optimizing spam filtering

File Edit View Insert Runtime Tools Help

+ Code + Text

```
df.describe()
```

```
[31] df.shape
```

```
[32] df['label'].value_counts().plot(kind="bar",figsize=(12,6))
     plt.xticks(np.arange(2),('Non spam','spam'),rotation=0);
```

```
[33] # performing feature Scaling opjeration using standard scaller on x part of the dataset because]
     # there different type of values in the colums
     sc=StandardScaler()
     x_bal=sc.fit_transform(x_bal)
     x_bal = pd.DataFrame(x_bal,columns=names)
```

```
[34] #splitting data into train and validation sets using train_test_split

     from sklearn.model_selection import train_test_split
     X_train,X_test,y_train,y_test = train_test_split(X,y,test_size = 0.20, random_state = 0)

     ##train size 80% and test size 20%
```

```
[ ] from sklearn.tree import DecisionTreeClassifier
    model = DecisionTreeClassifier()
    model.fit(X_train_res,Y_train_res)
```

optimizing spam filtering - Colab

colab.research.google.com/drive/1lBMLtGNspuP51Mpoq5eQmn20F7t7t6rQ#scrollTo=opAmPpqtz5Nf

optimizing spam filtering

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

```
df["label"].value_counts().plot(kind="bar",figsize=(12,6))
plt.xticks(np.arange(2),('Non spam','spam'),rotation=0);
```

[33]
```
# performing feature Scaling op[eration using standard scaller on x part of the dataset because]
# there different type of values in the colums
sc=StandardScaler()
x_bal=sc.fit_transform(x_bal)
x_bal = pd.DataFrame(x_bal,columns=names)
```

[34]
```
#splitting data into train and validation sets using train_test_split

from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size = 0.20, random_state = 0)

##train size 80% and test size 20%
```

[ ]
```
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier()
model.fit(X_train_res,Y_train_res)
```

[ ]
```
from sklearn.ensemble import RandomForestClassifier
model1 =RandomForestClassifier()
model1.fit(X_train_res,y_train_res)
```

ENG 8:30 PM