Baseline classifier

December 22, 2022

[1]:

```
# We import some useful libraries.
import pandas as pd
import numpy as np
import sklearn
from sklearn.model_selection import cross_val_score
from sklearn.metrics import confusion_matrix
from sklearn import model_selection
```

[2]:

```
from sklearn.model_selection import train_test_split

data= pd.read_csv("train.csv")

data['Lead'].replace({'Male':1, 'Female':0}, inplace = True)

# Separate the target variable from the dataframe as we cannot train the model␣
 ↪with the target variable.
X = data.drop(columns = ["Lead"])
y = data['Lead']

# We split the data into train and test dataframes.
# random_state seed gives us the same train and test datasets no matter the␣
 ↪times we split it.
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state = 4045)
```

[3]:

```
from sklearn.dummy import DummyClassifier
dummy_clf = DummyClassifier(strategy = "constant", constant = 1)
dummy_clf.fit(X_train, y_train)
```

[3]:

```
DummyClassifier(constant=1, strategy='constant')
```

[4]:

```
from sklearn.metrics import accuracy_score, precision_score, recall_score,␣
 ↪f1_score

print('Training set metrics:')
print('Accuracy:', accuracy_score(y_train, dummy_clf.predict(X_train)))
print('Precision:', precision_score(y_train, dummy_clf.predict(X_train)))
print('Recall:', recall_score(y_train, dummy_clf.predict(X_train)))
print('F1:', f1_score(y_train, dummy_clf.predict(X_train)))
```

```python

print('\n')

print('Test set metrics:')
print('Accuracy:', accuracy_score(y_test, dummy_clf.predict(X_test)))
print('Precision:', precision_score(y_test, dummy_clf.predict(X_test)))
print('Recall:', recall_score(y_test, dummy_clf.predict(X_test)))
print('F1:', f1_score(y_test, dummy_clf.predict(X_test)))
```

```
Training set metrics:
Accuracy: 0.7573812580231065
Precision: 0.7573812580231065
Recall: 1.0
F1: 0.8619430241051862


Test set metrics:
Accuracy: 0.75
Precision: 0.75
Recall: 1.0
F1: 0.8571428571428571
```