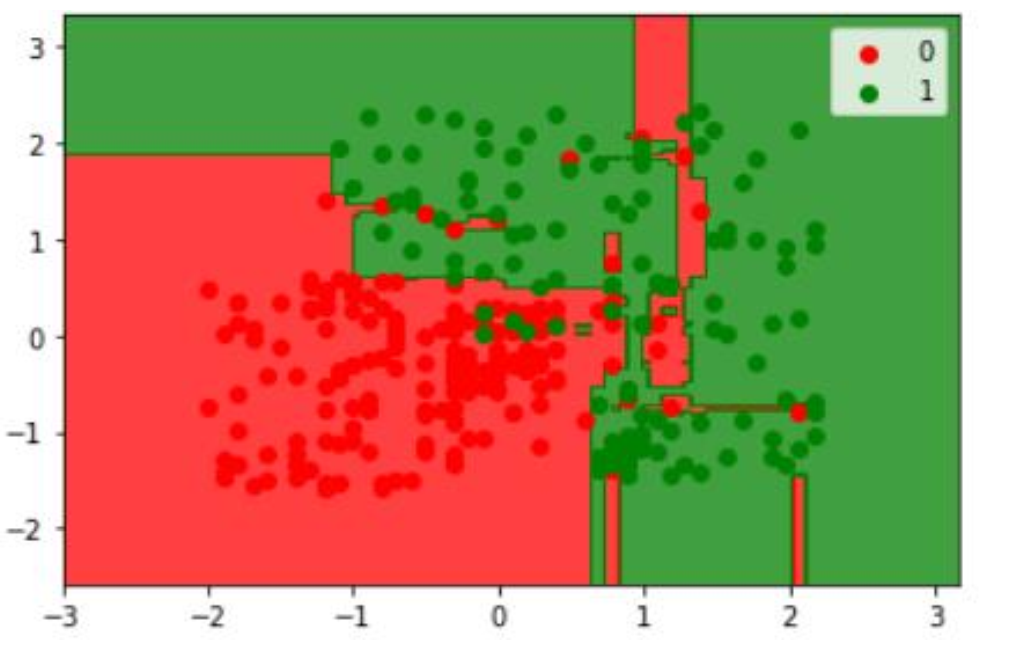
***SOCIAL NETWORK ANALYSIS USING MACHINE LEARNING TECHNIQUES***

**PROBLEM STATEMENT:**

Try to understand the dataset of Social\_Network\_Ads.csv and try to find the best suitable MLalgorithm and write the code in python for algorithm from scratch and try to achieve the belowoutput plot.

****

**SOLUTION:**

The objective of the problem was to investigate the advertising strategy employed by company to which group of people they must advertise more. Our goal was to compare purchasing of the product by person based on sex, age and estimated salary. The dataset (Social\_Network\_ads.csv) contains attributes such as user id, gender, estimate and purchased details. It includes an target/label variable(purchased) ,so it comes under supervised learning. Data preprocessing , it includes importing data, Checking for missing values, splitting data and feature scaling. Model Selection: Since, it is a classification problem. KNN classifier and Naive Bayes classifier are used to compare their results. Reasons for the model:

* **Naive Bayes Classifier**: It uses a similar method to predict the probability of different class based on various attributes. Multinomial Naive Bayes classifier is suitable for multinomial models. It is suitable for classification with discrete  
  features.
* **KNN:** The algorithm can compete with the most accurate models because it makes highly accurate predictions. Its purpose is to use a database in which the data points are separated into several classes to predict the classification of a new sample point.

Comparison of the results between the scratch code and scikit-learn.

**PACKAGES USED:**

* NumPy
* Pandas
* Matplotlib
* Scikit – learn

**ASSUMPTIONS:**

**Hypothesis H1:** The number of males who purchased the product are more than number of females who purchased the product.

**Hypothesis H0:**  The complement of H1.

**DATA PREPROCESSING:**

**Importing dataset:**

dataset = pd.read\_excel('Social\_Network\_Ads.xlsx')

Data is imported using pandas.

**Null values:**

Checking for null values in the data.

Gender 0

Age 0

EstimatedSalary 0

Purchased 0

dtype: int64

The above output contains no null values . so no need for further process.

**Describe( )** :

It is used to view some basic statistical details.

User ID Age EstimatedSalary Purchased

count 4.000000e+02 400.000000 400.000000 400.000000

mean 1.569154e+07 37.655000 69742.500000 0.357500

std 7.165832e+04 10.482877 34096.960282 0.479864

min 1.556669e+07 18.000000 15000.000000 0.000000

25% 1.562676e+07 29.750000 43000.000000 0.000000

50% 1.569434e+07 37.000000 70000.000000 0.000000

75% 1.575036e+07 46.000000 88000.000000 1.000000

max 1.581524e+07 60.000000 150000.000000 1.000000

From the above output, we can infer that the average age of the person is 37 and the minimum estimated salary is 15000 and maximum is 150000.

**Feature scaling:**

[ 0.38358493 1.11381995]

[-1.79512465 -1.3505973 ]

[ 0.18552042 -0.13288524]

[ 0.8787462 -1.43757673]

[-1.99318916 0.47597078]

[-0.30964085 0.27301877]

[ 1.86906873 -1.06066585]

[-0.4086731 0.07006676]

[ 1.07681071 -0.88670699] . . . .

The above output shows the feature scaling for the age and estimated salary.

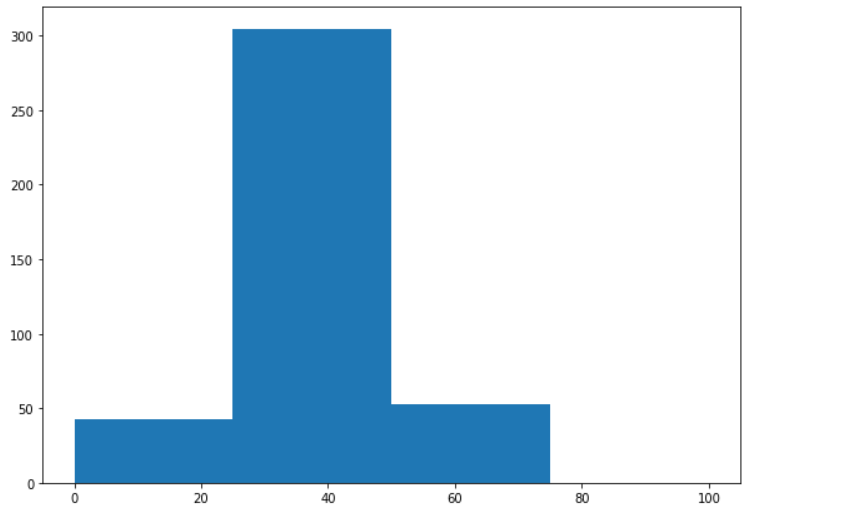
**Data visualization:**

To further analyse, let's first plot a histogram for the 'Age' variable.

a=dataset.iloc[:,[2]].values

fig, ax = plt.subplots(figsize =(10, 7))

ax.hist(a, bins = [0, 25, 50, 75, 100])



The above graph that the variable 'Age' has a normal distribution of data.

**MODEL PREDICTION:**

**Model 1** : K-Nearest Neighbour Classifier

**OUTPUT**

#From scratch

Accuracy on test set by our model : 78.0

Precision 0.6425

recall 0.5

# using scikit learn:

Confusion matrix

[[64 4]

[ 3 29]]

classification report

precision recall f1-score support

0 0.96 0.94 0.95 68

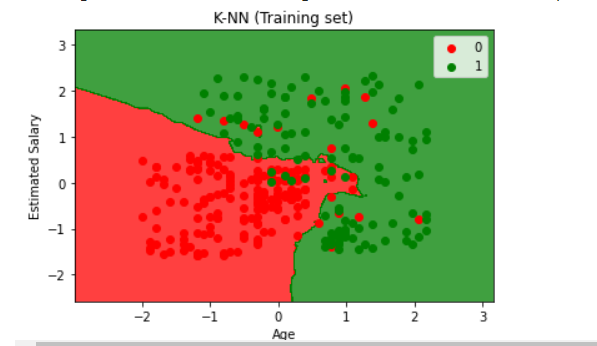
1 0.88 0.91 0.89 32

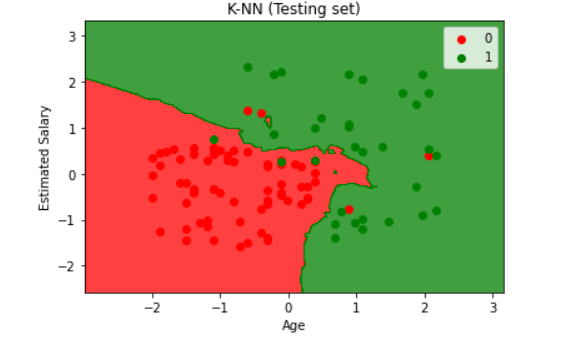
accuracy 0.93 100

macro avg 0.92 0.92 0.92 100

weighted avg 0.93 0.93 0.93 100

**VISUALIZATION OF RESULT:**





**PREDICTING NEW VALUS:**

New values

Age

17

Estimated Salary

23555

Predicting the result:

: [1]

**Model 2:** Naïve Bayes classifier

**OUTPUT:**

classification report

precision recall f1-score support

0 0.90 0.96 0.93 68

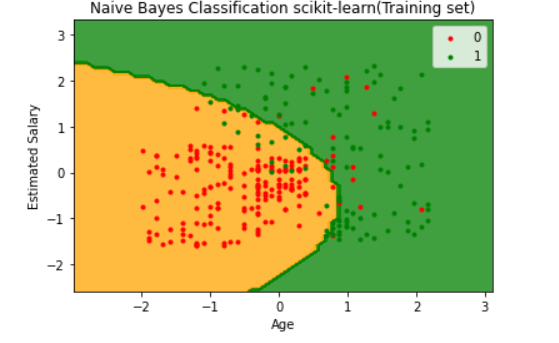
1 0.89 0.78 0.83 32

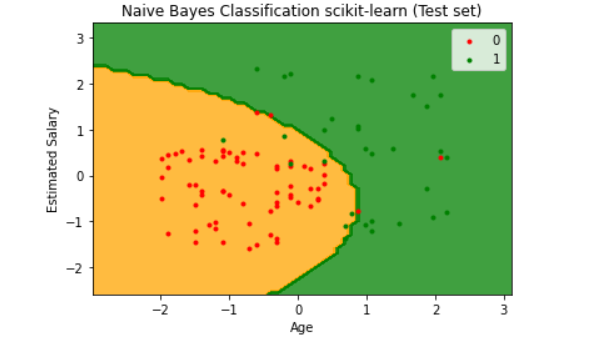
accuracy 0.90 100

macro avg 0.90 0.87 0.88 100

weighted avg 0.90 0.90 0.90 100

**VISUALIZATION OF RESULTS:**





**PREDICTING NEW VALUES:**

New values

Age

22

Estimated Salary

3445

Predicting the result:

: [1]

**CONCLUSION:**

From model 1, the accuracy score of kNN is 93% ,which is pretty good. Comparing with scikit – learn, scratch code gives less accuracy . The confusion matrix shows that TP(True Positive) is 64 , it predicts positive and it’s true. From model 2, the accuracy of the Naive bayes classifier is 90% accuracy is good. The prediction of new purchase by giving age and salary and it predicts 1.

The obtained results showed the use value of both machine learning models. The k-nearest neighbour model showed slightly better performance than the Naive bayes model, but definitely, both models have shown that they can be very successful in solving classification problems.

*Finally, The assumption(hypothesis h1) becomes true.* The number of males who purchased the product are more than number of females who purchased the product.