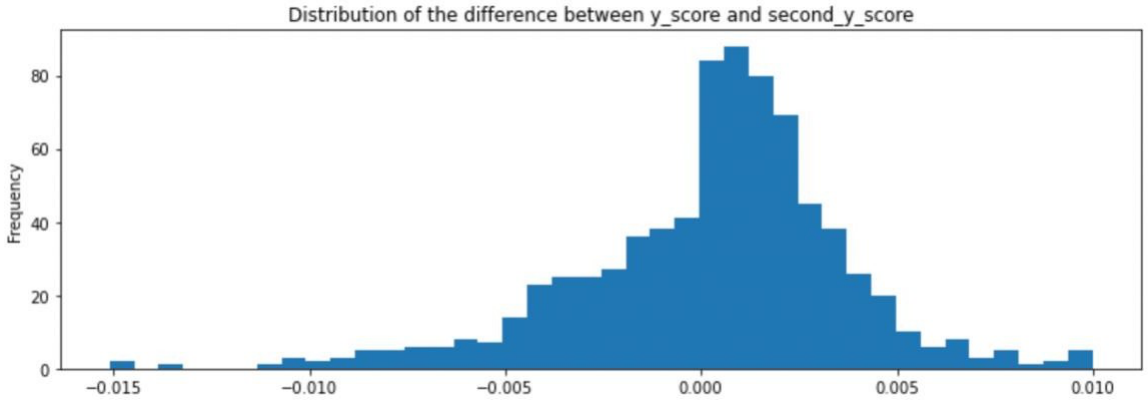
	beta_0	beta_preg	beta_gluc	beta_bp	beta_skin	beta_insulin	beta_bmi	beta_dpf	beta_age
0	-8.784884	0.12283	0.036297	-0.015709	-0.001265	-0.001549	0.100327	1.225135	0.015293



Comparing the posterior distributions across the two models



Comparing the posterior distributions of β_temp between these two models



Comparing the difference between y_score and second_y_score

Performance metrics using Normal priors for Bayesian classification:

By employing a Bayesian logistic regression model with Normal distribution as its Prior, the results obtained indicate a recall of 89% for class 0 and 57% for class 1.

	precision	recall	f1-score	support
0	0.79	0.89	0.84	500
1	0.73	0.57	0.64	268
accuracy			0.78	768
macro avg	0.76	0.73	0.74	768
weighted avg	0.77	0.78	0.77	768

Performance metrics using Normal priors for Bayesian classification.

DISCUSSIONS

Based on the aforementioned outcomes, it appears that the logistic regression, Bayesian with a uniform prior, and Bayesian with a normal prior yield nearly identical recall. But employing Bayesian methodology could be more beneficial, particularly when working with limited data. When dealing with a small dataset in a frequentist model, outliers can cause an improper fit. On the other hand, using a Bayesian approach, a prior distribution over the parameters can act as a form of regularization, preventing improbable extreme values. Moreover, if we possess a better understanding of the data distribution, we could have utilized more suitable priors, resulting in superior outcomes than those obtained with a frequentist approach.

To implement Bayesian methodologies over a chain of MCMC samples, it is computationally not efficient on a general local machine. Certain optimizers from the theano module on python are used to handle the effect of splitting the running of thousands of samples across multiple cores of a machine. Provided that we have a machine with excellent specification of cores or a virtual machine, Bayesian classification can yield results quickly as equivalent to the frequentist logistic regression.

LITERATURE CITED

https://towardsdatascience.com/bayesian-statistics-101-4c4bc5fde1e1
https://towardsdatascience.com/bayesian-logistic-regression-with-pymc3-8e17c576f31a