# LR

January 10, 2022

## 1 LogisticalRegression mode Pair Programmed

### 1.0.1 by Zahir and Jaap

[1]:

```python
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import GridSearchCV
from ipynb.fs.full.base_model import BaseModel

import json
import numpy as np

import time
import warnings
warnings.filterwarnings('ignore')
from datetime import datetime
import pickle
import joblib
```

[5]:

```python
class LogisticalRegression(BaseModel):
    instance = "Logistical Regression"

    @classmethod
    def grid_search(self,model,x_train, x_test, y_train, y_test,scoring):

        solvers = ['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga']
        penalty = ['none', 'l1','l2','elasticnet']
        c_values = [100, 10, 1.0, 0.1, 0.01]

        param_grid = dict(solver=solvers,penalty=penalty,C=c_values)

        start_time = time.perf_counter()
        clf = GridSearchCV(model, param_grid, cv=5, scoring=scoring, n_jobs=5)
        end_time = time.perf_counter()
        print(f"Duration Gridsearch: {end_time - start_time:04f}")

        start_time = time.perf_counter()
        clf.fit(x_train, y_train)
        end_time = time.perf_counter()

        # SAVING MODEL
#         print("Saving model")
#         dt = datetime.today().strftime('%Y-%m-%d-%H:%M:%S')
#         filename = './pkls/'+dt+'-LR.pkl'
#         print(filename)
#         joblib.dump(clf, filename)

        print(f"Duration fitting: {end_time - start_time:04f}")
        print()

        print("Best parameters set found on development set:")
        print(clf.best_params_)
        print(clf.best_estimator_)

        super().model_accuracy(clf, x_train, x_test, y_train, y_test)


    @classmethod
    def train(self, data, scoring="precision_weighted"):

        train = data["train"]
        test = data["test"]

        x_train = np.array(train['features'])
        y_train = np.array(train['emotions'])
        x_test = np.array(test['features'])
        y_test = np.array(test['emotions'])


         #GridSearch
        self.grid_search(LogisticRegression(), x_train, x_test, y_train,␣
↪y_test,scoring)
```