Project-1

Apply 1-logistic Regression 2-SVM 3-Decision Tree 4-RandomForest on the Loan dataset and check were you will get the best possible accuracy project note : Dependent Variable is Loan Status

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

data= pd.read_csv('loan.csv')

data
```

```
      Loan_ID  Gender Married Dependents     Education
Self_Employed  \
0    LP001002    Male      No          0      Graduate            No

1    LP001003    Male     Yes          1      Graduate            No

2    LP001005    Male     Yes          0      Graduate           Yes

3    LP001006    Male     Yes          0  Not Graduate            No

4    LP001008    Male      No          0      Graduate            No

..        ...     ...     ...        ...           ...           ...

609  LP002978  Female      No          0      Graduate            No

610  LP002979    Male     Yes         3+      Graduate            No

611  LP002983    Male     Yes          1      Graduate            No

612  LP002984    Male     Yes          2      Graduate            No

613  LP002990  Female      No          0      Graduate           Yes


     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term
\
0               5849                0.0         NaN             360.0

1               4583             1508.0       128.0             360.0

2               3000                0.0        66.0             360.0
```

```
3               2583             2358.0       120.0             360.0

4               6000                0.0       141.0             360.0

..               ...                ...         ...               ...

609             2900                0.0        71.0             360.0

610             4106                0.0        40.0             180.0

611             8072              240.0       253.0             360.0

612             7583                0.0       187.0             360.0

613             4583                0.0       133.0             360.0


     Credit_History Property_Area Loan_Status
0               1.0         Urban           Y
1               1.0         Rural           N
2               1.0         Urban           Y
3               1.0         Urban           Y
4               1.0         Urban           Y
..              ...           ...         ...
609             1.0         Rural           Y
610             1.0         Rural           Y
611             1.0         Urban           Y
612             1.0         Urban           Y
613             0.0     Semiurban           N

[614 rows x 13 columns]

data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Loan_ID            614 non-null    object
 1   Gender             601 non-null    object
 2   Married            611 non-null    object
 3   Dependents         599 non-null    object
 4   Education          614 non-null    object
 5   Self_Employed      582 non-null    object
 6   ApplicantIncome    614 non-null    int64
 7   CoapplicantIncome  614 non-null    float64
 8   LoanAmount         592 non-null    float64
 9   Loan_Amount_Term   600 non-null    float64
 10  Credit_History     564 non-null    float64
```

```
 11  Property_Area      614 non-null    object
 12  Loan_Status        614 non-null    object
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB
```

```
data.columns
```

```
Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome',
'LoanAmount',
       'Loan_Amount_Term', 'Credit_History', 'Property_Area',
'Loan_Status'],
      dtype='object')
```

```
data.isnull().sum()
```

```
Loan_ID               0
Gender               13
Married               3
Dependents           15
Education             0
Self_Employed        32
ApplicantIncome       0
CoapplicantIncome     0
LoanAmount           22
Loan_Amount_Term     14
Credit_History       50
Property_Area         0
Loan_Status           0
dtype: int64
```

```
data=data.dropna()
data
```

```
      Loan_ID  Gender Married Dependents     Education
Self_Employed  \
1    LP001003    Male     Yes          1      Graduate            No

2    LP001005    Male     Yes          0      Graduate           Yes

3    LP001006    Male     Yes          0  Not Graduate            No

4    LP001008    Male      No          0      Graduate            No

5    LP001011    Male     Yes          2      Graduate           Yes

..        ...     ...     ...        ...           ...           ...

609  LP002978  Female      No          0      Graduate            No

610  LP002979    Male     Yes         3+      Graduate            No
```

```
611  LP002983    Male     Yes          1      Graduate            No

612  LP002984    Male     Yes          2      Graduate            No

613  LP002990  Female      No          0      Graduate           Yes


     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term
\
1               4583             1508.0       128.0             360.0

2               3000                0.0        66.0             360.0

3               2583             2358.0       120.0             360.0

4               6000                0.0       141.0             360.0

5               5417             4196.0       267.0             360.0

..               ...                ...         ...               ...

609             2900                0.0        71.0             360.0

610             4106                0.0        40.0             180.0

611             8072              240.0       253.0             360.0

612             7583                0.0       187.0             360.0

613             4583                0.0       133.0             360.0


     Credit_History Property_Area Loan_Status
1               1.0         Rural           N
2               1.0         Urban           Y
3               1.0         Urban           Y
4               1.0         Urban           Y
5               1.0         Urban           Y
..              ...           ...         ...
609             1.0         Rural           Y
610             1.0         Rural           Y
611             1.0         Urban           Y
612             1.0         Urban           Y
613             0.0     Semiurban           N

[480 rows x 13 columns]

data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 480 entries, 1 to 613
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Loan_ID            480 non-null    object 
 1   Gender             480 non-null    object 
 2   Married            480 non-null    object 
 3   Dependents         480 non-null    object 
 4   Education          480 non-null    object 
 5   Self_Employed      480 non-null    object 
 6   ApplicantIncome    480 non-null    int64  
 7   CoapplicantIncome  480 non-null    float64
 8   LoanAmount         480 non-null    float64
 9   Loan_Amount_Term   480 non-null    float64
 10  Credit_History     480 non-null    float64
 11  Property_Area      480 non-null    object 
 12  Loan_Status        480 non-null    object 
dtypes: float64(4), int64(1), object(8)
memory usage: 52.5+ KB
```

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

```
datatype = data.dtypes==object
columns = data.columns[datatype].tolist()
data[columns]=data[columns].apply(lambda val : le.fit_transform(val))
```

```
C:\Users\Smita\AppData\Local\Temp\ipykernel_13000\1689896601.py:3:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  data[columns]=data[columns].apply(lambda val :
le.fit_transform(val))
```

```
data
```

```
     Loan_ID  Gender  Married  Dependents  Education  Self_Employed  \
1          0       1        1           1          0              0   
2          1       1        1           0          0              1   
3          2       1        1           0          1              0   
4          3       1        0           0          0              0   
5          4       1        1           2          0              1   
..       ...     ...      ...         ...        ...            ...   
609      475       0        0           0          0              0   
610      476       1        1           3          0              0   
611      477       1        1           1          0              0   
```

```
612      478       1        1           2          0              0
613      479       0        0           0          0              1

     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term
\
1               4583             1508.0       128.0             360.0

2               3000                0.0        66.0             360.0

3               2583             2358.0       120.0             360.0

4               6000                0.0       141.0             360.0

5               5417             4196.0       267.0             360.0

..               ...                ...         ...               ...

609             2900                0.0        71.0             360.0

610             4106                0.0        40.0             180.0

611             8072              240.0       253.0             360.0

612             7583                0.0       187.0             360.0

613             4583                0.0       133.0             360.0


     Credit_History  Property_Area  Loan_Status
1               1.0              0            0
2               1.0              2            1
3               1.0              2            1
4               1.0              2            1
5               1.0              2            1
..              ...            ...          ...
609             1.0              0            1
610             1.0              0            1
611             1.0              2            1
612             1.0              2            1
613             0.0              1            0

[480 rows x 13 columns]
```

```
data=data.astype('int64')
data
```

```
     Loan_ID  Gender  Married  Dependents  Education  Self_Employed  \
1          0       1        1           1          0              0
2          1       1        1           0          0              1
```

```
3          2       1        1           0          1              0
4          3       1        0           0          0              0
5          4       1        1           2          0              1
..       ...     ...      ...         ...        ...            ...
609      475       0        0           0          0              0
610      476       1        1           3          0              0
611      477       1        1           1          0              0
612      478       1        1           2          0              0
613      479       0        0           0          0              1

     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term
\
1               4583               1508         128               360

2               3000                  0          66               360

3               2583               2358         120               360

4               6000                  0         141               360

5               5417               4196         267               360

..               ...                ...         ...               ...

609             2900                  0          71               360

610             4106                  0          40               180

611             8072                240         253               360

612             7583                  0         187               360

613             4583                  0         133               360


     Credit_History  Property_Area  Loan_Status
1                 1              0            0
2                 1              2            1
3                 1              2            1
4                 1              2            1
5                 1              2            1
..              ...            ...          ...
609               1              0            1
610               1              0            1
611               1              2            1
612               1              2            1
613               0              1            0

[480 rows x 13 columns]
```

```
data.isnull().sum()
```

```
Loan_ID              0
Gender               0
Married              0
Dependents           0
Education            0
Self_Employed        0
ApplicantIncome      0
CoapplicantIncome    0
LoanAmount           0
Loan_Amount_Term     0
Credit_History       0
Property_Area        0
Loan_Status          0
dtype: int64
```

```
from sklearn.feature_selection import VarianceThreshold
sector=VarianceThreshold()
info=sector.fit_transform(data)
```

```
info=sector.fit(data)
info.get_support()
```

```
array([ True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True])
```

```
#here we take important columns ie. as above all are important columns
X=data[['Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome',
'LoanAmount',
       'Loan_Amount_Term', 'Credit_History', 'Property_Area']].values
y=data['Loan_Status'].values
```

```
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,rando
m_state=50)
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()
lr.fit(x_train,y_train)
y_pred=lr.predict(x_test)
```

```
from sklearn.metrics import mean_squared_error
mean_squared_error(y_pred,y_test)
```

```
0.17708333333333334
```

```
from sklearn.metrics import accuracy_score
accuracy_score(y_pred,y_test)*100
```

```
82.29166666666666
```

svm

data

```
     Loan_ID  Gender  Married  Dependents  Education  Self_Employed  \
1          0       1        1           1          0              0
2          1       1        1           0          0              1
3          2       1        1           0          1              0
4          3       1        0           0          0              0
5          4       1        1           2          0              1
..       ...     ...      ...         ...        ...            ...
609      475       0        0           0          0              0
610      476       1        1           3          0              0
611      477       1        1           1          0              0
612      478       1        1           2          0              0
613      479       0        0           0          0              1

     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term
\
1               4583               1508         128               360

2               3000                  0          66               360

3               2583               2358         120               360

4               6000                  0         141               360

5               5417               4196         267               360

..               ...                ...         ...               ...

609             2900                  0          71               360

610             4106                  0          40               180

611             8072                240         253               360

612             7583                  0         187               360

613             4583                  0         133               360


     Credit_History  Property_Area  Loan_Status
1                 1              0            0
2                 1              2            1
3                 1              2            1
4                 1              2            1
5                 1              2            1
..              ...            ...          ...
609               1              0            1
```

```
610               1               0               1
611               1               2               1
612               1               2               1
613               0               1               0

[480 rows x 13 columns]
```

```
X = data[['Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome',
'LoanAmount',
       'Loan_Amount_Term', 'Credit_History', 'Property_Area']].values
y = data['Loan_Status'].values
```

```
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,random_state=50)
from sklearn.svm import SVC
svc = SVC()
svc.fit(x_train,y_train)
```

```
SVC()
```

```
y_pred1=svc.predict(x_test)
```

```
from sklearn.metrics import accuracy_score
accuracy_score(y_pred,y_test)*100
```

```
82.29166666666666
```

```
from sklearn.svm import SVC
svc = SVC(kernel = 'linear')
svc.fit(x_train,y_train)
y_pred1=svc.predict(x_test)
from sklearn.metrics import accuracy_score
accuracy_score(y_pred,y_test)*100
```

```
82.29166666666666
```

```
from sklearn.svm import SVC
svc = SVC(kernel = 'sigmoid')
svc.fit(x_train,y_train)
y_pred1=svc.predict(x_test)
from sklearn.metrics import accuracy_score
accuracy_score(y_pred,y_test)*100
```

```
82.29166666666666
```

```
from sklearn.svm import SVC
svc = SVC(kernel = 'poly')
svc.fit(x_train,y_train)
y_pred1=svc.predict(x_test)
```

```
from sklearn.metrics import accuracy_score
accuracy_score(y_pred,y_test)*100
```

```
82.29166666666666
```

Decision Tree

data

```
     Loan_ID  Gender  Married  Dependents  Education  Self_Employed  \
1          0       1        1           1          0              0
2          1       1        1           0          0              1
3          2       1        1           0          1              0
4          3       1        0           0          0              0
5          4       1        1           2          0              1
..       ...     ...      ...         ...        ...            ...
609      475       0        0           0          0              0
610      476       1        1           3          0              0
611      477       1        1           1          0              0
612      478       1        1           2          0              0
613      479       0        0           0          0              1

     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term
\
1               4583               1508         128               360

2               3000                  0          66               360

3               2583               2358         120               360

4               6000                  0         141               360

5               5417               4196         267               360

..               ...                ...         ...               ...

609             2900                  0          71               360

610             4106                  0          40               180

611             8072                240         253               360

612             7583                  0         187               360

613             4583                  0         133               360


     Credit_History  Property_Area  Loan_Status
1                 1              0            0
```

```
2                1              2            1
3                1              2            1
4                1              2            1
5                1              2            1
..             ...            ...          ...
609              1              0            1
610              1              0            1
611              1              2            1
612              1              2            1
613              0              1            0

[480 rows x 13 columns]
```

```
data.columns
```

```
Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome',
'LoanAmount',
       'Loan_Amount_Term', 'Credit_History', 'Property_Area',
'Loan_Status'],
      dtype='object')
```

```
X= data[['Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome',
'LoanAmount',
       'Loan_Amount_Term', 'Credit_History', 'Property_Area']].values
y = data['Loan_Status'].values
```

```
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,random_state=50)
from sklearn.tree import DecisionTreeClassifier
cls =DecisionTreeClassifier()
cls.fit(x_train,y_train)
```

```
DecisionTreeClassifier()
```

```
ypred2=cls.predict(x_test)
```

```
from sklearn.metrics import accuracy_score
accuracy_score(ypred2,y_test)*100
```

```
80.20833333333334
```

RandomForest

```
data
```

```
     Loan_ID  Gender  Married  Dependents  Education  Self_Employed  \
1          0       1        1           1          0              0
2          1       1        1           0          0              1
3          2       1        1           0          1              0
```

```
4          3       1        0           0          0              0
5          4       1        1           2          0              1
..       ...     ...      ...         ...        ...            ...
609      475       0        0           0          0              0
610      476       1        1           3          0              0
611      477       1        1           1          0              0
612      478       1        1           2          0              0
613      479       0        0           0          0              1

     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term
\
1               4583               1508         128               360

2               3000                  0          66               360

3               2583               2358         120               360

4               6000                  0         141               360

5               5417               4196         267               360

..               ...                ...         ...               ...

609             2900                  0          71               360

610             4106                  0          40               180

611             8072                240         253               360

612             7583                  0         187               360

613             4583                  0         133               360


     Credit_History  Property_Area  Loan_Status
1                 1              0            0
2                 1              2            1
3                 1              2            1
4                 1              2            1
5                 1              2            1
..              ...            ...          ...
609               1              0            1
610               1              0            1
611               1              2            1
612               1              2            1
613               0              1            0

[480 rows x 13 columns]
```

```
from sklearn.feature_selection import VarianceThreshold
sector=VarianceThreshold()
info=sector.fit_transform(data)
```

```
info=sector.fit(data)
info.get_support()
```

```
array([ True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True])
```

```
X= data[['Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome',
'LoanAmount',
       'Loan_Amount_Term', 'Credit_History', 'Property_Area']].values
y = data['Loan_Status'].values
```

```
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,random_state=50)
from sklearn.ensemble import RandomForestClassifier
rfc =RandomForestClassifier()
rfc.fit(x_train,y_train)
```

```
RandomForestClassifier()
```

```
ypred3=cls.predict(x_test)
```

```
from sklearn.metrics import accuracy_score
accuracy_score(ypred2,y_test)*100
```

```
80.20833333333334
```