INTELLIGENT SYSTEMS LAB-3 (06/09/2021)

***NAME**: Harshvardhan Agarwal | **SEC**: A | **REG NO**.: 201800524*

The degree of freedom is the number of features present in the training data used to train the model.

Initially, features used for training = 30

Hence, degree of freedom = 30

Source Code

```
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn import metrics

cancer = datasets.load_breast_cancer()

print("Features: ", cancer.feature_names)
print("Labels: ", cancer.target_names)

X_train, X_test, y_train, y_test = train_test_split(cancer.data, cancer.target, test_size=0.3, random_state=42)
clf = svm.SVC(kernel='linear')  # Linear Kernel

# Train the model using the training sets
clf.fit(X_train, y_train)

# Predict the response for test dataset
y_pred = clf.predict(X_test)

# Model Accuracy: how often is the classifier correct?
print("Accuracy:", metrics.accuracy_score(y_test, y_pred))

# Model Precision: what percentage of positive tuples are labeled as such?
print("Precision:", metrics.precision_score(y_test, y_pred))

# Model Recall: what percentage of positive tuples are labelled as such?
print("Recall:", metrics.recall_score(y_test, y_pred))
```

Output

Accuracy: 0.9649122807017544

Precision: 0.9636363636363636

Recall: 0.9814814814814815

Let’s drop arbitrary features to decrease the degree of freedom.

Note: Different features might have a different correlation with the labels and thus the accuracy, precision, and recall values will vary accordingly.

Initial Features: ['mean radius' 'mean texture' 'mean perimeter' 'mean area'

'mean smoothness' 'mean compactness' 'mean concavity'

'mean concave points' 'mean symmetry' 'mean fractal dimension'

'radius error' 'texture error' 'perimeter error' 'area error'

'smoothness error' 'compactness error' 'concavity error'

'concave points error' 'symmetry error' 'fractal dimension error'

'worst radius' 'worst texture' 'worst perimeter' 'worst area'

'worst smoothness' 'worst compactness' 'worst concavity'

'worst concave points' 'worst symmetry' 'worst fractal dimension']

Let’s drop all features related to fractal dimensions i.e ‘**mean fractal dimension**’, ‘**fractal dimension error**’, and ‘**worst fractal dimension**’

Degree of freedom now = 27

Source Code

```
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn import metrics
import pandas as pd

cancer = load_breast_cancer()
df = pd.DataFrame(cancer.data, columns=cancer.feature_names)
df['target'] = pd.Series(cancer.target)
# print(df.head())

# Dropping fields related to fractal dimension to reduce degree of freedom
df.drop(["mean fractal dimension", "fractal dimension error", "worst fractal dimension"], axis=1, inplace=True)
# print(df.head())

data = df.drop(['target'], axis=1).copy()
# print(data.head())
target = df['target']
# print(target.head())
X_train, X_test, y_train, y_test = train_test_split(data, target, test_size=0.3, random_state=42)
clf = svm.SVC(kernel='linear')  # Linear Kernel

# Train the model using the training sets
clf.fit(X_train, y_train)

# Predict the response for test dataset
y_pred = clf.predict(X_test)
```

```
# Model Accuracy: how often is the classifier correct?
print("Accuracy:", metrics.accuracy_score(y_test, y_pred))

# Model Precision: what percentage of positive tuples are labeled as such?
print("Precision:", metrics.precision_score(y_test, y_pred))

# Model Recall: what percentage of positive tuples are labelled as such?
print("Recall:", metrics.recall_score(y_test, y_pred))
```

Output

```
Accuracy: 0.9707602339181286
Precision: 0.9724770642201835
Recall: 0.9814814814814815

Process finished with exit code 0
```

Our model is performing better now than it did before. This implies that the features that we dropped were not very relevant to the model and were increasing the noise. We can try different combinations and try to further improve our model's performance