

# 1 Algorithm

---

## Algorithm 1: SVM Baseline

---

**Input:** Files (Files train and test), tp (Tunned params), cv (k-Folds Cross Validation), metric (Metric), df (Default paramns LBD)

```

1 forall fold ∈ Files do
2   x_train, y_train, x_test, y_test = fold;
3   model = GridSearch(svn.SVC(df), tp, cv, metric);
4   model.fit(x_train, y_train);
5   y_pred = model.predict(x_test);
6   f1_macro.add(f1_score(y_test, y_pred, "f1_macro"));
7   f1_micro.add(f1_score(y_test, y_pred, "f1_micro"));
8 return mean(f1_macro), sd(f1_macro), mean(f1_micro), sd(f1_micro);

```

---

O algoritmo SVM tem complexidade  $O(n\_samples^3 \times n\_features)^1$ . Utilizando o *TFIDF* do dataset foi possível obter a informação de samples (linhas) e features (colunas), a Tabela 1 apresenta o impacto da entrada no algoritmo SVM. Essas informações foram obtidas do fold 0, neste caso será necessário calcular o valor para os demais folds (4).

Dataset	Samples	Features	SVM
Stanford Tweets	286	1333	31183743448
20NG	15071	98230	$3,362562409 \times 10^{17}$
ACM	19914	48919	$3,86799071 \times 10^{17}$

Table 1: Results for dataset **Stanford Tweets**.

## 2 Experiments

Methods	Macro F1	Micro F1	Time (s)
Metalazy artigo	83.81 (7.52)	83.86 (7.51)	
SVM Kernel linear	79.01 (5.14)	79.09 (5.12)	<b>1.17</b>
SVM Kernel rbf	79.84 (5.37)	79.94 (5.38)	1.31
Metalazy com SVM linear	80.71 (0.55)	80.49 (0.26)	56.65
Metalazy com SVM rbf	80.71 (0.55)	80.77 (0.51)	57.41

Table 2: Results for dataset **Stanford Tweets**.

---

<sup>1</sup><https://scikit-learn.org/stable/modules/svm.html#complexity>

<b>Methods</b>	<b>Macro F1</b>	<b>Micro F1</b>	<b>Time (s)</b>
Metalazy artigo	()	()	
SVM Kernel linear	()	()	
SVM Kernel rbf	()	()	
Metalazy com SVM linear	()	()	
Metalazy com SVM rbf	()	()	

Table 3: Results for dataset **Reut**.