

Logistic Regression and Linear Discriminant Analysis

December 22, 2022

0.0.1 Logistic Regression

```
[ ]: import warnings
warnings.filterwarnings('ignore')
results4 = pd.DataFrame({"columns":[] , "train accuracy":[] , "test accuracy":
    ↳ [] })

for comb in combinations:
    d = pd.read_csv(r"C:\Users\billt\OneDrive\Desktop\SML_project\train.csv")

    x = d.drop("Lead", axis="columns")
    y = d["Lead"]

    x = x.drop(list(set(x) - set(comb)), axis="columns")

    I = [i for i in range(len(x))]
    random.shuffle(I)

    x_train = x.iloc[:3*int(np.ceil(len(I)/4)),]
    y_train = y.iloc[:3*int(np.ceil(len(I)/4)),]

    x_test = x.iloc[3*int(np.ceil(len(I)/4)):,:]
    y_test = y.iloc[3*int(np.ceil(len(I)/4)):,:]

    scaler1 = StandardScaler()
    scaler1.fit(x_train)
    x_train=scaler1.transform(x_train)
    x_train = pd.DataFrame(x_train)

    #     scaler2 = StandardScaler()
    #     scaler2.fit(x_test)
    x_test=scaler1.transform(x_test)
    x_test = pd.DataFrame(x_test)

    logreg = skl_lm.LogisticRegression(max_iter=1000000, solver="lbfgs")
    logreg.fit(x_train, y_train)
```

```

l=list()
l.append(comb)
l.append(np.mean(logreg.predict(x_train.iloc[:,])==y_train))
l.append(np.mean(logreg.predict(x_test.iloc[:,])==y_test) )

results4.loc[len(results4.index)] = l

```

```
[ ]: results4.loc[results4["test accuracy"]==max(results4["test accuracy"])]
```

0.0.2 LDA

```
[ ]: results3 = pd.DataFrame({"columns":[] , "train accuracy":[] , "test accuracy":
    ↳ [] })

for comb in combinations:
    d = pd.read_csv(r"C:\Users\billt\OneDrive\Desktop\SML_project\train.csv")

    x = d.drop("Lead", axis="columns")
    y = d["Lead"]

    x = x.drop(list(set(x) - set(comb)), axis="columns")

    I = [i for i in range(len(x))]
    random.shuffle(I)

    x_train = x.iloc[:3*int(np.ceil(len(I)/4)),]
    y_train = y.iloc[:3*int(np.ceil(len(I)/4)),]

    x_test = x.iloc[3*int(np.ceil(len(I)/4)):,:]
    y_test = y.iloc[3*int(np.ceil(len(I)/4)):,:]

    #     scaler1 = StandardScaler()
    #     scaler1.fit(x_train)
    #     x_train=scaler1.transform(x_train)
    #     x_train = pd.DataFrame(x_train)

    #     x_test=scaler1.transform(x_test)
    #     x_test = pd.DataFrame(x_test)

    lda = skl_da.LinearDiscriminantAnalysis(solver="svd")
    lda.fit(x_train, y_train)

    l=list()
    l.append(comb)
    l.append(np.mean(lda.predict(x_train.iloc[:,])==y_train))

```

```
l.append(np.mean(lda.predict(x_test.iloc[:,]) == y_test) )  
  
results3.loc[len(results3.index)] = l
```

```
[ ]: results3.loc[results3["test accuracy"] == max(results3["test accuracy"])]
```