**Project Report:**

**1. Title:**

Comparison of Generative and Discriminative Classifiers – Naïve Bayes Classifier v/s Logistic Regression Classifier

**2. Authors:**   
 a. Heeru Srichand Ahuja

b. Tanmay Khokle

c. Narsing Rao Akula

**3. GitHub:**   
 <https://github.com/heeruahuj/DS5010-PROJECTREPORT/>

**4. Summary:**   
  
Our project aims to deliver a tool that can perform classification on a data set using two classifiers- the Naïve Bayes classifier and the Logistic Regression Classifier. This package will not only enable us to perform classification on our data set but also compare between generative and discriminative classifiers.   
The Naïve Bayes classifier is a probabilistic classifier and works on the Bayes Theorem with the assumption of the features or predictors being independent of each other. It models the joint distribution p(x,y) in the form p(x|y)p(y), which is derived from Bayes Theorem, assuming that the features are independent. It is a generative classifier. The Logistic Regression classifier is a predictive analysis that is used to describe data and explain the relationship between one dependent variable and one or more independent variables. It is used to model conditional probability (p(y|x)) by maximizing the likelihood. With the support of NumPy and Pandas, we will implement the Naïve Bayes Classifier and the Logistic Regression Classifier. We aim to implement both without their built-in functionalities.

**5. Design:**  
As part of our design for the implementation, we will be constructing one module (for e.g., Classifier.py), with one class for each classifier (i.e., Naïve Bayes Classifier and Logistic Regression Classifier).    
The class for Logistic regression will contain methods for:

* Calculation of the sigmoid
* Calculation of the cost function
* Using gradient descent to fit the training data.
* Prediction on a test set

The class for Naïve Bayes classifier will contain methods for:

* Calculation of the posterior probabilities
* Calculation of the prior
* Fitting the model to the training data
* Prediction on a test set

To enable robustness, we are implementing our project as a package that can be imported and both the classifiers can be directly applied onto a dataset.    
For vector operations and handling data in our implementation, we will be using NumPy, pandas, and matplotlib. To summarize, we will be implementing the classifiers from scratch without using their inbuilt functions.    
One hurdle we may face during our implementation could be the implementation of a flexible code for multiple datasets or large datasets. Additionally, we want to leave room for the user to modify or customize according to their requirements

.

**6. Usage:**

**7. Discussion:**

**8. Statement of contributions:**   
 a. Heeru Srichand Ahuja

b. Tanmay Khokle

c. Narsing Rao Akula

**9. References:**