```
import numpy as np
import pandas as pd
import sklearn
from sklearn import svm
from sklearn import metrics
from sklearn.model_selection import train_test_split
#data url load the data in
data_url = "https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wdbc.data"

df = pd.read_csv(data_url, delimiter = ",", header = None)
display(df)


y = df[1].values
x = df.iloc[:, 2:32]
x = np.array(x)

rs_cols = ["train accuracy", "test accuracy", "precision", "recall"] #column names for the result table
rs_rows = ["SVM1", "SVM2", "SVM3"] #row names for the result table
rs_table = pd.DataFrame(columns=rs_cols, index=rs_rows) #creating the result table

display(rs_table)

print("Wait a moment computing SVM1 Values..")

clf = svm.SVC(C = 1e10, kernel = 'linear') # Linear Kernel

train_acc = []
test_acc = []
prec = []
recall = []


for i in range(20):
    (x_train, x_test, y_train, y_test) = train_test_split(x, y, test_size = 0.3)

    scaler = sklearn.preprocessing.StandardScaler()
    x_train = scaler.fit_transform(x_train)
    x_test = scaler.fit_transform(x_test)

    clf = clf.fit(x_train, y_train)

    #for train dataset
    y_train_pred = clf.predict(x_train)
    train_acc.append(metrics.accuracy_score(y_train, y_train_pred))

    #for test dataset
    y_test_pred = clf.predict(x_test)

    test_acc.append(metrics.accuracy_score(y_test, y_test_pred))
    prec.append(metrics.precision_score(y_test, y_test_pred, pos_label = 'M'))
    recall.append(metrics.recall_score(y_test, y_test_pred, pos_label = 'M'))

#compute average of 20 performance
mean_train_acc = np.mean(train_acc)
mean_test_acc = np.mean(test_acc)
mean_prec = np.mean(prec)
mean_recall = np.mean(recall)

rs_table.loc['SVM1']['train accuracy'] = mean_train_acc
rs_table.loc['SVM1']['test accuracy'] = mean_test_acc
rs_table.loc['SVM1']['precision'] = mean_prec
rs_table.loc['SVM1']['recall'] = mean_recall

print(rs_table)




print("Wait a Moment Calculating for SVM2....")

clf = svm.SVC(C = 1e10, kernel = 'rbf') # RBF Kernel

train_acc = []
test_acc = []
prec = []
```

```
recall = []


for i in range(20):
    (x_train, x_test, y_train, y_test) = train_test_split(x, y, test_size = 0.3)

    scaler = sklearn.preprocessing.StandardScaler()
    x_train = scaler.fit_transform(x_train)
    x_test = scaler.fit_transform(x_test)

    clf = clf.fit(x_train, y_train)

    #for train dataset
    y_train_pred = clf.predict(x_train)
    train_acc.append(metrics.accuracy_score(y_train, y_train_pred))

    #for test dataset
    y_test_pred = clf.predict(x_test)

    test_acc.append(metrics.accuracy_score(y_test, y_test_pred))
    prec.append(metrics.precision_score(y_test, y_test_pred, pos_label = 'M'))
    recall.append(metrics.recall_score(y_test, y_test_pred, pos_label = 'M'))

#compute average of 20 performance
mean_train_acc = np.mean(train_acc)
mean_test_acc = np.mean(test_acc)
mean_prec = np.mean(prec)
mean_recall = np.mean(recall)

rs_table.loc['SVM2']['train accuracy'] = mean_train_acc
rs_table.loc['SVM2']['test accuracy'] = mean_test_acc
rs_table.loc['SVM2']['precision'] = mean_prec
rs_table.loc['SVM2']['recall'] = mean_recall

print(rs_table)


print("Calculating for SVM3....")

clf = svm.SVC(C = 10, kernel = 'rbf') # RBF Kernel with varying C value

train_acc = []
test_acc = []
prec = []
recall = []


for i in range(20):
    (x_train, x_test, y_train, y_test) = train_test_split(x, y, test_size = 0.3)

    scaler = sklearn.preprocessing.StandardScaler()
    x_train = scaler.fit_transform(x_train)
    x_test = scaler.fit_transform(x_test)

    clf = clf.fit(x_train, y_train)

    #for train dataset
    y_train_pred = clf.predict(x_train)
    train_acc.append(metrics.accuracy_score(y_train, y_train_pred))

    #for test dataset
    y_test_pred = clf.predict(x_test)

    test_acc.append(metrics.accuracy_score(y_test, y_test_pred))
    prec.append(metrics.precision_score(y_test, y_test_pred, pos_label = 'M'))
    recall.append(metrics.recall_score(y_test, y_test_pred, pos_label = 'M'))

#compute average of 20 performance
mean_train_acc = np.mean(train_acc)
mean_test_acc = np.mean(test_acc)
mean_prec = np.mean(prec)
mean_recall = np.mean(recall)

rs_table.loc['SVM3']['train accuracy'] = mean_train_acc
rs_table.loc['SVM3']['test accuracy'] = mean_test_acc
rs_table.loc['SVM3']['precision'] = mean_prec
rs_table.loc['SVM3']['recall'] = mean_recall
```

```
print(rs_table)
```

	0	1	2	3	4	5	6	7	8	9	...	22	23	24	25	26	27	28
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	...	25.380	17.33	184.60	2019.0	0.16220	0.66560	0.7119
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	...	24.990	23.41	158.80	1956.0	0.12380	0.18660	0.2416
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	...	23.570	25.53	152.50	1709.0	0.14440	0.42450	0.4504
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	...	14.910	26.50	98.87	567.7	0.20980	0.86630	0.6869
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430	...	22.540	16.67	152.20	1575.0	0.13740	0.20500	0.4000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
564	926424	M	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	...	25.450	26.40	166.10	2027.0	0.14100	0.21130	0.4107
565	926682	M	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	...	23.690	38.25	155.00	1731.0	0.11660	0.19220	0.3215
566	926954	M	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302	...	18.980	34.12	126.70	1124.0	0.11390	0.30940	0.3403
567	927241	M	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200	...	25.740	39.42	184.60	1821.0	0.16500	0.86810	0.9387
568	92751	B	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000	...	9.456	30.37	59.16	268.6	0.08996	0.06444	0.0000

569 rows × 32 columns

	train accuracy	test accuracy	precision	recall
SVM1	NaN	NaN	NaN	NaN
SVM2	NaN	NaN	NaN	NaN
SVM3	NaN	NaN	NaN	NaN

```
Wait a moment computing SVM1 Values..
     train accuracy test accuracy precision    recall
SVM1            1.0      0.947368  0.911634  0.951289
SVM2            NaN           NaN       NaN       NaN
SVM3            NaN           NaN       NaN       NaN
Wait a Moment Calculating for SVM2....
     train accuracy test accuracy precision    recall
SVM1            1.0      0.947368  0.911634  0.951289
SVM2            1.0      0.953216  0.925467  0.950779
SVM3            NaN           NaN       NaN       NaN
Calculating for SVM3....
     train accuracy test accuracy precision    recall
SVM1            1.0      0.947368  0.911634  0.951289
SVM2            1.0      0.953216  0.925467  0.950779
SVM3       0.992462      0.970175   0.97054  0.949447
```

Start coding or generate with AI.