S. B. JAIN INSTITUTE OF TECHNOLOGY, MANAGEMENT & RESEARCH, NAGPUR.

Practical No. 4

Aim: Apply and implement Random Forest Algorithm in Machine Learning.

Name of Student : **Shrutika Pradeep Bagdi**

Roll No. : **CS22130**

Semester/Year : **6th / 3rd**

Academic Session : **2024-2025**

Date of Performance :

Date of Submission :

OBJECTIVE/EXPECTED LEARNING OUTCOME:

The objectives and expected learning outcome of this practical are:

- It's more accurate than the decision tree algorithm.
- It provides an effective way of handling missing data.
- It can produce a reasonable prediction without hyper-parameter tuning.
- It solves the issue of overfitting in decision trees.
- In every random forest tree, a subset of features is selected randomly at the node's splitting point.

THEORY:

A random forest is a supervised machine learning algorithm that is constructed from decision tree algorithms. This algorithm is applied in various industries such as banking and e-commerce to predict behavior and outcomes.This article provides an overview of the random forest algorithm and how it works. The article will present the algorithm's features and how it is employed in real-life applications. It also points out the advantages and disadvantages of this algorithm.

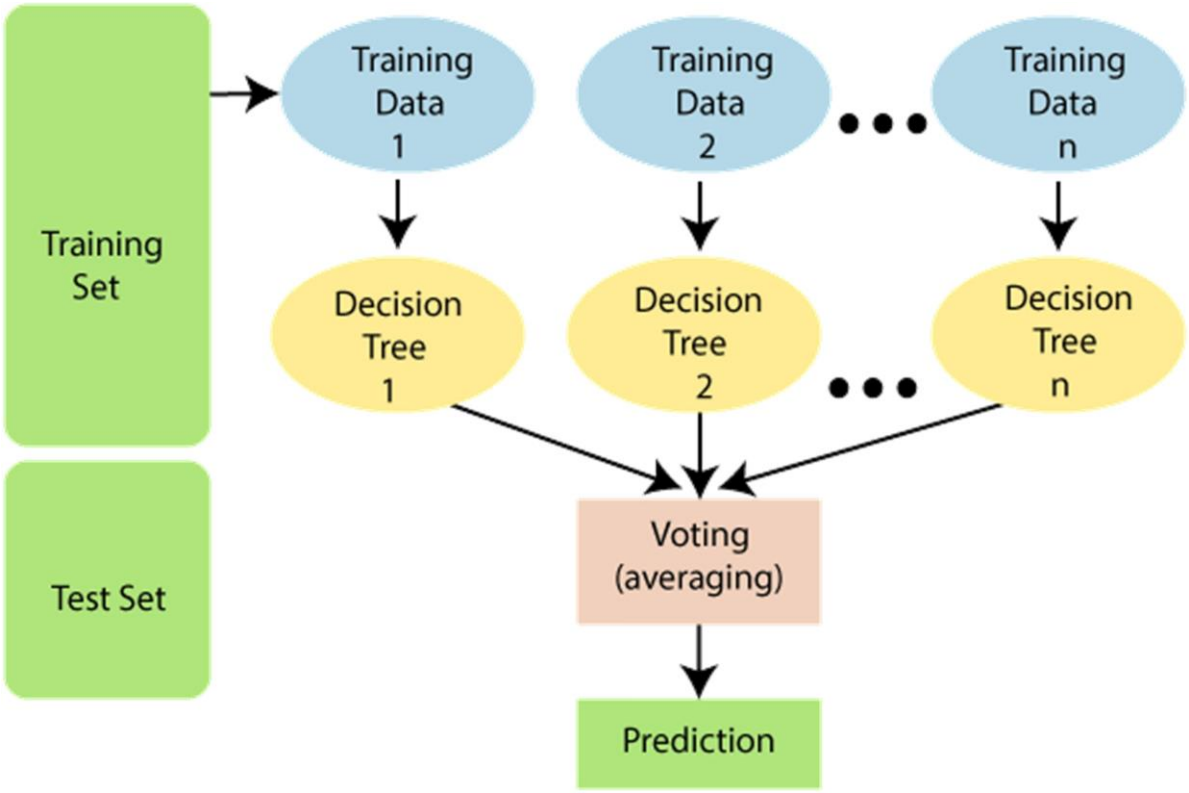
What is Random forest Algorithm?

A random forest is a machine learning technique that's used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems. A random forest algorithm consists of many decision trees. The 'forest' generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms.

Why do we need a Random forest Algorithm?

Below are some points that explain why we should use the Random Forest algorithm:

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.



Algorithmic steps for Random Forest clustering

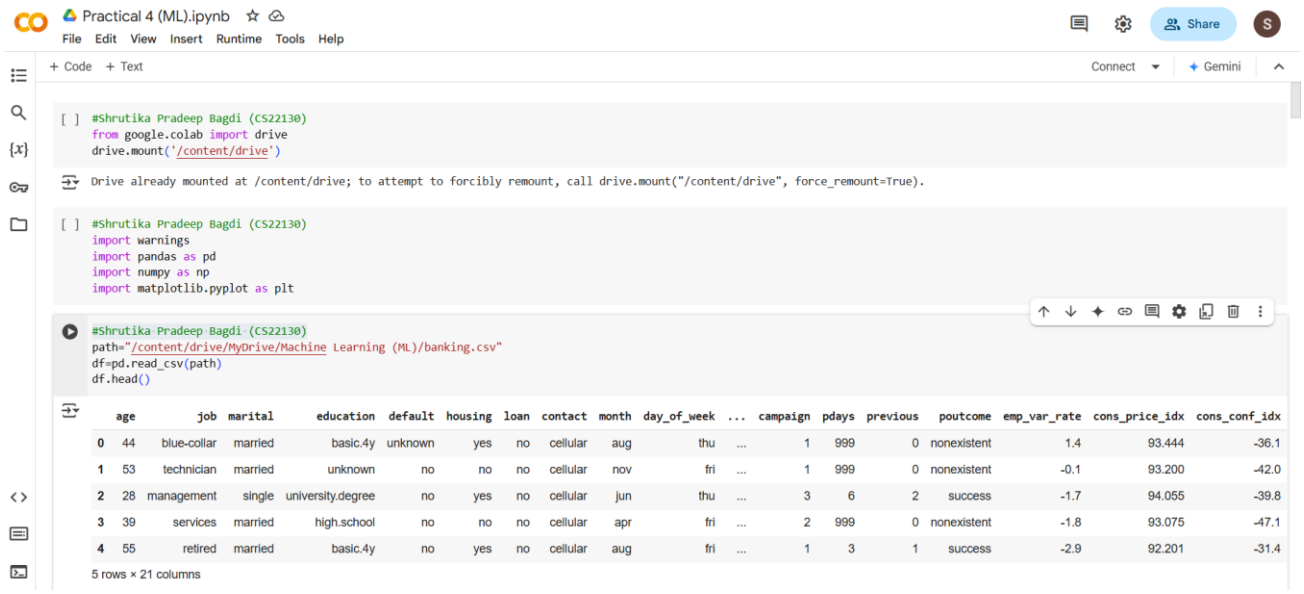
Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase. The Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

Step-1: Select random K data points from the training set.
Step-2: Build the decision trees associated with the selected data points (Subsets).
Step-3: Choose the number N for decision trees that you want to build.
Step-4: Repeat Step 1 & 2.
Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

PROGRAM CODE:

OUTPUT (SCREENSHOT):



Practical 4 (ML).ipynb

File Edit View Insert Runtime Tools Help

Share

+ Code + Text

Connect Gemini

```
[ ] #Shrutika Pradeep Bagdi (CS22130)
    from google.colab import drive
    drive.mount('/content/drive')
```

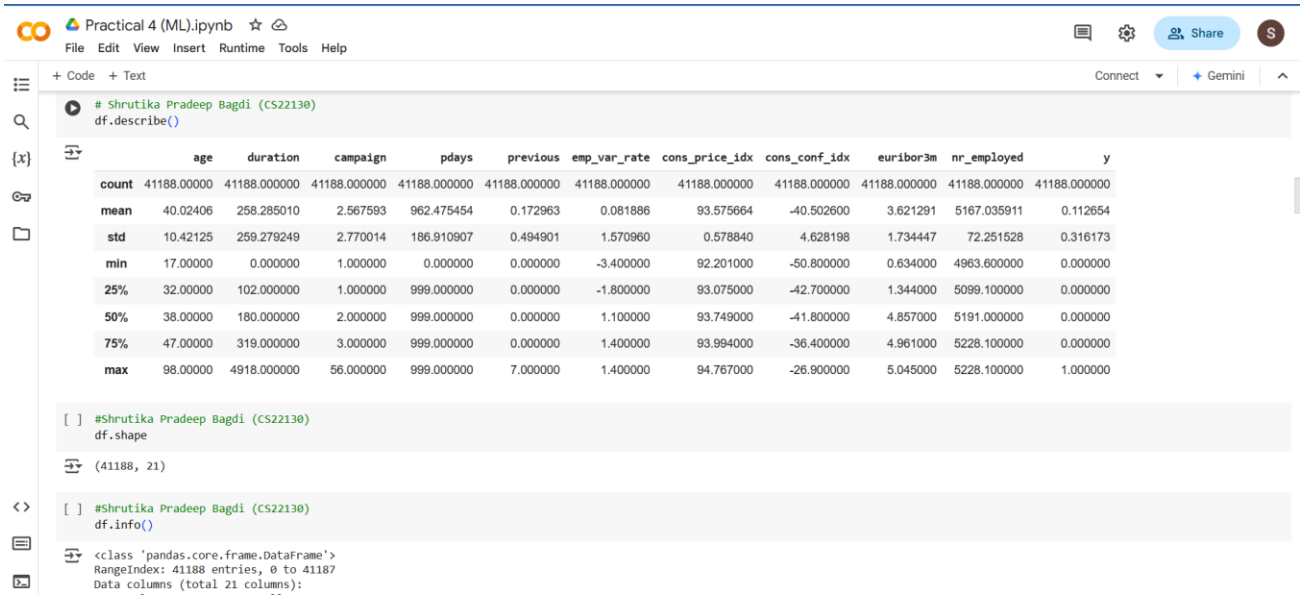
```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
[ ] #Shrutika Pradeep Bagdi (CS22130)
    import warnings
    import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
```

```
#Shrutika Pradeep Bagdi (CS22130)
path="/content/drive/MyDrive/Machine Learning (ML)/banking.csv"
df=pd.read_csv(path)
df.head()
```

	age	job	marital	education	default	housing	loan	contact	month	day_of_week	...	campaign	pdays	previous	poutcome	emp_var_rate	cons_price_idx	cons_conf_idx
0	44	blue-collar	married	basic.4y	unknown	yes	no	cellular	aug	thu	...	1	999	0	nonexistent	1.4	93.444	-36.1
1	53	technician	married	unknown	no	no	no	cellular	nov	fri	...	1	999	0	nonexistent	-0.1	93.200	-42.0
2	28	management	single	university.degree	no	yes	no	cellular	jun	thu	...	3	6	2	success	-1.7	94.055	-39.8
3	39	services	married	high.school	no	no	no	cellular	apr	fri	...	2	999	0	nonexistent	-1.8	93.075	-47.1
4	55	retired	married	basic.4y	no	yes	no	cellular	aug	fri	...	1	3	1	success	-2.9	92.201	-31.4

5 rows × 21 columns



Practical 4 (ML).ipynb

File Edit View Insert Runtime Tools Help

Share

+ Code + Text

Connect Gemini

```
# Shrutika Pradeep Bagdi (CS22130)
df.describe()
```

	age	duration	campaign	pdays	previous	emp_var_rate	cons_price_idx	cons_conf_idx	euribor3m	nr_employed	y
count	41188.00000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000
mean	40.02406	258.285010	2.567593	962.475454	0.172963	0.081886	93.575664	-40.502600	3.621291	5167.035911	0.112654
std	10.42125	259.279249	2.770014	186.910907	0.494901	1.570960	0.578840	4.628198	1.734447	72.251528	0.316173
min	17.00000	0.000000	1.000000	0.000000	0.000000	-3.400000	92.201000	-50.800000	0.634000	4963.600000	0.000000
25%	32.00000	102.000000	1.000000	999.000000	0.000000	-1.800000	93.075000	-42.700000	1.344000	5099.100000	0.000000
50%	38.00000	180.000000	2.000000	999.000000	0.000000	1.100000	93.749000	-41.800000	4.857000	5191.000000	0.000000
75%	47.00000	319.000000	3.000000	999.000000	0.000000	1.400000	93.994000	-36.400000	4.961000	5228.100000	0.000000
max	98.00000	4918.000000	56.000000	999.000000	7.000000	1.400000	94.767000	-26.900000	5.045000	5228.100000	1.000000

```
[ ] #Shrutika Pradeep Bagdi (CS22130)
    df.shape
```

```
(41188, 21)
```

```
[ ] #Shrutika Pradeep Bagdi (CS22130)
    df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41188 entries, 0 to 41187
Data columns (total 21 columns):
```

Practical 4 (ML).ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

```
#Shrutika Pradeep Bagdi (CS22130)
X=df.iloc[:,:-1]
print(X)
```

```
       age          job  marital            education  default housing loan  \
0       44  blue-collar  married             basic.4y  unknown     yes   no
1       53   technician  married              unknown       no      no   no
2       28   management   single    university.degree       no     yes   no
3       39     services  married          high.school       no      no   no
4       55      retired  married             basic.4y       no     yes   no
...    ...          ...      ...                  ...      ...     ...  ...
41183   59      retired  married          high.school  unknown      no  yes
41184   31    housemaid  married             basic.4y  unknown      no   no
41185   42       admin.   single    university.degree  unknown     yes  yes
41186   48   technician  married  professional.course       no      no  yes
41187   25      student   single          high.school       no      no   no

          contact month day_of_week  duration  campaign  pdays  previous  \
0       cellular   aug         thu       210         1    999         0
1       cellular   nov         fri       138         1    999         0
2       cellular   jun         thu       339         3      6         2
3       cellular   apr         fri       185         2    999         0
4       cellular   aug         fri       137         1      3         1
...          ...   ...         ...       ...       ...    ...       ...
41183  telephone   jun         thu       222         1    999         0
41184  telephone   may         thu       196         2    999         0
41185  telephone   may         wed        62         3    999         0
41186  telephone   oct         tue       200         2    999         0
41187  telephone   may         fri       112         4    999         0

           poutcome  emp_var_rate  cons_price_idx  cons_conf_idx  euribor3m  \
0       nonexistent           1.4          93.444          -36.1      4.963
1       nonexistent          -0.1          93.200          -42.0      4.021
```

Practical 4 (ML).ipynb

File Edit View Insert Runtime Tools Help

Share

+ Code + Text

Connect Gemini

```
#Shrutika Pradeep Bagdi (CS22130)
y=df.iloc[:,-1]
print(y)
```

```
0        0
1        0
2        1
3        0
4        1
        ..
41183    0
41184    0
41185    0
41186    0
41187    0
Name: y, Length: 41188, dtype: int64
```

```
#Shrutika Pradeep Bagdi (CS22130)
X = df.drop(columns=['y'])
y = df['y']
```

```
X.head()
```

	age	job	marital	education	default	housing	loan	contact	month	day_of_week	duration	campaign	pdays	previous	poutcome	emp_var_rate	cons_price_idx	cons_conf
0	44	blue-collar	married	basic.4y	unknown	yes	no	cellular	aug	thu	210	1	999	0	nonexistent	1.4	93.444	
1	53	technician	married	unknown	no	no	no	cellular	nov	fri	138	1	999	0	nonexistent	-0.1	93.200	
2	28	management	single	university.degree	no	yes	no	cellular	jun	thu	339	3	6	2	success	-1.7	94.055	
3	39	services	married	high.school	no	no	no	cellular	apr	fri	185	2	999	0	nonexistent	-1.8	93.075	

Practical 4 (ML).ipynb

File Edit View Insert Runtime Tools Help

Share

+ Code + Text

Connect Gemini

```
[ ] #Shrutika Pradeep Bagdi (CS22130)
    df.shape
```

```
(41188, 21)
```

```
[ ] #Shrutika Pradeep Bagdi (CS22130)
    X = pd.get_dummies(X, dtype = int)
```

```
x
```

	age	duration	campaign	pdays	previous	emp_var_rate	cons_price_idx	cons_conf_idx	euribor3m	nr_employed	...	month_oct	month_sep	day_of_week_fri	day_of_week_mon	day_of_
0	44	210	1	999	0	1.4	93.444	-36.1	4.963	5228.1	...	0	0	0	0	
1	53	138	1	999	0	-0.1	93.200	-42.0	4.021	5195.8	...	0	0	1	0	
2	28	339	3	6	2	-1.7	94.055	-39.8	0.729	4991.6	...	0	0	0	0	
3	39	185	2	999	0	-1.8	93.075	-47.1	1.405	5099.1	...	0	0	1	0	
4	55	137	1	3	1	-2.9	92.201	-31.4	0.869	5076.2	...	0	0	1	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
41183	59	222	1	999	0	1.4	94.465	-41.8	4.866	5228.1	...	0	0	0	0	
41184	31	196	2	999	0	1.1	93.994	-36.4	4.860	5191.0	...	0	0	0	0	
41185	42	62	3	999	0	1.1	93.994	-36.4	4.857	5191.0	...	0	0	0	0	
41186	48	200	2	999	0	-3.4	92.431	-26.9	0.742	5017.5	...	1	0	0	0	
41187	25	112	4	999	0	1.1	93.994	-36.4	4.859	5191.0	...	0	0	1	0	

Practical 4 (ML).ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

```
[ ] #Shrutika Pradeep Bagdi (CS22130)
    X.columns
```

```
Index(['age', 'duration', 'campaign', 'pdays', 'previous', 'emp_var_rate',
       'cons_price_idx', 'cons_conf_idx', 'euribor3m', 'nr_employed',
       'job_admin.', 'job_blue-collar', 'job_entrepreneur', 'job_housemaid',
       'job_management', 'job_retired', 'job_self-employed', 'job_services',
       'job_student', 'job_technician', 'job_unemployed', 'job_unknown',
       'marital_divorced', 'marital_married', 'marital_single',
       'marital_unknown', 'education_basic.4y', 'education_basic.6y',
       'education_basic.9y', 'education_high.school', 'education_illiterate',
       'education_professional.course', 'education_university.degree',
       'education_unknown', 'default_no', 'default_unknown', 'default_yes',
       'housing_no', 'housing_unknown', 'housing_yes', 'loan_no',
       'loan_unknown', 'loan_yes', 'contact_cellular', 'contact_telephone',
       'month_apr', 'month_aug', 'month_dec', 'month_jul', 'month_jun',
       'month_mar', 'month_may', 'month_nov', 'month_oct', 'month_sep',
       'day_of_week_fri', 'day_of_week_mon', 'day_of_week_thu',
       'day_of_week_tue', 'day_of_week_wed', 'poutcome_failure',
       'poutcome_nonexistent', 'poutcome_success'],
      dtype='object')
```

```
[ ] #Shrutika Pradeep Bagdi (CS22130)
    from sklearn.model_selection import train_test_split
    X_train, X_test, y_train, y_test = train_test_split(X, y, train_size = 0.7, test_size = 0.25, random_state = 42)
```

```
#Shrutika Pradeep Bagdi (CS22130)
X_train.shape,X_test.shape
```

```
((28831, 63), (10297, 63))
```

Practical 4 (ML).ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

```
[ ] #Shrutika Pradeep Bagdi (CS22130)
    from sklearn.ensemble import RandomForestClassifier
    rf = RandomForestClassifier(n_estimators = 100, max_depth=5, random_state = 42)
    rf.fit(X_train, y_train)
```

```
RandomForestClassifier
RandomForestClassifier(max_depth=5, random_state=42)
```

```
[ ] #Shrutika Pradeep Bagdi (CS22130)
    from sklearn.metrics import accuracy_score, classification_report,confusion_matrix
```

```
[ ] #Shrutika Pradeep Bagdi (CS22130)
    y_pred = rf.predict(X_test)
```

```
#Shrutika Pradeep Bagdi (CS22130)
score = accuracy_score(y_pred, y_test)
print(score)
```

```
0.901039137612897
```

```
[ ] #Shrutika Pradeep Bagdi (CS22130)
    print(classification_report(y_pred, y_test))
    cm = confusion_matrix(y_pred, y_test)
```

```
              precision    recall  f1-score   support

           0       0.99      0.90      0.95      9992
```

Practical 4 (ML).ipynb

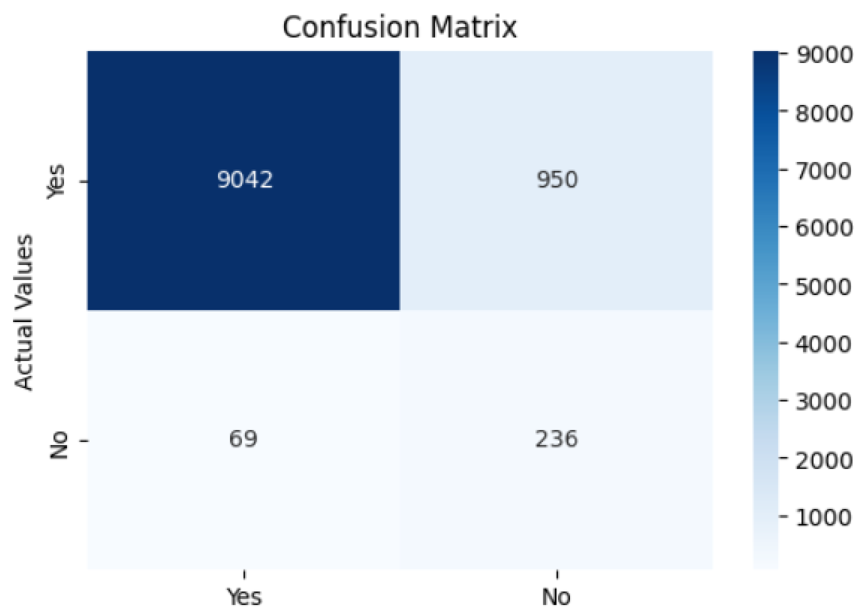
File Edit View Insert Runtime Tools Help

+ Code + Text

```
print("Confusion Matrix:",confusion_matrix(y_pred, y_test))
```

```
Confusion Matrix: [[9042  950]
 [  69  236]]
```

```
#Shrutika Pradeep Bagdi (CS22130)
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, cmap='Blues', fmt='g',
            xticklabels=['Yes', 'No'], yticklabels=['Yes', 'No'])
plt.xlabel('Predicted Values')
plt.ylabel('Actual Values')
plt.title('Confusion Matrix')
plt.show()
```



CONCLUSION:

DISCUSSION AND VIVA VOCE:

- Explain the steps of Random forest Algorithm
- What are some *Stopping Criteria* for *Random forest Algorithm*
- What do you mean by Bagging?
- Why does the Random Forest algorithm not require split sampling methods?

REFERENCE

- https://www.analyticsvidhya.com/blog/2021/05/bagging-25-questions-to-test-your-skills-on-random-forest-algorithm/
- https://www.ibm.com/in-en/topics/random-forest#:~:text=Random%20forest%20is%20a%20commonly,both%20classification%20and%20regression%20problems.