



SAPIENZA  
UNIVERSITÀ DI ROMA

# Data Mining Technology for Business and Society

Homework 3

Tran Luong Bang - *tran.1956419@studenti.uniroma1.it*

Juan Mata Naranjo - *matanaranjo.1939671@studenti.uniroma1.it*

## Introduction

The final goal of this homework is to process a sentence using deep learning techniques to decide whether such sentence is true (SUPPORTS) or false (REFUTES). To do so we first have to generate a set of features which can characterize the sentences such that we can train our classifiers. This featurization of the sentences has been done using two techniques: (i) Generating embeddings directly from the original sentence (part 1 – *sentence\_transformers*) and (ii) Extracting the relevant wikipedia pages from the sentence using the GENRE model and then constructing the embedding from some information of the Wikipedia page (part 2). Once we constructed the features which will allow us to classify our sentences we can go ahead and train our Binary Classifiers. For the purpose of this exercise, we have decided to use **Logistic Regression** (from now on also denoted as *Classifier 1*) and **Linear Support Vector Machine** (from now on also denoted as *Classifier 2*). It is relevant to highlight as well that given that our data set is unbalanced we have decided to use an under sampling approach to make our training set into a balanced one. We decided to use an under sampling technique to make the training phase more efficient (since we initially had computational troubles), leaving us a total of ~58k sentences to train on. The development sample consisted of 10444 observations and the test sample of 10100 observations.

## Part 1

We have looked for our ideal hyperparameters for both Binary Classifiers using the Grid Search approach, i.e. looking at the performance of our classifiers using different parameters by choosing the best performing one. The performance has been evaluated over the training data itself, reason for which we have also used a Cross Validator with 3 folds. To optimize computation time and therefore have the possibility to explore a wider range of hyperparameters we have used PySpark.

If the goal was to be sure to correctly catch all REFUTES we would look at the recall on REFUTES. The trivial solution is to classify all as REFUTES.

### Logistic Regression (Classifier 1)

The set of parameters explored are:

Hyperparameter	Configuration	Best Configuration	Interpretation
regParam	[0.01, 0.5]	0.01	Regularization Parameter
elasticNetParam	[0, 0.5, 1]	0.5	Elastic Net parameter, 0 means L2 penalty is used while 1 means L1 penalty is used
maxIter	[1, 3, 5, 10]	10	Maximum number of iterations to find minimum
tol	[1e-06]	1e-06	Tolerance level
threshold	[0.5]	0.5	Classification threshold over linear regression
fitIntercept	[True]	True	Fix intercept or not?

The code for this search is the following:

```
lr = LogisticRegression(labelCol='label', featuresCol='features')
paramGrid = (ParamGridBuilder())
```

```

        .addGrid(lr.regParam, [0.01, 0.5])
        .addGrid(lr.elasticNetParam, [0,0.5,1.0])
        .addGrid(lr.maxIter, [1,3,5,10])
        .build())
evaluator = BinaryClassificationEvaluator()
cv = CrossValidator(estimator=lr, estimatorParamMaps=paramGrid, evaluator = evaluator, numFolds=3 )

```

The confusion matrix, computed over the dev set and using the best configuration, is:

Classifier 1	Predicted Refutes	Predicted Supports	Recall
Actual Refutes	3549	1703	0.676
Actual Supports	1325	3867	0.745
Precision	0.728	0.694	0.710 (Accuracy)

## Linear Support Vector Machine (Classifier 2)

The set of parameters explored are:

Hyperparameter	Configuration	Best Configuration	Interpretation
regParam	[0.01, 0.5]	0.01	Regularization Parameter
maxIter	[1, 3, 5, 10]	10	Maximum number of iterations to find minimum
tol	[1e-06]	1e-06	Tolerance level
threshold	[0]	0	Threshold in binary classification applied to the linear model prediction
fitIntercept	[True]	True	Fix intercept or not?
standardization	[True]	True	Standardize data before fitting or not?

The code for this search is the following:

```

lscv = LinearSVC(labelCol='label', featuresCol='features')
paramGrid = (ParamGridBuilder()
        .addGrid(lscv.regParam, [0.01,0.5])
        .addGrid(lscv.maxIter, [1,3,5,10])
        .build())
evaluator = BinaryClassificationEvaluator()
cv = CrossValidator(estimator=lscv, estimatorParamMaps=paramGrid, evaluator = evaluator, numFolds=3)

```

For which the confusion matrix, computed over the dev set, is:

Classifier 2	Predicted Refutes	Predicted Supports	Recall
Actual Refutes	3373	1879	0.642
Actual Supports	1110	4082	0.786
Precision	0.752	0.685	0.714 (Accuracy)

The predictions results are saved in files *test\_set\_pred\_1.jsonl* and *test\_set\_pred\_2.jsonl*.

## Part 2

Similarly, to the previous part we have used a Grid Search approach combined with Cross Validation, and once again using PySpark to train our classifiers. The results for both classifiers can be found below:

### Logistic Regression (Classifier 1)

The set of parameters explored are:

Hyperparameter	Configuration	Best Configuration	Interpretation
regParam	[0.01, 0.5]	0.01	Regularization Parameter
elasticNetParam	[0, 0.5, 1]	0	Elastic Net parameter, 0 means L2 penalty is used while 1 means L1 penalty is used
maxIter	[1, 3, 5, 10, 100]	100	Maximum number of iterations to find minimum
tol	[1e-06]	1e-06	Tolerance level
threshold	[0.5]	0.5	Classification threshold over linear regression
fitIntercept	[True]	True	Fix intercept or not?

The code for this search is the following:

```
lr = LogisticRegression(labelCol='label', featuresCol='features')
paramGrid = (ParamGridBuilder()
              .addGrid(lr.regParam, [0.01, 0.5])
              .addGrid(lr.elasticNetParam, [0, 0.5, 1.0])
              .addGrid(lr.maxIter, [1, 3, 5, 10, 100])
              .build())
evaluator = BinaryClassificationEvaluator()
cv = CrossValidator(estimator=lr, estimatorParamMaps=paramGrid, evaluator = evaluator, numFolds=3 )
```

The confusion matrix, computed over the dev set and using the best configuration, is:

Classifier 1	Predicted Refutes	Predicted Supports	Recall
Actual Refutes	3763	1489	0.716
Actual Supports	1342	3850	0.742
Precision	0.737	0.721	0.729 (Accuracy)

### Linear Support Vector Machine (Classifier 2)

The set of parameters explored are:

Hyperparameter	Configuration	Best Configuration	Interpretation
regParam	[0.01, .5]	0.01	Regularization Parameter
maxIter	[1, 3, 5, 10, 100]	100	Maximum number of iterations to find minimum
tol	[1e-06]	1e-06	Tolerance level
threshold	[0]	0	Threshold in binary classification applied to the linear model prediction
fitIntercept	[True]	True	Fix intercept or not?
standardization	[True]	True	Standardize data before fitting or not?
regParam	[0.01, .5]	.01	Regularization Parameter

The code for this search is the following:

```
lscv = LinearSVC(labelCol='label', featuresCol='features')
paramGrid = (ParamGridBuilder()
              .addGrid(lscv.regParam, [0.01, 0.5])
              .addGrid(lscv.maxIter, [1, 3, 5, 10, 100])
              .build())
evaluator = BinaryClassificationEvaluator()
cv = CrossValidator(estimator=lscv, estimatorParamMaps=paramGrid, evaluator = evaluator, numFolds=3 )
```



The confusion matrix, computed over the dev set and using the best configuration, is:

Classifier 2	Predicted Refutes	Predicted Supports	Recall
Actual Refutes	3680	1572	0.701
Actual Supports	1260	3932	0.757
Precision	0.745	0714	0.729 (Accuracy)

The predictions results are saved in files *new\_test\_set\_pred\_1.jsonl* and *new\_test\_set\_pred\_2.jsonl*.

## Bonus Point

The test set fill we have submitted was: *new\_test\_set\_pred\_1.jsonl* which was trained using a Logistic Regression using the second processing approach (GENRE+sentence\_transformer).

Rank	Participant team	R-Prec	Recall@5	Accuracy	KILT-AC	Last submission at
1	multitaskdpr (Multitask DPR + BART)	74.48	87.52	86.32	63.94	6 months ago
2	Host_23415_Team (BERT + DPR) 	72.93	73.52	69.68	58.58	9 months ago
3	kilt (RAG)	61.94	75.55	86.31	53.45	9 months ago
4	Host_23415_Team (BART + DPR) 	55.33	74.29	86.74	47.68	9 months ago
5	Host_23415_Team (NSMN) 	49.24	70.16	66.10	41.88	9 months ago
6	stupidTeam	0.00	0.00	69.71	0.00	4 minutes ago
7	JuanTran	0.00	0.00	71.38	0.00	11 minutes ago
8	Alessandro_Tansel	0.00	0.00	71.42	0.00	3 hours ago
9	Marco Aurelio Sterpa	0.00	0.00	67.98	0.00	3 hours ago