**IMPLEMENTATION**

**MODULES:**

* Data Collection
* Dataset
* Data Preparation
* Model Selection
* Analyze and Prediction
* Accuracy on test set
* Saving the Trained Model

**MODULES DESCRIPTION:**

**Data Collection:**

This is the first real step towards the real development of a machine learning model, collecting data. This is a critical step that will cascade in how good the model will be, the more and better data that we get, the better our model will perform.

There are several techniques to collect the data, like web scraping, manual interventions, etc.

**Data Preparation:**

we will transform the data. By getting rid of missing data and removing some columns. First we will create a list of column names that we want to keep or retain.

Next we drop or remove all columns except for the columns that we want to retain.

Finally we drop or remove the rows that have missing values from the data set.

**Model Selection:**

While creating a machine learning model, we need two datasets, one for training and other for testing. But now we have only one. So let’s split this in two with a ratio of 80:20. We will also divide the data frame into feature columns and label columns.

Here we imported the train\_test\_split function of sklearn. Then use it to split the dataset. Also, test\_size *= 0.2*, it makes the split with 80% as train dataset and 20% as test dataset.

The random\_state parameter seeds random number generator that helps to split the dataset.

The function returns four datasets. Labelled them as train\_x, train\_y, test\_x, test\_y*.* If we see the shape of these datasets we can see the split of the dataset.

We used many algorithms like Svm, Linear Regression, Random Forest regression, K Nearest Neighbors (KNN), Decision Trees, LSTM model, SVR (Linear).

**LINEAR REGRESSION:**

In regression modeling, a target class is predicated on the independent features. This method can be thus used to find out the relationship between independent and dependent variables and also for forecasting. Linear regression, a type of regression modeling, is the most usable statistical technique for predictive analysis in machine learning. Each observation in linear regression depends on two values, one is the dependent variable and the second is the independent variable. Linear regression determines a linear relationship between these dependent and independent variables. There are two factors (x, y) that are involved in linear regression analysis. The equation below shows how y is related to x known as regression.

y = β0 + β1x + ε

or equivalently

E(y) = β0 + β1x

Here, ε is the error term of linear regression. The error term here uses to account the variability between both x and y, β0 represents y-intercept, β1 represents slope. To put the concept of linear regression in the machine learning context, in order to train the model x is represented as an input training dataset, y represents the class labels present in the input dataset. The goal of the machine learning algorithm then is to find the best values for β0 (intercept) and β1(coefficient) to get the best-fit regression line. To get the best fit implies the difference between the actual values and predicted values should be minimum, so this minimization problem can be represented as:

Minimize 1 n Xn i=1 (predi − yi) 2

g = 1 n Xn i=1 (predi − yi) 2

Here, g is called a cost function, which is the root mean square of the predicted value of y (predi) and actual y (yi), n is the total number of data points.

**SUPPORT VECTOR MACHINE:**

A support vector machine (SVM) is a type of supervised ML algorithm used for both regression and classification. SVM regression being a non-parametric technique depends on a set of mathematical functions. The set of functions called kernel transforms the data inputs into the desired form. SVM solves the regression problems using a linear function, so while dealing with problems of non-linear regression, it maps the input vector(x) to n-dimensional space called a feature space (z). This mapping is done by non-linear mapping techniques after that linear regression is applied to space. Putting the concept in ML context with a multivariate training dataset (xn) with N number of observations with yn as a set of observed responses. The linear function can be depicted as:

f (x) = x 0β + b

the latter adjusts for the number of features in a prediction model. In the case of R 2 adjusted , the increase in new features can lead to its increase if the newly added features are useful to the prediction model. However, if the newly added features are useless, its value will decrease. The R 2 adjusted can be defined as:

R 2 adjusted = 1 − (1 − R 2 ) n − 1 n − (k + 1) (11)

Here, n is the sample size and k is the number of independent variables in the regression equation.

**MEAN SQUARE ERROR (MSE) :**

Mean square error is another way to measure the performance of regression models [22]. MSE takes the distance of data points from the regression line and squaring them. Squaring is necessary because it removes the negative sign from the value and gives more weight to larger differences. The smaller mean squared error shows the closer you are to finding the line of best fit.

MSE can be calculated as: MSE = 1 n Xn i=1 (yi − ˆyi)

**Naïve BAYES algorithms:**

The Naïve Bayes model is used to resolve classification problems by using probability techniques. The Naïve Bayes algorithm for this article can be denoted as equation

P(Class|WORD)=(P(WORD|Class) × P(Class))/(P(WORD))

where WORD is (word1,word2, . . .wordn) from within an uploaded email and ‘Class’ is either ‘Spam’ or ‘Ham’. The algorithm calculates the probability of a class from the bag of words provided by the program. Where P(Class | WORD) is a posterior probability, P(WORD | Class) is likelihood and P(Class) is the prior probability .

If ‘Class’ = Spam, the equation could be rewritten to find the spam email from the given words, and this can be further simplified as equation

P(Class|WORD) = 0 n i=1 P(word\_i|Spam) × P(Spam)/ P(word\_1, word\_2, . . . word\_n)

There are three types of Naïve Bayes algorithms: Multinomial, Gaussian and Bernoulli. Multinomial Naïve Bayes algorithm has been selected to perform the spam email identification because it is text related and outperforms Gaussian and Bernoulli.

Multinomial Naïve Bayes (MNB) classifier uses Multinomial Distribution for each given feature, focusing on term frequency. The Multinomial Naïve Bayes can be denoted as an equation.

P(p|n) ∝ P(p) Y 1≤k≤nd P(tk |p)

where the number of token is represented by nd, n is the number of emails and P(tk |p) is calculated by:

P(tk|p) = (count(tk|p) + 1)/ (count(tp) + |V|)

In the equations, P(tk |p) is identified as the conditional probability for MNB. The tk is the spam term occurrence within an email and P(p) is classed as the prior probability. 1 and |V| are identified as the smoothing constant for the algorithm.

To test this algorithm, the MNB module was loaded from the Scikit-learn library. The parameters for this model are optional. If none is specified, the default values are: Alpha value set to ‘1.0’, Fit Prior is set to ‘True’ and Class Prior is set to ‘None’ .

The algorithm-1 shows the pseudocode for Multinomial Naïve Bayes with spam classification where ‘‘Tr’’ is Training and ‘‘Te’’ is Testing. The P(ˆ tk |p) is the estimating/predicting variable, also known as the conditional probability.

For future prediction we used Random Forest regression

Random Forest is one of the most powerful methods that is used in machine learning for regression problems. The random forest comes in the category of the supervised regression algorithm. This algorithm is carried out in two different stages: the first one deals with the creation of the forest of the given dataset, and the other one deals with the prediction from the regressor.

**Saving the Trained Model:**

Once you’re confident enough to take your trained and tested model into the production-ready environment, the first step is to save it into a .pkl file using a library like pickle .

Make sure you have pickle installed in your environment.

Next, let’s import the module and dump the model into . pkl file