# 

Assignment#2: Machine Learning: Naive Bayes Classifier Model.

Swapnali Dashrath

Oakland University

CSI6160 - Machine Learning

Prof. Mohammad Wardat.

October 12, 2024.

**Naïve Bays Modeling and Evaluation Report**

This report presents the implementation and evaluation of two Naive Bayes classification model using Iris simple datasets using an ML library and also builds their own model from scratch. on Iris Dataset using existing libraries (numpy, pandas, seaborn, matplotlib etc). Implementation is using Python (code file submitted along with report) and Evaluation is based on two evaluation metrics from Accuracy, Precision, Recall and F1-Score.Also, this report explore the building the model from scratch.

**Problem Statement:**

We are trying to use attributes of flowers to predict the species of the flower, specifically, we are trying to use Sepal length, Sepal width, Petal length, Petal width to predict if an Iris flower is of species setosa' 'versicolor' 'virginica based on Naïve Bays Algorithm as well as building model from scratch.

**Approach:**

1. Analyzing Data – Iris Dataset
2. Data Cleanup and formatting using Data Frames.
3. Visualize the data using Data Plots.
4. Defining Relationship of data features with target.
5. Exploratory Data Analysis (EDA) – Pairplots.
6. Training Data Split.
7. Data prep for modeling
8. Modeling using Naïve Bays Classifier.
9. Evaluation based on Accuracy, precision, recall and F1 score.

**Iris Dataset:**

The dataset consists of 150 samples of iris flowers, divided into three species: Setosa, Versicolor, and Virginica, with four features: sepal length, sepal width, petal length, and petal width.

**Classifiers Implemented**

1. Naïve Bays using existing Library
2. Building Model using existing Model

**Data Splitting**

The dataset was split into training (70%) and testing (30%) sets. This allows us to train the models on the training data and evaluate them on unseen test data.

**Modeling**

Our base line model is just randomly guessing the species of flower, or guessing a single species for every data point with certain accuracy.

**Evaluation Metrics**

We used the following metrics to evaluate the models:

* Accuracy: Proportion of correctly classified instances.
* Confusion Matrix: Summarizes the performance by showing true positives, false positives, true negatives, and false negatives.

**Results**

**1**. **Naïve Bays Approach – GaussianNB Classifier**

* **Accuracy**: 100% (accuracy on the test set), For different test data set model provided 100% accuracy.

Accuracy: 100.00%

precision recall f1-score support

0 1.00 1.00 1.00 10

1 1.00 1.00 1.00 9

2 1.00 1.00 1.00 11

accuracy 1.00 30

macro avg 1.00 1.00 1.00 30

weighted avg 1.00 1.00 1.00 30

1. **Naïve Bays Model building from scratch**

Naive Bayes classification accuracy is 0.965

**Interpretations:**

We are successfully able to achieve 100% accuracy using GaussianNB Classifier using Naïve Bays Model building using existing library and from scratch implementations.

**References**

Almeida, T. & Hidalgo, J. (2011). SMS Spam Collection [Dataset]. UCI Machine Learning Repository. <https://doi.org/10.24432/C5CC84>.

Normalized Neard. (2021),  [Naive Bayes Classifier in Python (from scratch!)](https://www.youtube.com/watch?v=IMrxB8Mq5KU)

[(youtube.com)](https://www.youtube.com/watch?v=IMrxB8Mq5KU) [(174) Naive Bayes Classifier in Python (from scratch!) - YouTube](https://www.youtube.com/watch?v=3I8oX3OUL6I)

CampusX (2021). [Email Spam Classifier | SMS Spam Classifier | End to End Project | Heroku Deployment (youtube.com)](https://www.youtube.com/watch?v=YncZ0WwxyzU) [Email Spam Classifier | SMS Spam Classifier | End to End Project | Heroku Deployment](https://www.youtube.com/watch?v=YncZ0WwxyzU)