Group I
Predict Term Subscription

TEAM MEMBERS:

Vyshnavi Reddy Mungi
Maniteja Vemunoori
Akshitha Reddy Poreddy
Xuan Mai Tran
Haji Mastan Vali Shaik

TABLE OF CONTENTS

TABLE OF CONTENTS

INTRODUCTION - CONTEXT ANALYSIS

ML predictions for bank data

ML to predict term deposit, based on data

Analyzing past call to know about subscription

Machine Learning to analyze bank approval data

INTRODUCTION - CONTEXT ANALYSIS

Banking Industry Context

- The importance of the banking system is for business activities and citizen life.
- The health of banking system will reflect truly the health of economy.
- Especially there’re news of banks shut down recently.

Team Role

- We as a data scientist team cooperate with Marketing department to figure out the valuable insights from the marketing campaign to support for the health of our bank.

Business Context

- Stakeholders want to gain insight from the recent marketing campaign to make business decisions.

INTRODUCTION - CONTEXT ANALYSIS

Business Question

- Analyzing the marketing campaign to predict customer subscription of a term deposit.

Dataset Info

- Direct marketing campaigns of a Portuguese banking institution
- Collected May 2008 - Nov 2010.
- Includes ~ 40,000 records and 20 attributes which consist 3 types of info:
  - bank client data
  - related data of the marketing campaign
  - social & economic context

Implementations of ML

- **Data Visualization**: to learn about the dataset
- **Choose Important Features**: Feature Selection
- **Testing different algorithms** to predict the effect of important features on classification of customer subscription's decision.

Term subscription made easy with our prediction algorithm

Insights - Bank Term Subscription prediction

Bank history analysis using ML

Predict term subscription with machine learning models

Feature selection in term approval predictions

Feature selection methods for predictive modeling

Data preprocessing techniques for predictions

Exploring different classification models for bank applications

Model selection for predicting subscription approvals

Importance of model interpretability for banks and financial institutions.

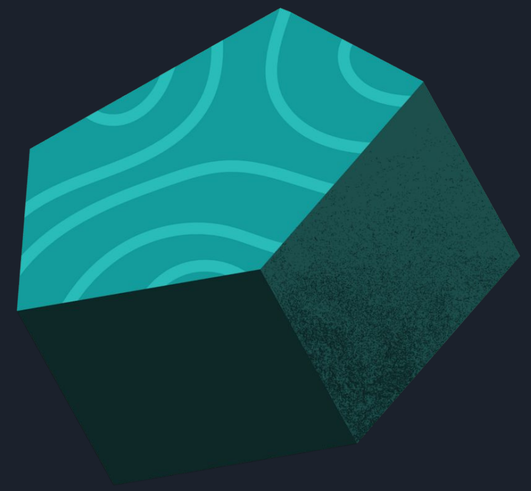
Data Description

Building a predictive model that, given the above features, can accurately predict whether a term deposit will be subscribed or not. This involves data preprocessing, exploratory data analysis, model training, and evaluation. The dataset we are using captures various financial and personal details that could be used for assessing an individual's eligibility for a term deposit and for financial analysis purposes. Depending on the specific goals of the analysis, we can use these attributes to gain insights into the financial profile of individuals and build predictive models for term subscription approval process. This type of predictive modeling is common in the financial sector to automate and streamline the term subscription approval process.

Step 1 - Data Collection

Data attributes – name of columns

1 - age (numeric)
2 - job : type of job (categorical: 'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown')
3 - marital : marital status (categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed)
4 - education (categorical: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unknown')
5 - default: has credit in default? (categorical: 'no','yes','unknown')
6 - housing: has housing loan? (categorical: 'no','yes','unknown')
7 - loan: has personal loan? (categorical: 'no','yes','unknown')



Step 1 - Data Collection

Data attributes – name of columns

Bank client data

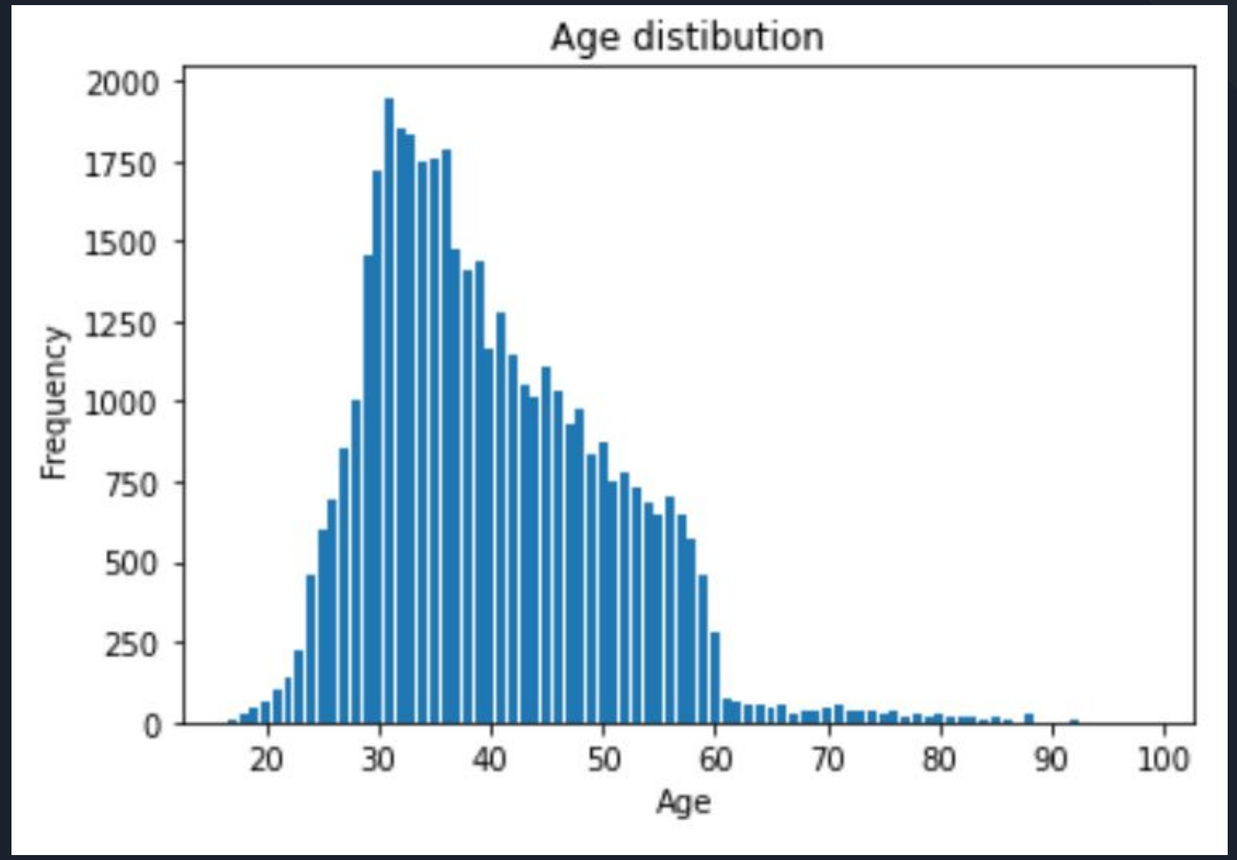
related with the last contact of the current campaign:
8 - contact: contact communication type (categorical: 'cellular','telephone')
9 - month: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')
10 - day_of_week: last contact day of the week (categorical: 'mon','tue','wed','thu','fri')
11 - duration: last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y='no'). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.
#other attributes:
12 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)
13 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
14 - previous: number of contacts performed before this campaign and for this client (numeric)
15 - poutcome: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')

Step 2 - Data Visualization

Bar Chart - “Age” Distribution

Explanation

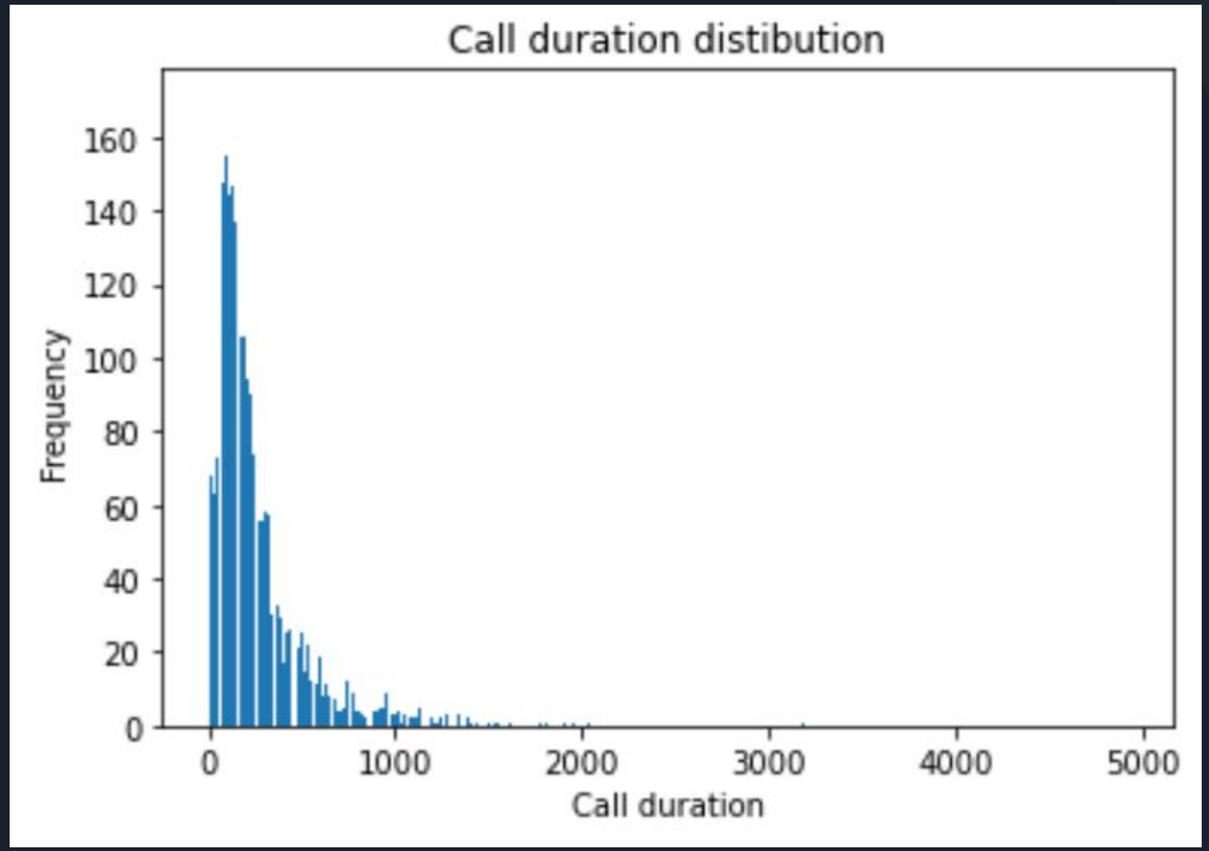
- From the bar chart of "age" distribution, most of surveyed customers have age range from 25 to 60 years old. We can see that most of them are still in the labor age.
- The mode is in the age range of 30 to 35 years old. This is the age that when people often have some years of working experiences and may leverage their career to senior or manager position.
- This bar chart is also a positively skewed (or right-skewed) distribution where most values are sitting around the left tail while the right tail of is longer with some outliers.



Bar Chart - “Call Duration” Distribution

Explanation

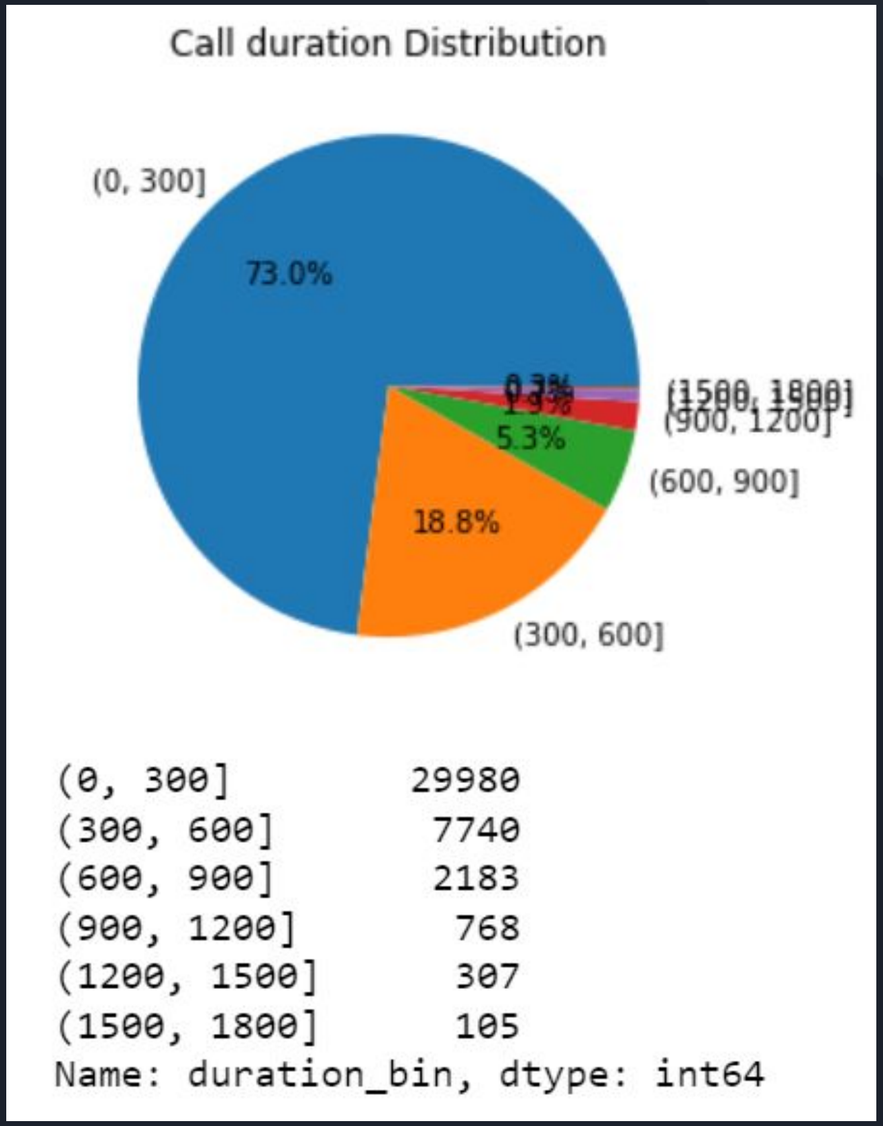
- From the bar chart of "call duration" feature, we can see that most data ranges from the value of 0 to 600.
- The mode is around the value of 100.
- This bar chart also represents for a positively skewed (or right-skewed) distribution where most values are sitting around the left tail while the right tail of is longer with some outliers over the value of 600.



Pie Chart - “Call Duration” Distribution

Explanation

- This pie chart of "call duration" together with the bar chart above helps us to understand more clearly about this feature.
- From the pie chart, we can see that about 70% having the call duration in range from 0 to 300.
- And, nearly 20% of the call duration is in range from 300 to 600.
- About 10% of call duration has the outlier values above 600.
- Hence, this pie chart of "call duration" matches with its bar chart.

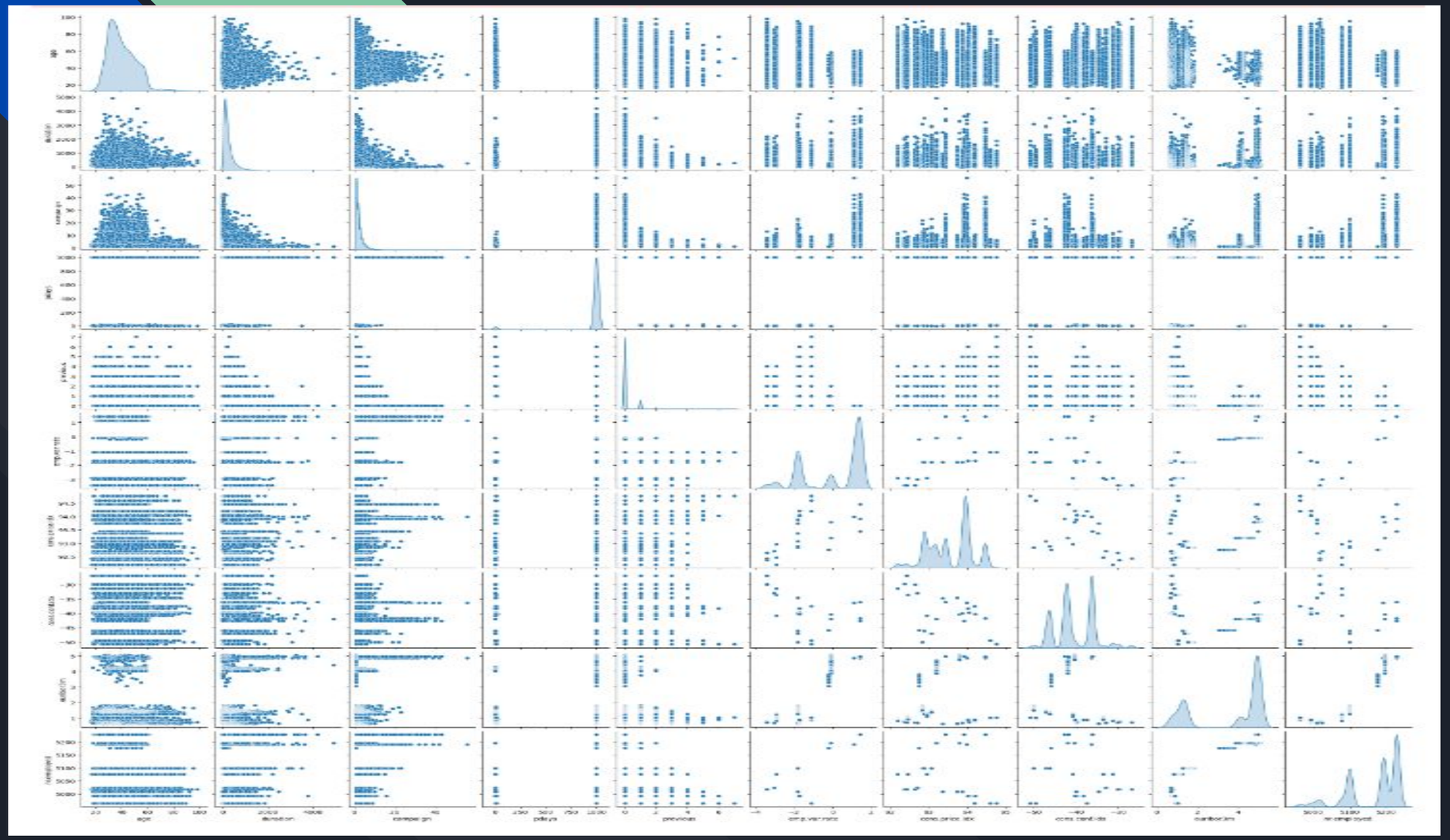



```
(0, 300]        29980
(300, 600]       7740
(600, 900]       2183
(900, 1200]       768
(1200, 1500]      307
(1500, 1800]      105
Name: duration_bin, dtype: int64
```

Step 3 - Feature Selection

Univariate	*RFE*	*Feature importance*
age	job	Duration
job	marital	age
marital	education	month
education	default	job
default	housing	day_of_week
contact	loan	pdays
month	contact	campaign
duration	month	education
campaign	day_of_week	poutcome
pdays	campaign	housing
previous	previous	marital
poutcome	poutcome	previous

Scatter Plot



Interpretation of results

By combining the results from all these methods, we can identify the top numerical features that are most important for predicting the target variable in the bank marketing campaign dataset.

Common features selected by all methods:

duration
age
default
campaign
poutcome
job
education
housing
loan
marital

STEP 4: REGRESSION

Select Attributes - Build Logistic Regression Model

- duration: last contact duration, in seconds (numeric)
- age: (numeric)
- y - target variable: has the client subscribed a term deposit? (binary: 'yes','no')

```
X=data[['age','duration']]
y=data['y']


# creating a logistic regression model
logr = linear_model.LogisticRegression()
logr.fit(X,y)
```

```
▾ LogisticRegression
LogisticRegression()
```

```
# probability of the model
prob = logr.predict_proba(X)
prob
```

```
array([[0.89704803, 0.10295197],
       [0.92852524, 0.07147476],
       [0.92465099, 0.07534901],
       ...,
       [0.91900113, 0.08099887],
       [0.83704307, 0.16295693],
       [0.88520597, 0.11479403]])
```

Confusion Matrix

ind confusion matrix

```
predicting the model
red =  logr.predict(X)
red
```

```
rray([0, 0, 0, ..., 0, 0, 0], dtype=int64)
```

```
 score for logistic regression model
core = logr.score(X,y)
core
```

```
.8931727687676022
```

```
rom sklearn.metrics import classification_report, confusion_matrix
onfM= confusion_matrix(y,pred)
onfM
```

```
rray([[36021,   527],
      [ 3873,   767]], dtype=int64)
```

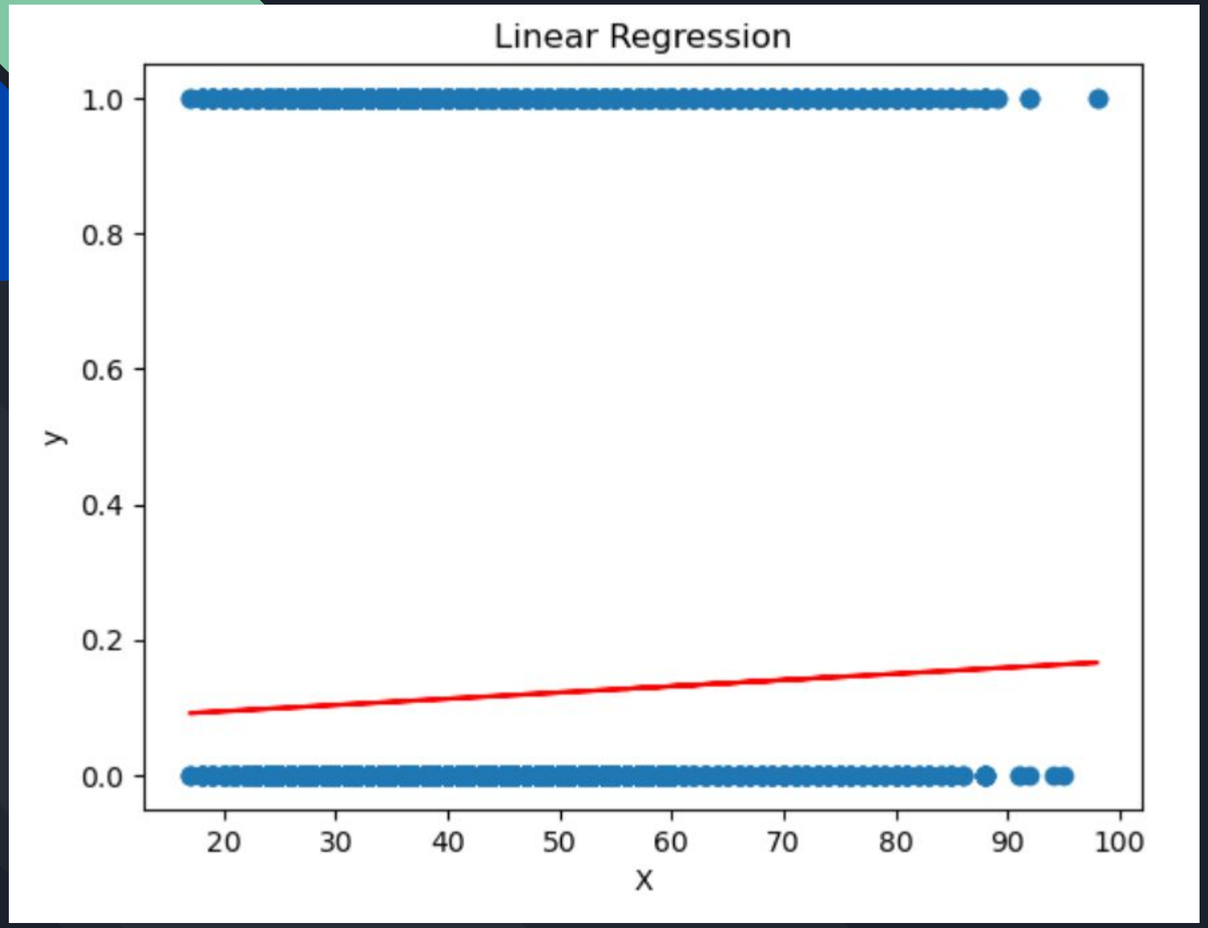
Plotting Linear Regression - Code

Plot the linear regression

```
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

# Fit the linear regression model
model = LinearRegression()

model.fit(X, y)

m = model.coef_[0]
c = model.intercept_

# Check if the line is in the form y = mx + c
print(f"The equation of the line is y = {m}x + {c}.")

# Plot using matplotlib
plt.scatter(X, y)
plt.plot(X, m * X + c, color='red')

# Plot the data and the fitted line
# plt.scatter(X, y)
# plt.plot(X, model.predict(X), color='red')

# Add labels and title
plt.xlabel('X')
plt.ylabel('y')
plt.title('Linear Regression')

# Show the plot
plt.show()
```

```
The equation of the line is y = 0.0009222783974769146x + 0.07574084482608853.
```

Plotting Linear Regression - Visualization



Interpretation

- The line graph of linear regression show a positive relationship between "Age" and "y-Deposit Subscription".
- It means that when age increases, there is a slight chance that people will subscribe to the deposit term.
- It may be because there are a saving account growing up with ages.
- Thus, the marketing manager may consider to focus the campaign on aged people.

STEP 5: DECISION TREE
Plotting Decision Tree - Code

Build a Decision Tree

In [4]:
```
from sklearn.tree import DecisionTreeClassifier, plot_tree
import pandas as pd
import matplotlib.pyplot as plt
```

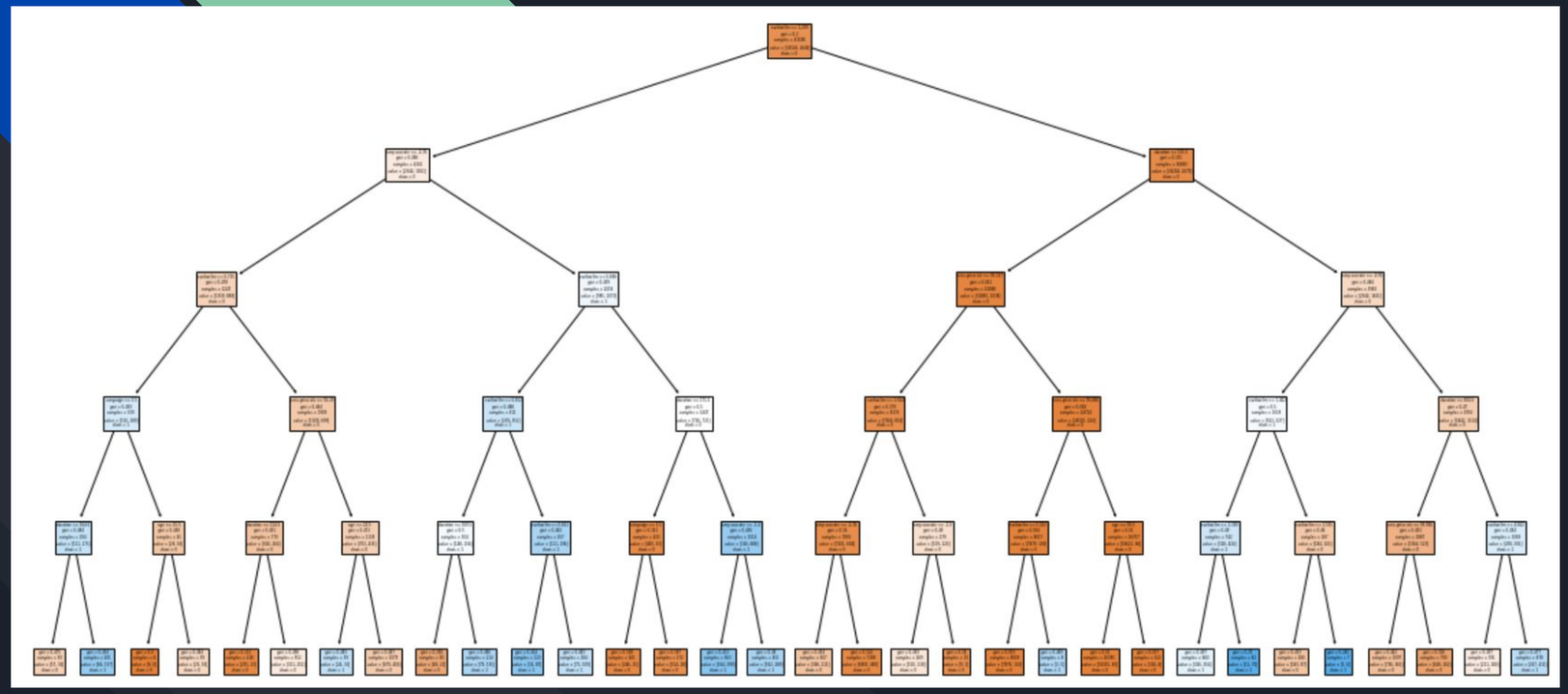
In [5]:
```
# Initialize Decision Tree Classifier
clf = DecisionTreeClassifier(max_depth=5, min_samples_split=10, min_samples_leaf=5, max_features='auto')

# Train the model
clf.fit(X, y)
```

Out[5]:
```
DecisionTreeClassifier(max_depth=5, max_features='auto', min_samples_leaf=5,
                       min_samples_split=10)
```

Plot the Decision Tree

In [6]:
```
# Visualize the decision tree
plt.figure(figsize=(20,10))
plot_tree(clf, feature_names=X.columns, class_names=[str(c) for c in sorted(y.unique())], filled=True)
plt.show()
```

Plotting Decision Tree - Visualization



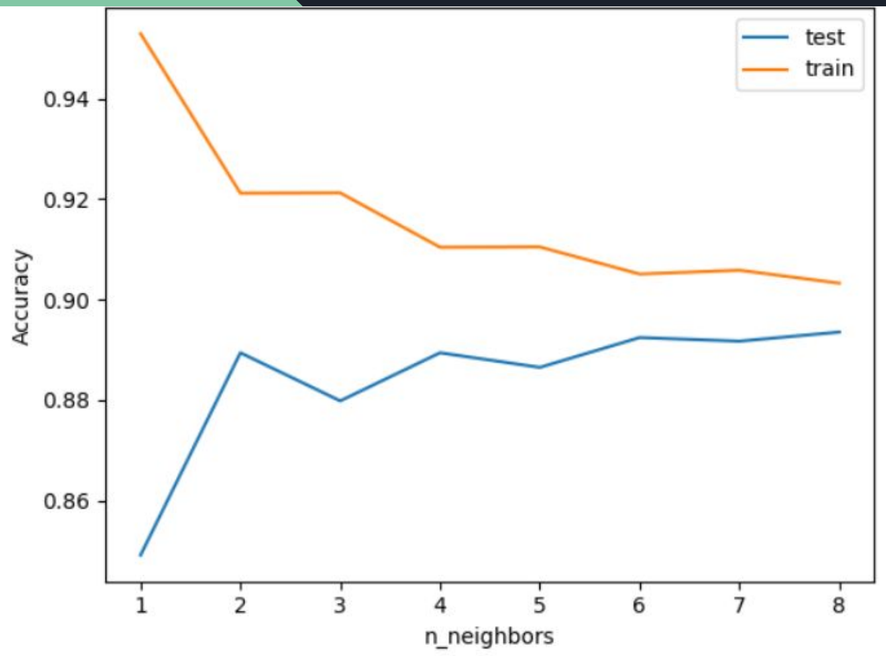
Interpretation from decision tree

Observation

Based on the dataframe of feature importance of Decision Tree, we can see the "duration" is the most important feature which is placed at the top of the Decision Tree.
Following that, "age" and "poutcome" are the next importance features placed at the 2nd top of the Decision Tree.

Step 6 :KNN & NB

Elbow graph and K



What is the best number of classes

According to the above graph, we get an optimal value of k when n_neighbors=2, there is sharp point as elbow when n_neighbors=2.

Step 6 :KNN & NB

```
from sklearn.neighbors import KNeighborsClassifier
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap
import numpy as np

n_ne = 2

X=data[['age','duration']].values
y=y
# considering 4 because, we got 4 categories
h=2
# define the model, fit the model
clf = KNeighborsClassifier(n_ne, weights='distance')
clf.fit(X,y)
```

```
KNeighborsClassifier(n_neighbors=2, weights='distance')
```

```
# make a grid with h=4, make the frame
x_min, x_max=X[:,0].min()-1,X[:,0].max()
y_min, y_max=X[:,1].min()-1,X[:,1].max()
xx, yy=np.meshgrid(np.arange(x_min,x_max,h),np.arange(y_min, y_max,h))

# predict unknown values using KNN
Z = clf.predict(np.c_[xx.ravel(),yy.ravel()])
Z = Z.reshape(xx.shape)
```

```
# define colors for classes and items
c_light = ListedColormap(['#FFAAAA','#AAFFAA','#00AAFF'])
c_dark = ListedColormap(['#FF0000','#00FF00','#0000FF'])
```

Find the accuracy

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test=train_test_split(X,y, test_size=0.2)

train_accuracy=clf.score(X_train, y_train)
test_accuracy=clf.score(X_test, y_test)
print("Train accuracy : ",train_accuracy)
print("Test accuracy : ",test_accuracy)
```

```
from sklearn.naive_bayes import GaussianNB
clf = GaussianNB()
clf.fit(X,y)
```

```
GaussianNB()
```

predict the unknown random

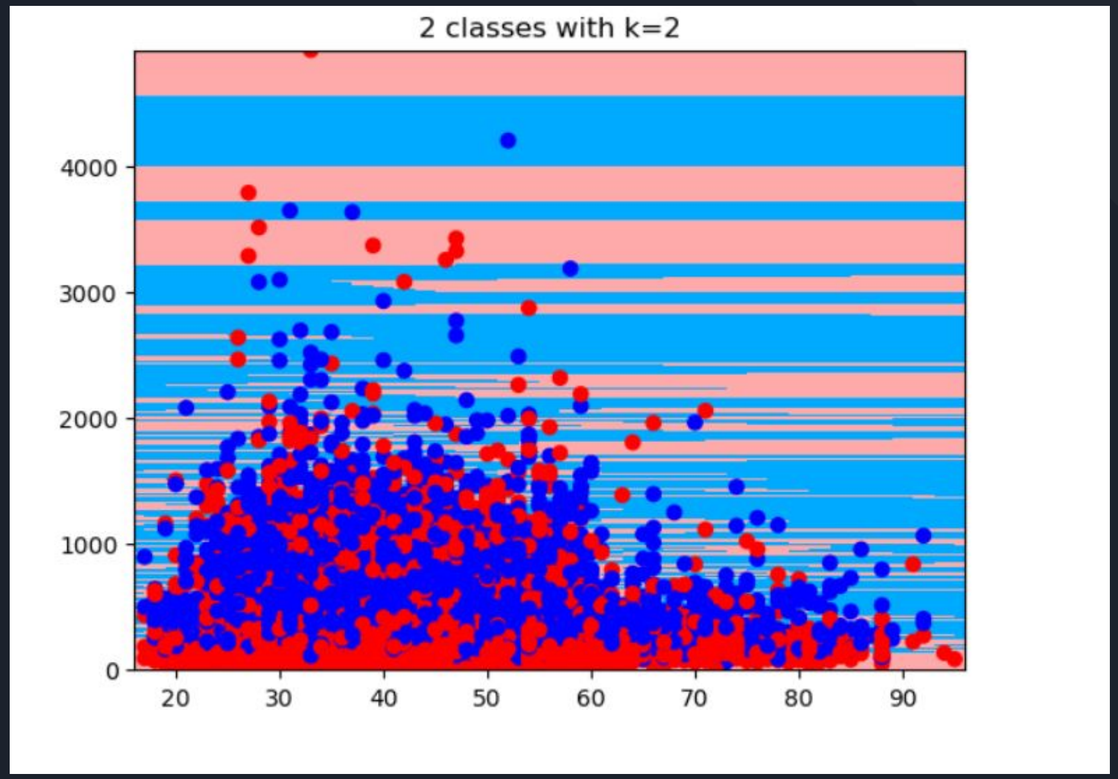
```
xmin = X.min()
print(xmin)
xmax = X.max()
print(xmax)
```

```
0
4918
```

```
#rng create 2000 random variables between 0 to 1 having 2 columns i.e 1000 rows each column
#range of X is -13 and 5, the predicted value should be between -13 and 5
import numpy as np
xmin = X.min()
print(xmin)
xmax = X.max()
print(xmax)
rng=np.random.RandomState(0)
# This array is then scaled and shifted using [-6, -13] + [11, 18], which effectively scales and
#shifts the random numbers to fall within the range specified.
Xnew = [-6,0] + [11,18] * rng.rand(2000,2)
ynew = clf.predict(Xnew)
Xnew
```

KNN & NB

KNN graph, accuracy and interpretation



Accuracy:

Train accuracy : 0.9582701062215478
Test accuracy : 0.9579995144452537

Interpretation

Based on the above KNN, we got that the young people between the age 20-30 are being actively targetted for the subscription and the proportion of accepting the subscriptions is also distributed evenly.

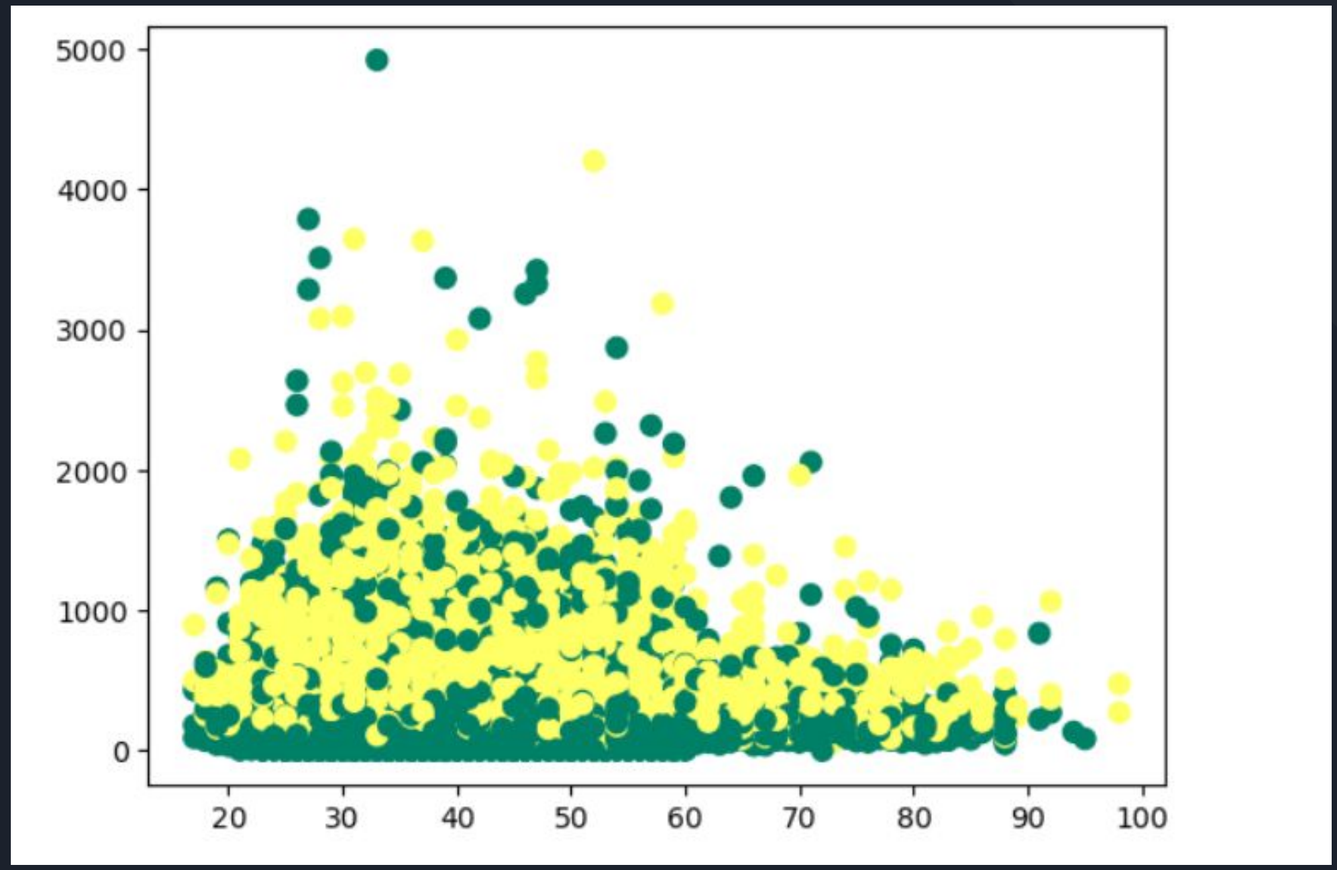
KNN & NB

NB graph, accuracy and interpretation

Accuracy:

Train accuracy : 0.892443095599393

Test accuracy : 0.89414906530711 34

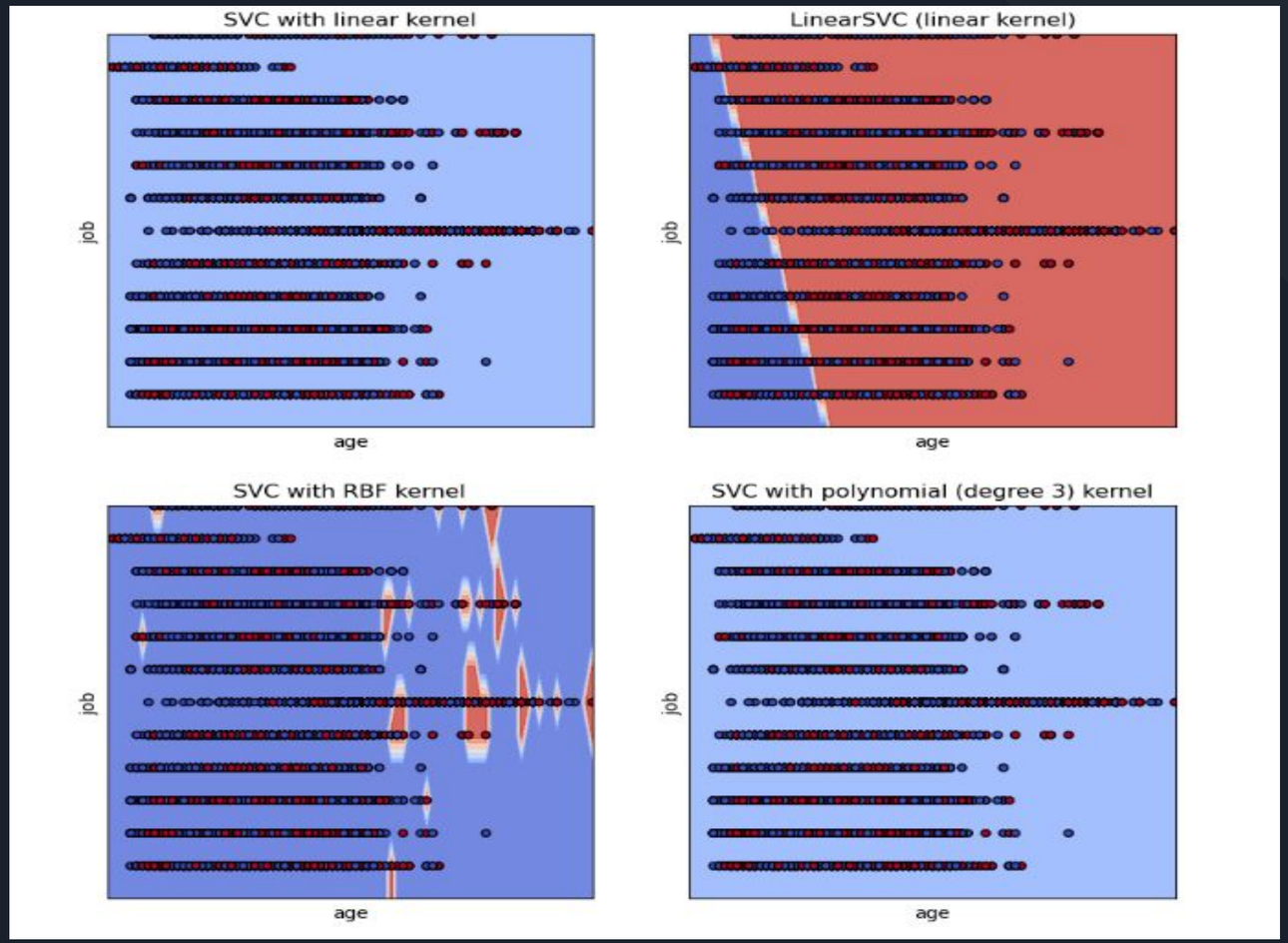


Interpretation

The accuracy for both KNN and NB graph is different. It's essential to assess the accuracy of the model on both the training and test sets to gauge its generalization performance and potential overfitting.

Analyzing the confusion matrix to understand where the model is making errors (e.g., false positives or false negatives) and considering the implications of these errors for the campaign strategy

Step 7 : SVM



Accuracy and interpretation

Accuracy:

Accuracy for SVC(C=1, kernel='linear') is : 88.76

Accuracy for LinearSVC(C=1) is : 88.76

Accuracy for SVC(C=1, gamma=0.7) is : 88.93744436351867

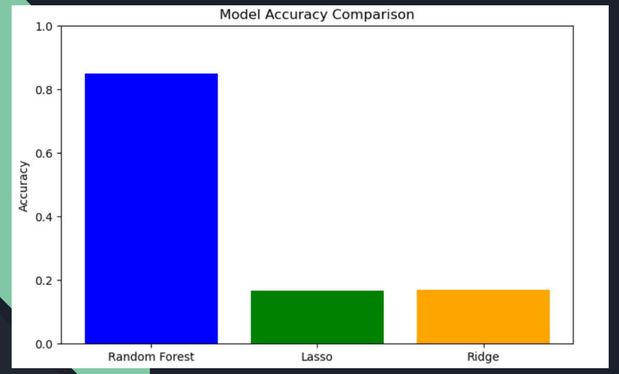
Accuracy for SVC(C=1, kernel='poly') is:88.76750020231448

Interpretation

SVM kernels are used to transform the data into a higher dimensional space where it becomes easier to classify the data. Different kernels often provide different data transformations.
The accuracy of 4 kernels are quite similar, maybe that's the reason why 4 SVM Graphs look like the same in data distribution.
In general, the accuracy of 4 kernels are quite high with over 80%. It shows the kernel functions classify the data well.
As we see, the RBF kernel has the highest accuracy score of 88.93%. It means that RBF kernel are able to classify the data better than other kernels in this case.

Step 8 : Random Forest & Regularization

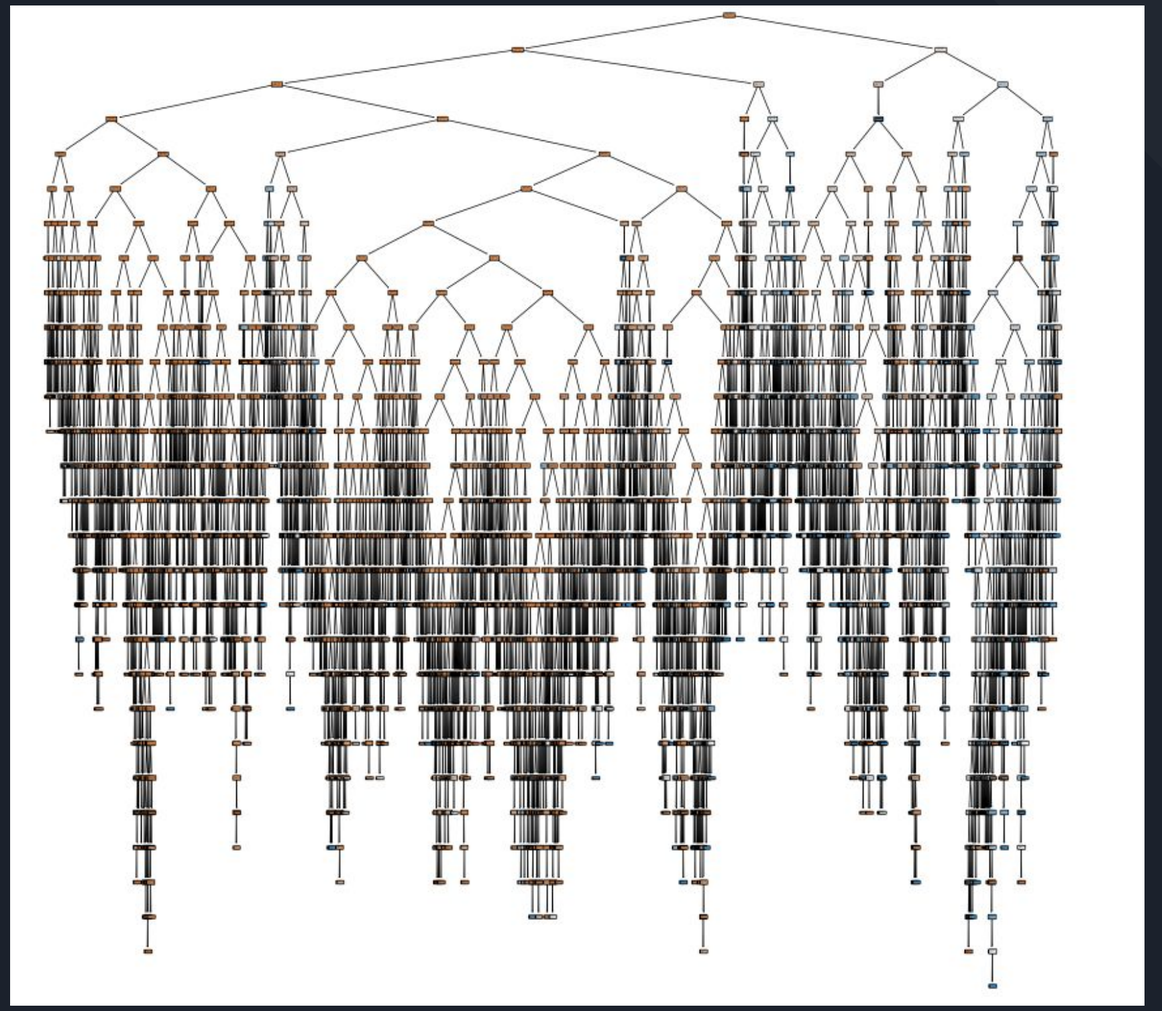
Accuracy and interpretation



Accuracy for RF : 0.88

Interpretation

The bar chart provides a visual comparison of how well each model performs in terms of accuracy. Higher bars indicate higher accuracy, implying better model performance. By comparing the heights of the bars, one can quickly determine which model performs better on the test dataset. This visualization helps in selecting the most suitable model for the task at hand or in identifying if further optimization is required.



Random Forest , Lasso and Ridge Regressor Code

In [9]:
```
# split data to tran and test
X_train, X_test, y_train, y_test=train_test_split(X,y, test_size=0.3 )
# define the model, fit the model
clf=RandomForestClassifier(n_estimators=50)
clf.fit(X_train, y_train)
RandomForestClassifier(n_estimators=50)
y_pred=clf.predict(X_test)

from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
CM=confusion_matrix(y_pred,y_test)
print(CM)
AS=accuracy_score(y_pred,y_test)
print(AS)
CR=classification_report(y_pred,y_test)
print(CR)
```

In [20]:
```
from sklearn.linear_model import Lasso

lasso=Lasso(alpha=1)
lasso.fit(X_train,y_train)

Pred_Lr=lasso.predict(X_test)

Err=np.mean((Pred_Lr-y_test)**2)
Err
```

Out[20]: 0.084239248354793

In [21]:
```
from sklearn.linear_model import Ridge

R=Ridge(alpha=1)
R.fit(X_train,y_train)

Pred_R=R.predict(X_test)

Err=np.mean((Pred_R-y_test)**2)
Err
```

Out[21]: 0.08423581910385497

CONCLUSION – EXTRACTED INSIGHTS

- KNN is the algorithm producing the highest accuracy to confirm the positive relationship between “***Age***” and **“*Term Subscription*”** with the highest potentials in the age range of *30 to 35* yr old.
- Besides, "***duration***" **f**eature is the most important feature affecting to whether a customer wants to subscribe a term deposit or not. Thus, a marketer can consider a marketing campaign focusing on customers having longer last contact duration.
- **“*euribor3m*” and “*cons.price.idx*"** reflect truly the reality that economic indicators affect significantly the customers’ subscription decision.

CONCLUSION – BUSINESS SUGGESTIONS

- Focus the marketing campaigns on the target customers who are 30-35 yr old.
- For personalized marketing strategy, focus on customers having longer last contact duration that shows their high interests on our services.
- Promote saving or protection / treasury management plans for customers during the tough economic situations.
- Boosting more marketing for reinvestment options on banking services like diversifying term deposits, and related services such as stock/bond investment through online banking.

REFERENCES

[1] Moro,S., Rita,P., and Cortez,P.. (2012). Bank Marketing. UCI Machine Learning Repository. https://doi.org/10.24432/C5K306.

[2] “Bank Marketing - dataset by uci,” data.world. https://data.world/uci/bank-marketing (accessed Feb. 06, 2024).

[3] Hou, Sipu, et al. "Applying Machine Learning to the Development of Prediction Models for Bank Deposit Subscription."International Journal of Business Analytics, vol. 9, no. 1, Jan. 2022, pp. 1-12, doi:10.4018/IJBAN.288514

CONTRIBUTION

Vyshnavi Reddy Mungi - Focused on the initial stages of data processing, including data collection , visualization and feature selection. Ensured that we had a comprehensive dataset with well-defined column names and provided insightful visualizations to understand the data better. Additionally, analyzed the regression, decision tree, KNN & NB techniques, SVM, and Random Forest & Regularization techniques.

Poreddy Akshitha: Worked on how different algorithms like KNN, NB, Decision Trees and Random Forest works on dataset. Hyperparameter tuning, finding accuracy and based on graphs from different algorithms interpreted , drawn a conclusion. Visualization, transformation and normalization of data has been done on dataset.

Maniteja Vemunoori: Performed Data preprocessing steps like standardization, normalization and making the data into same scale. Different features has been selected based on different feature selection methods. WOrked on Algorithms for dataset.

Xuan Mai Tran presentation design, implementation and interpretation of data visualization, logistic regression, linear regression, implementation and interpretation of Decision Tree, contributions to KNN & NB graphs, contribution and interpretation of SVM kernels, making and presentation on Intro and Conclusion parts.

Haji Mastan Vali Shaik: Performed data visualization techniques like Bar chart for age and call distributions , Pie chart , Decision Tree, SVM. Analyzed the best model by comparing the accuracy of the various models.

Thank You