Metrics syntax:

from sklearn.metrics import (metric)

mse_adv = mean_squared_error(y_test_adv, y_pred_adv)

metrics=accuracy_score(Y_test,y_pred)

precision = precision_score(Y_test, Y_pred)

metrics = classification_report(y_test, y_pred)

metrics= confusion_matrix(y_test, y_pred)

metrics= mean_absolute_error(y_test, y_pred)

Models:

linear_model – LinearRegression()

LogisticRegression()

ensemble – AdaBoostClassifier()

RandomForestClassifier()

svm - SVC

neighbors – KneighborsClassifier()

tree - DecisionTreeClassifier()

RANDOM FOREST

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

data = pd.read_csv("diabetes.csv")
data = data.drop(columns=['patientID'])

x = data.drop(columns=['class'])
y = data["class"]

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)

model = RandomForestClassifier()
model.fit(x_train, y_train)

pred = model.predict(x_test)
metrics = accuracy_score(y_test, pred)
print("Accuracy:", metrics)
```

SVM

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import precision_score

data=pd.read_csv("diabetes.csv")
data.drop(columns=['patientID'])

x=data.drop(columns=["class"])
y=data["class"]

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)

model=SVC()
model.fit(x_train,y_train)

y_pred=model.predict(x_test)
precision=precision_score(y_pred,y_test)
print("precision:",precision_score)
```

Linear Regression

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error

data=pd.read_csv("salary.csv")
data.dropna(inplace=True)

x=data['YearsExperience'].values.reshape(-1,1)
y=data['Salary'].values

model=LinearRegression()
model.fit(x,y)

y_pred=model.predict(x)

mae=mean_absolute_error(y_pred,y)
print("mean absolute error:",mae)

plt.scatter(x,y,color='blue')
plt.plot(x,y_pred,color='red')
plt.xlabel("YearsExperience")
plt.ylabel("Salary")
plt.title("LinearRegression")
plt.show()
```

ADABOOST

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import AdaBoostClassifier
from sklearn.metrics import classification_report

data=pd.read_csv("social.csv")
data.dropna(inplace=True)

x=data.drop(columns=['UserID'])
y=data['Clicked']

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)

model=AdaBoostClassifier()
model.fit(x_train,y_train)

y_pred=model.predict(x_test)

metric=classification_report(y_pred,y_test)
print("classification report:",metric)
```