

```

import pandas as pd

df = pd.read_csv('pd_speech_features.csv')
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 756 entries, 0 to 755
Columns: 755 entries, id to class
dtypes: float64(749), int64(6)
memory usage: 4.4 MB

df.head()

{"type": "dataframe", "variable_name": "df"}

null_values=df.isnull().sum()

null_values

id                0
gender            0
PPE              0
DFA              0
RPDE             0
..              .
tqwt_kurtosisValue_dec_33  0
tqwt_kurtosisValue_dec_34  0
tqwt_kurtosisValue_dec_35  0
tqwt_kurtosisValue_dec_36  0
class            0
Length: 755, dtype: int64

y = df.loc[:, 'class']
X = df.drop(['class', 'id'], axis=1)

from sklearn.preprocessing import StandardScaler

X

{"type": "dataframe", "variable_name": "X"}

y

0      1
1      1
2      1
3      1
4      1
..    ..
751    0
752    0
753    0

```

```
754     0
```

```
755     0
```

```
Name: class, Length: 756, dtype: int64
```

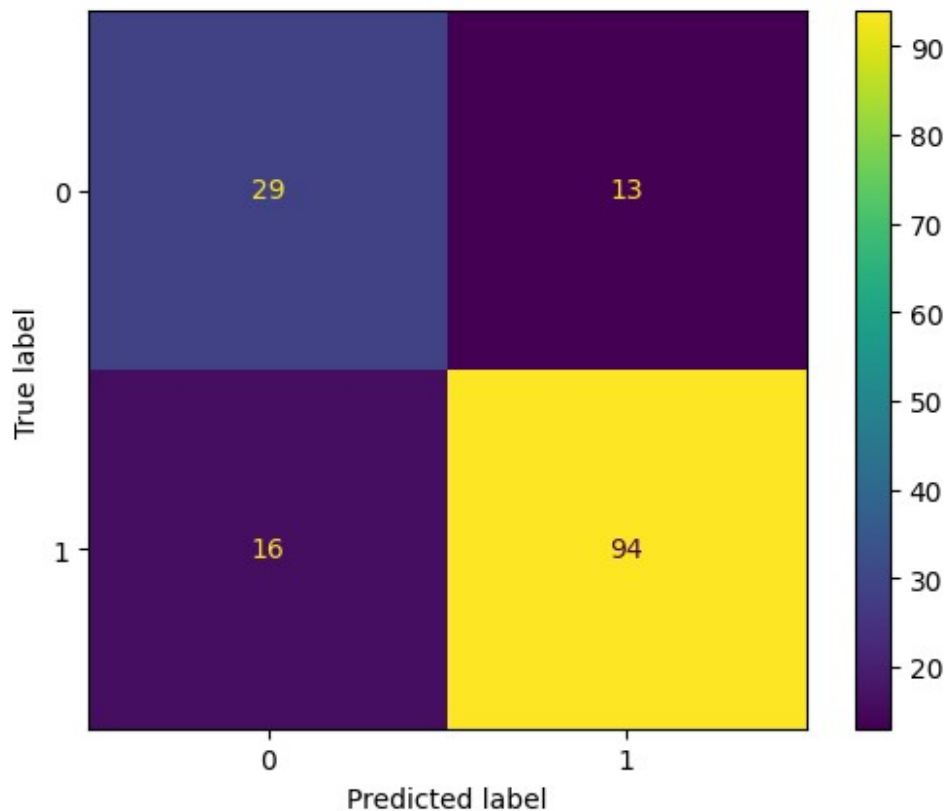
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=80)
```

```
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

```
from sklearn.svm import SVC
classifier_1 = SVC(kernel='linear')
classifier_2 = SVC(kernel='rbf')
classifier_3 = SVC(kernel='poly')
```

```
from sklearn.metrics import accuracy_score, confusion_matrix,
ConfusionMatrixDisplay
import matplotlib.pyplot as plt
```

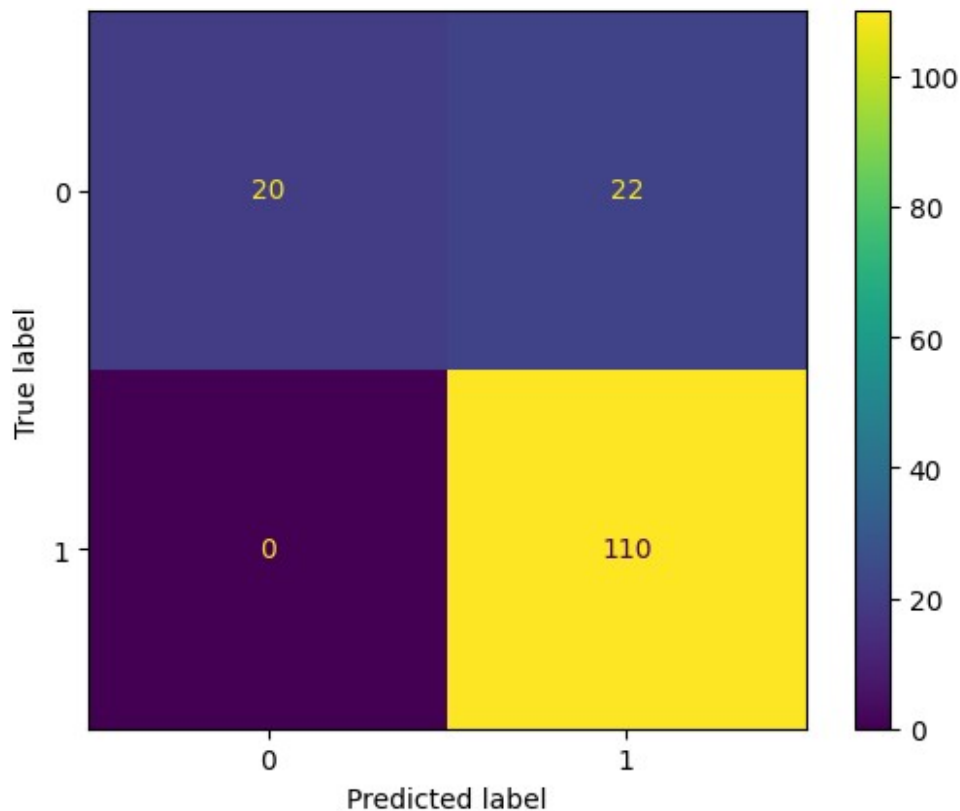
```
classifier_1.fit(X_train, y_train)
y_pred_1 = classifier_1.predict(X_test)
confusion_matrix_1 = confusion_matrix(y_test, y_pred_1)
cm_display_1 = ConfusionMatrixDisplay(confusion_matrix =
confusion_matrix_1, display_labels = [0, 1])
cm_display_1.plot()
plt.show()
```



```
accuracy_1 = accuracy_score(y_test,y_pred_1)
print(f"Acuracy dla liniowego kernela{accuracy_1}")
```

Acuracy dla liniowego kernela0.8092105263157895

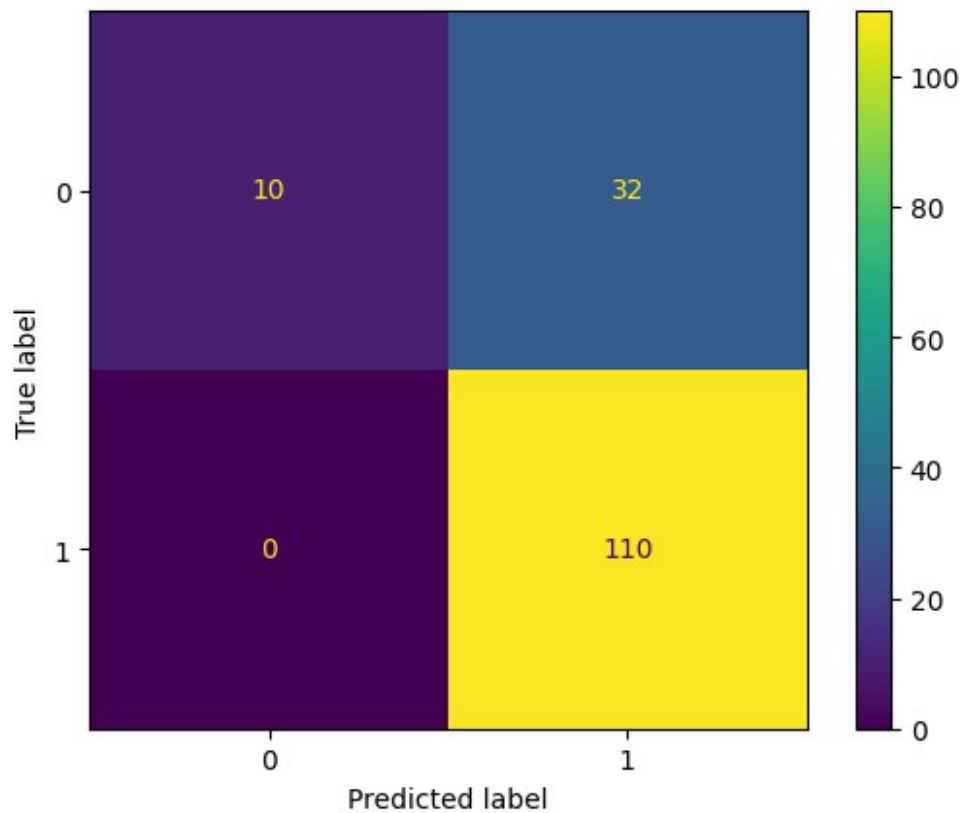
```
classifier_2.fit(X_train, y_train)
y_pred_2 = classifier_2.predict(X_test)
confusion_matrix_2 = confusion_matrix(y_test, y_pred_2)
cm_display_2 = ConfusionMatrixDisplay(confusion_matrix =
confusion_matrix_2, display_labels = [0, 1])
cm_display_2.plot()
plt.show()
```



```
accuracy_2 = accuracy_score(y_test,y_pred_2)
print(f"Acuracy kernela rbf {accuracy_2}")
```

Acuracy kernela rbf 0.8552631578947368

```
classifier_3.fit(X_train, y_train)
y_pred_3 = classifier_3.predict(X_test)
confusion_matrix_3 = confusion_matrix(y_test, y_pred_3)
cm_display_3 = ConfusionMatrixDisplay(confusion_matrix =
confusion_matrix_3, display_labels = [0, 1])
cm_display_3.plot()
plt.show()
```



```
accuracy_3 = accuracy_score(y_test,y_pred_3)
print(f"Acuracy kernela wielowymianowego {accuracy_3}")
```

Acuracy kernela wielowymianowego 0.7894736842105263