ANÀLISI DE SENTIMENT DE PEL·LÍCULES

CONSTRUCCIÓ D'UN CLASSIFICADOR DE SENTIMENTS EN RESSENYES DE PEL·LÍCULES AMB SCIKIT-LEARN I PYTHON

TÍTOL
OFICIAL
DE PRECISIÓ
SOVIÈTICA

Data i hora
12 de desembre
MMXXIII
12:34:22

Agenda

INTRODUCCIÓ

DATASET

PREPROCESSAMENT

ANÀLISI

CONCLUSIONS

INTRO

INTRO

dataset molt polit a kaggle

INTRO

scikit-learn és guai

dataset molt polit a kaggle

INTRO

La Construcció d'un Classificador de Sentiments en Ressenyes de Pel·lícules.

Un model de **machine learning** amb la llibreria **scikit-learn** a Python per predir si una ressenya de pel·lícula és positiva o negativa a partir de la interpretació del text.

DATASET

https://www.kaggle.com/code/lakshmi25npathi/sentiment-analysis-of-imdb-movie-reviews

Gràcies Kaggle!

S'ha fet servir un conjunt de dades de l'IMDB amb **50.000 ressenyes de pel·lícules** disponible a Kaggle.

Try Pitch

```
df = pd.read_csv('kaggle_movies_db.csv')
df
```

	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive
...	...	...
95	I thought this movie did a down right good job...	positive
96	Bad plot, bad dialogue, bad acting, idiotic di...	negative
97	I am a Catholic taught in parochial elementary...	negative
98	I'm going to have to disagree with the previou...	negative
99	No one expects the Star Trek movies to be high...	negative

00 rows × 2 columns

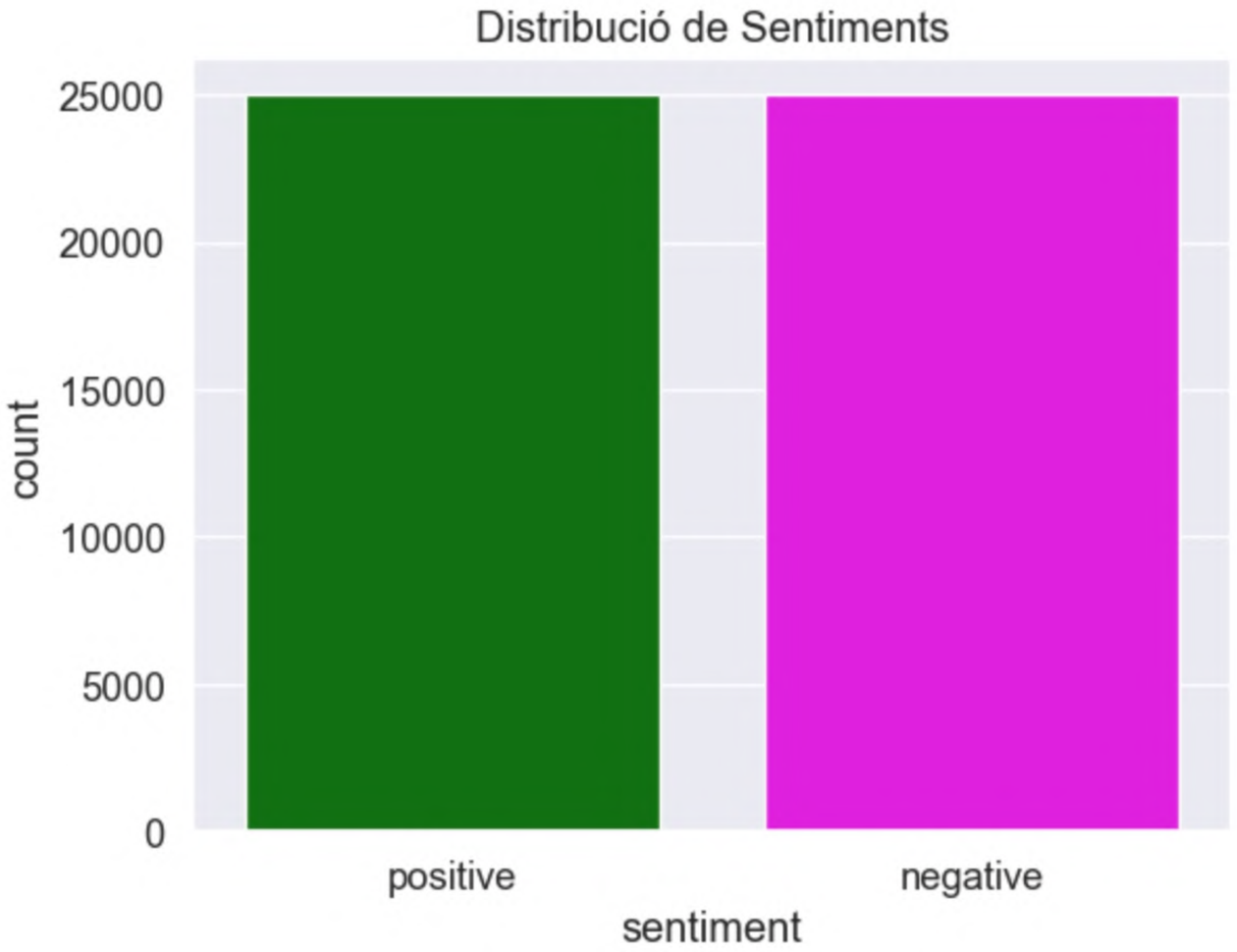
ntinuació, explorem les dades del dataset

```
df.describe()
```

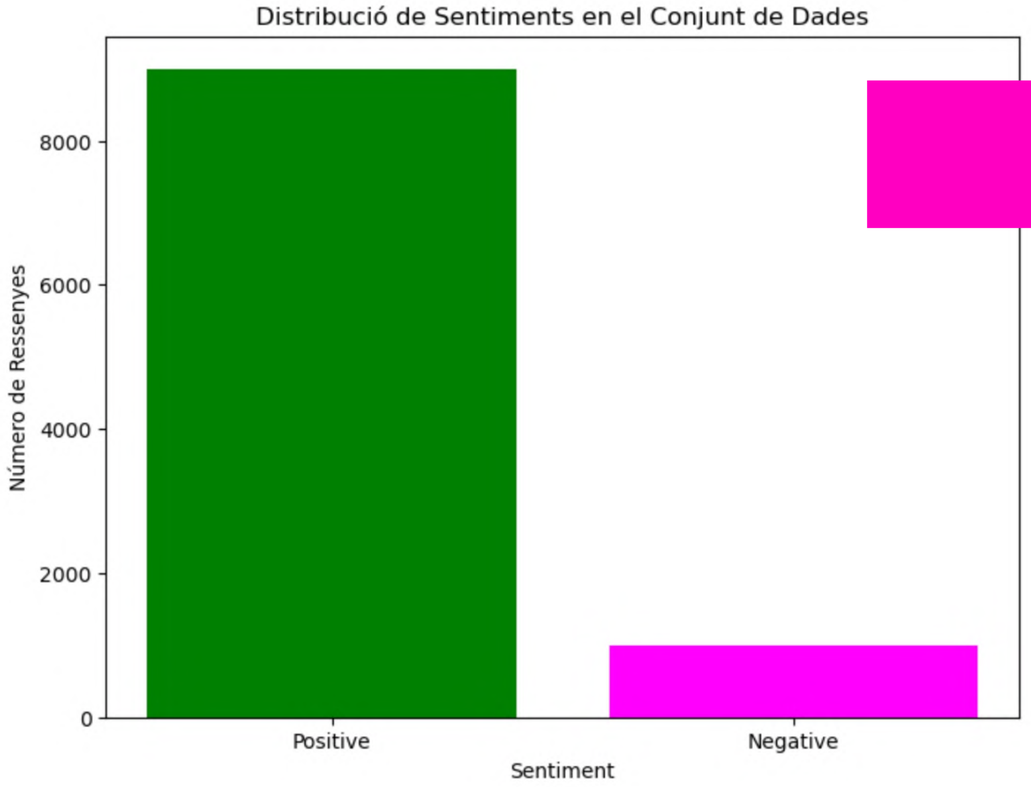
	review	sentiment
unt	50000	50000
que	49582	2
top	Loved today's show!!! It was a variety and not...	positive
freq	5	25000

DATASET

https://www.kaggle.com/code/lakshmi25npathi/sentiment-analysis-of-imdb-movie-reviews



PREPROCESSAMENT



imbalanced-learn documentation

Date: Jul 08, 2023 **Version**: 0.11.0

Useful links: Binary Installers | Source Repository | Issues & Ideas | Q&A Support

Imbalanced-learn (imported as `imblearn`) is an open source, MIT-licensed library relying on scikit-learn (imported as `sklearn`) and provides tools when dealing with classification with imbalanced classes.

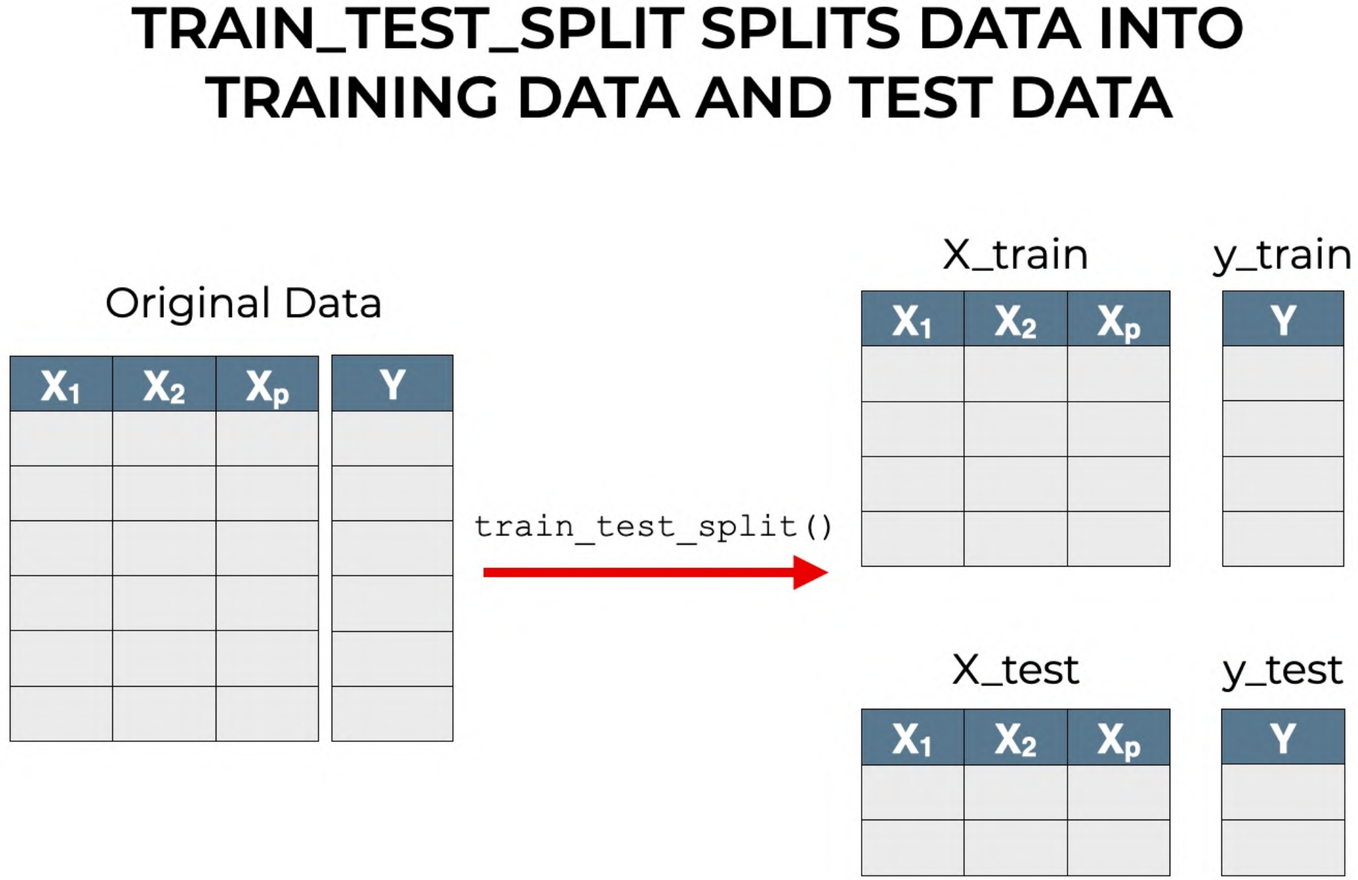
```
sentiment
negative     1000
positive     1000
dtype: int64
```

ANÀLISI: SCIKIT LEARN

```
from imblearn.under_sampling import RandomUnderSampler
from sklearn.model_selection import train_test_split

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import SVC

from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import LogisticRegression

from sklearn.metrics import f1_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix

from sklearn.model_selection import GridSearchCV
```

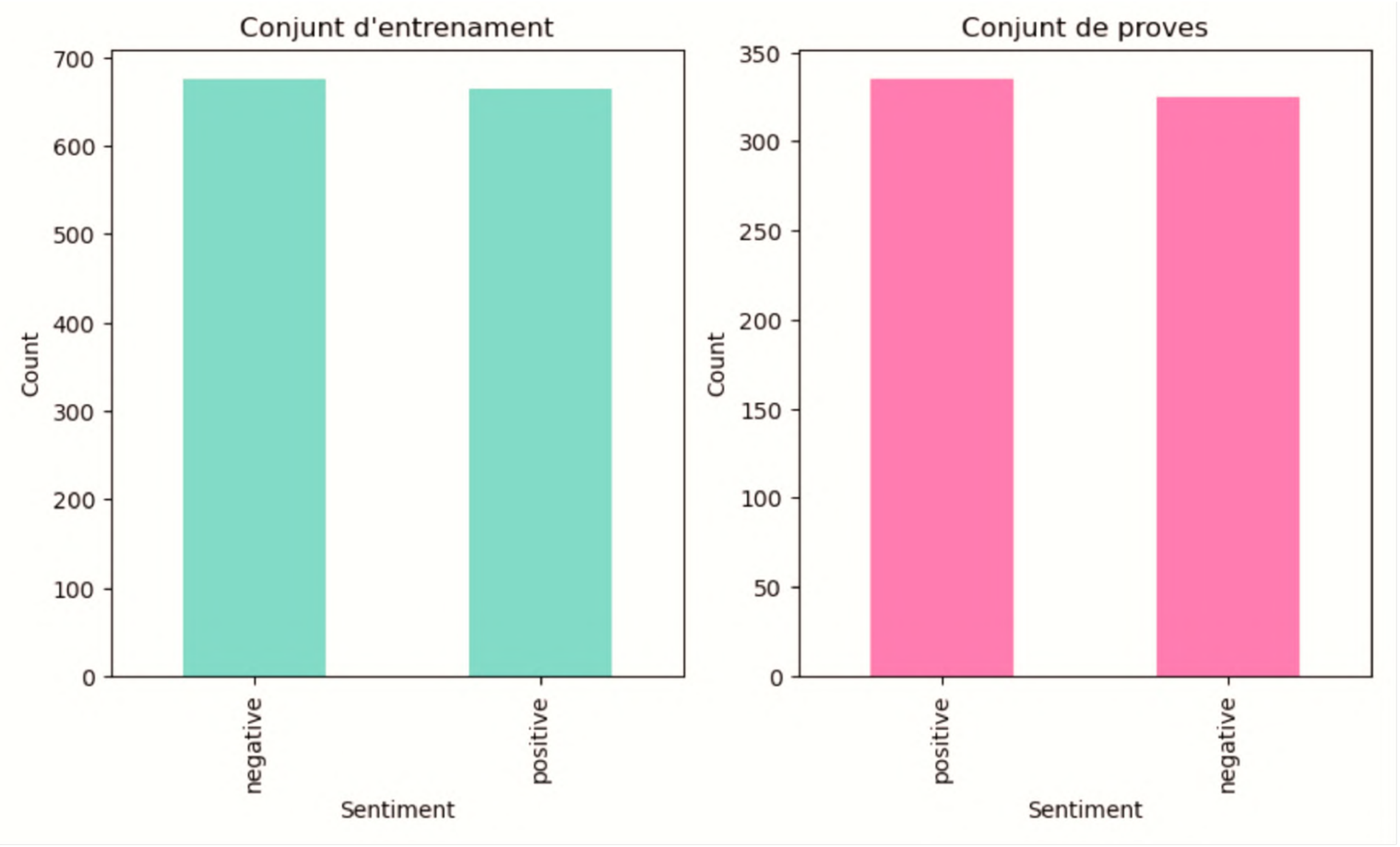

Font: https://www.sharpsightlabs.com/blog/scikit-train_test_split/

ANÀLISI: TRAIN TEST SPLIT



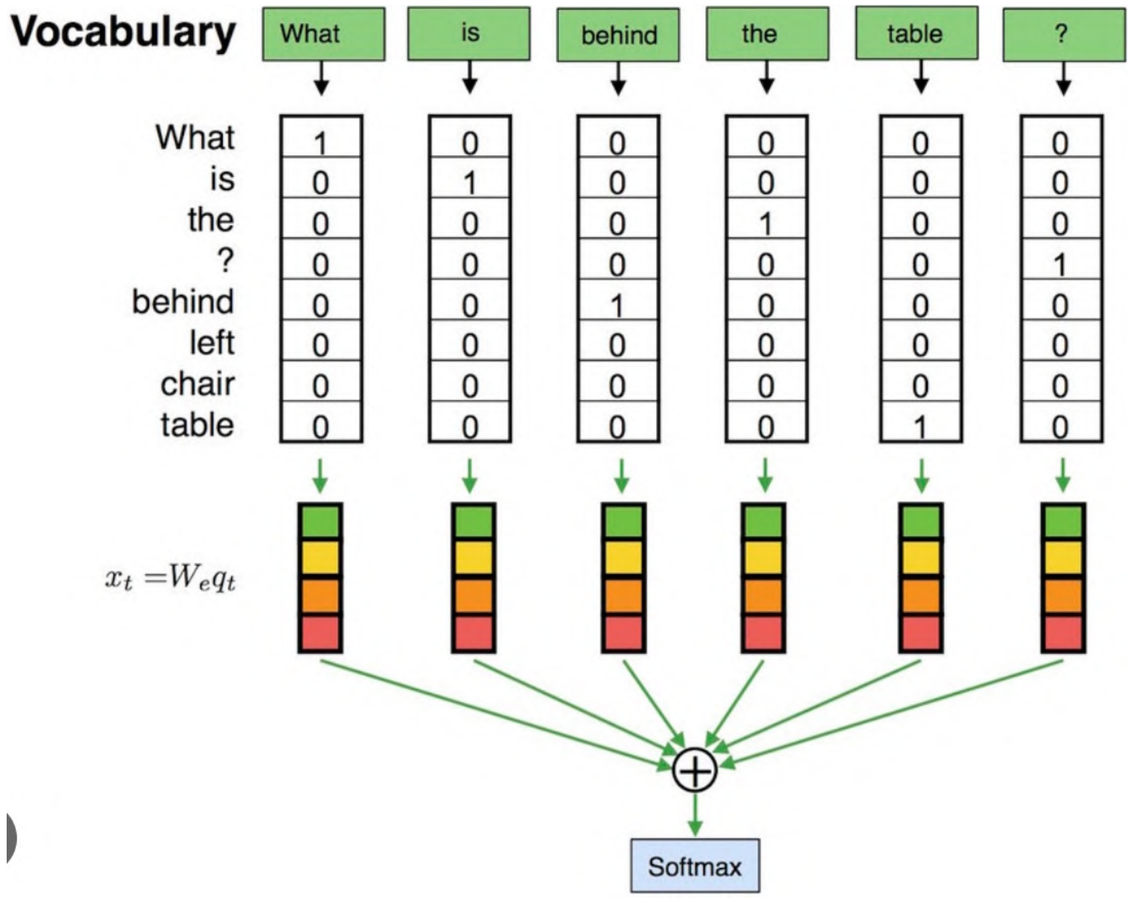
Font: https://www.sharpsightlabs.com/blog/scikit-train_test_split/

ANÀLISI: TRAIN TEST SPLIT



Font: https://www.sharpsightlabs.com/blog/scikit-train_test_split/

ANÀLISI: BOW (BAG OF WORDS)



Font: https://www.researchgate.net/figure/Bag-Of-Words-BOW-representation-of-the-input-that-is-next-follow-by-Softmax_fig2_308884219

ANÀLISI: APRENENTATGE SUPERVISAT

01

- Support Vector Machines (SVM)

02

- Decision Tree

03

- Naive Bayes

04

- Logistic Regression

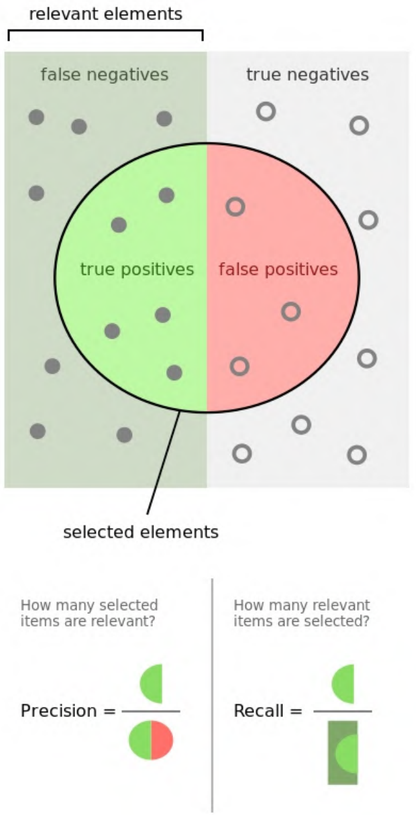
05

- Avaluació del Model

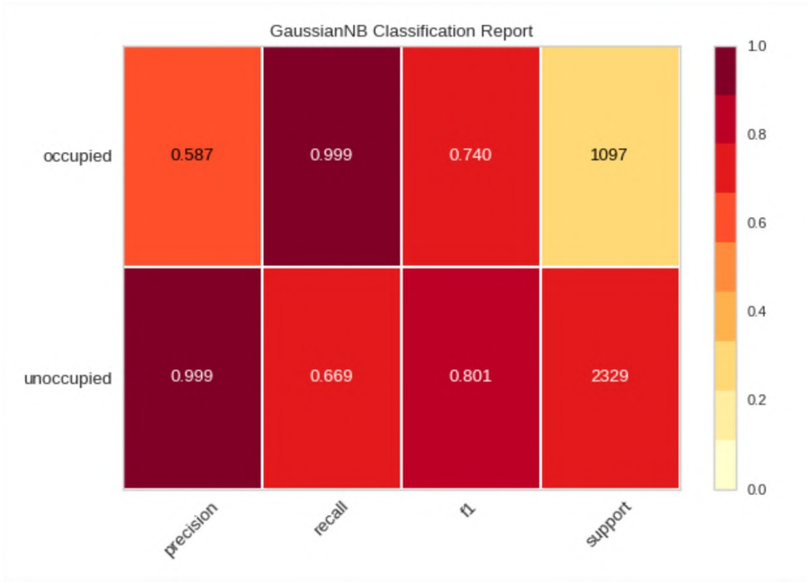

```python
# Avalua el rendiment del model SVM
svc_score = svc.score(test_x_tfidf, test_y)
print(f"Rendiment del model SVM: {round(svc_score, 2)}")

# Avalua el rendiment del model Decision Tree
dec_tree_score = dec_tree.score(test_x_tfidf, test_y)
print(f"Rendiment del model Decision Tree: {round(dec_tree_score, 2)}")

# Avalua el rendiment del model Naive Bayes
gnb_score = gnb.score(test_x_tfidf.toarray(), test_y)
print(f"Rendiment del model Naive Bayes: {round(gnb_score, 2)}")

# Avalua el rendiment del model Logistic Regression
log_reg_score = log_reg.score(test_x_tfidf, test_y)
print(f"Rendiment del model Logistic Regression: {log_reg_score}")

```

```
Rendiment del model SVM: 0.85
Rendiment del model Decision Tree: 0.51
Rendiment del model Naive Bayes: 0.63
Rendiment del model Logistic Regression: 0.85
```

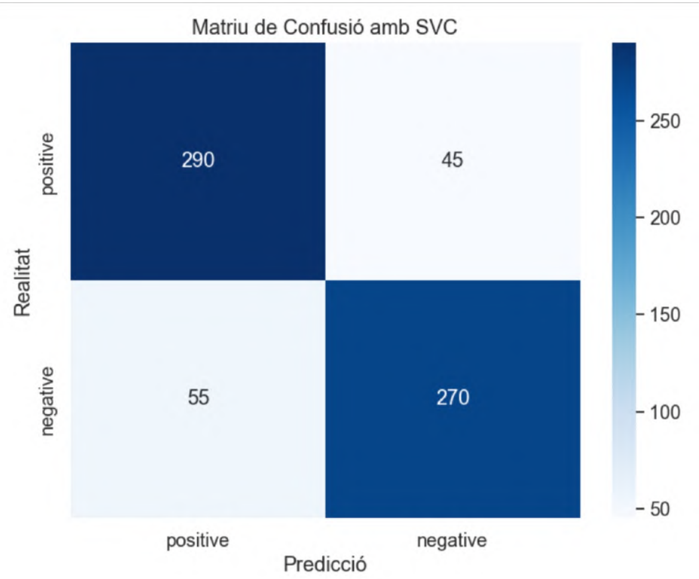
F1 Score



Classification Report



Confusion Matrix



ANÀLISI: AVALUACIÓ DEL MODEL SEGONS SVM (SUPPORT VECTORS MACHINE)

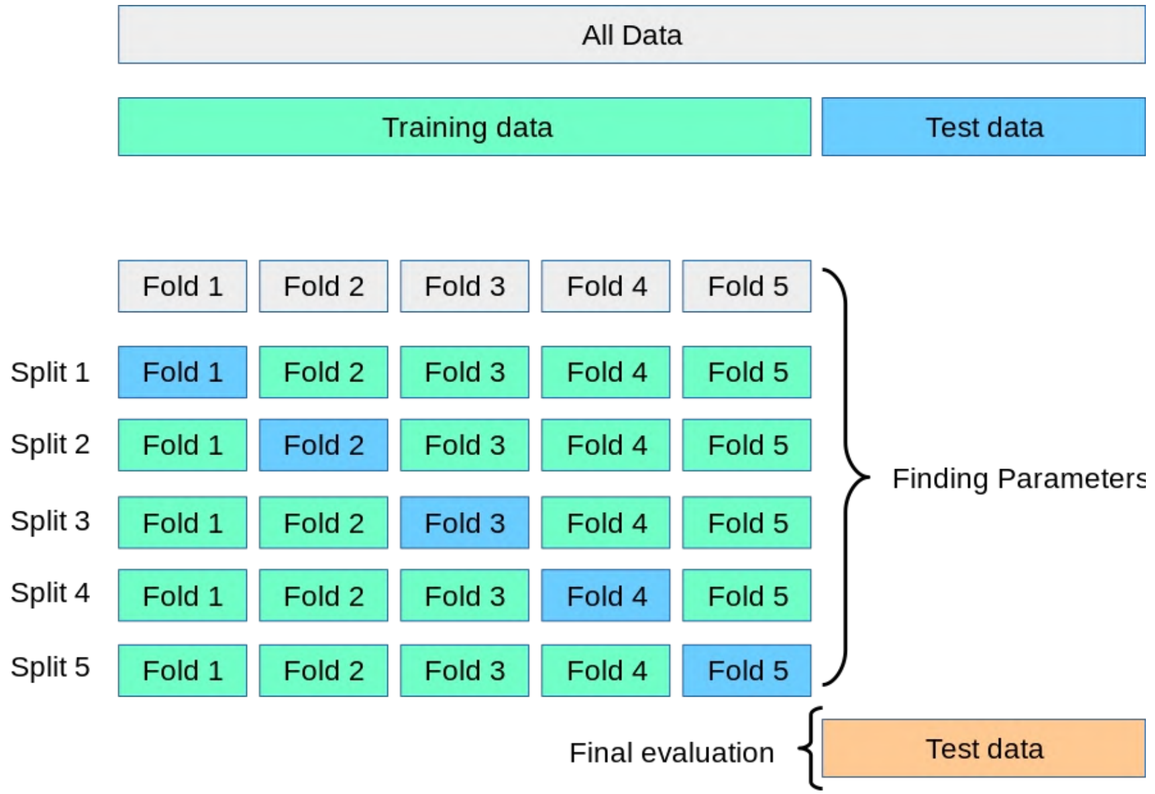
Classification Report

Informe de classificació amb SVC:

	precision	recall	f1-score	support
positive	0.84	0.87	0.85	335
negative	0.86	0.83	0.84	325
accuracy			0.85	660
macro avg	0.85	0.85	0.85	660
weighted avg	0.85	0.85	0.85	660

ANÀLISI: AVALUACIÓ DEL MODEL SEGONS SVM (SUPPORT VECTORS MACHINE)

AFINACIÓ DEL MODEL: GRID SEARCH CV

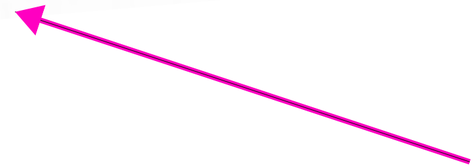


CONCLUSIÓ

```python
import time

# Inicia el rellotge
start_time = time.time()

# Resta el temps actual al temps d'inici per obtenir el temps d'inici
svc_grid = GridSearchCV(svc, parameters, cv=5)
svc_grid.fit(train_x_vector, train_y)

# Atura el rellotge
end_time = time.time()

# Calcula la diferència entre el temps d'inici i el temps final
execution_time = end_time - start_time

# Mostra els paràmetres òptims
print("Paràmetres òptims:", svc_grid.best_params_)

# Mostra el temps d'execució
print("Temps d'execució:", execution_time, "segons")

```

```
Paràmetres òptims: {'C': 4, 'kernel': 'rbf'}
Temps d'execució: 35.86082625389099 segons
```

Exactitud del model amb paràmetres optimitzats: 0.8560606060606061



```python
# Millors paràmetres obtinguts de la cerca de paràmetres
best_params = {'C': 4, 'kernel': 'rbf'}

# Crea una nova instància de SVC amb els millors paràmetres
optimized_svc = SVC(**best_params)

# Entrena el model amb les dades d'entrenament
optimized_svc.fit(train_x_vector, train_y)

# Fes prediccions amb les dades de prova
predictions = optimized_svc.predict(test_x_vector)

# Avalua el rendiment del model
accuracy = optimized_svc.score(test_x_vector, test_y)
print(f"Exactitud del model amb paràmetres optimitzats: {accuracy}")

```

```
Exactitud del model amb paràmetres optimitzats: 0.8560606060606061
```

TENGO PRUEBAS!

```python
# Prediccions utilitzant el model SVM entrenat
prediction_1 = svc.predict(tfidf_vectorizer.transform(['A great film']))
prediction_2 = svc.predict(tfidf_vectorizer.transform(['A nice film but it could be better']))
prediction_3 = svc.predict(tfidf_vectorizer.transform(['A nice film']))
prediction_4 = svc.predict(tfidf_vectorizer.transform(['I was going to say something good, but I simply cannot be

# Mostrem les prediccions
print("Predicció per 'A great film':", prediction_1)
print("Predicció per 'A nice film but it could be better':", prediction_2)
print("Predicció per 'A nice film':", prediction_3)
print("Predicció per 'I was going to say something good, but I simply cannot because the film is so bad.':", prec
```

```
Predicció per 'A great film': ['positive']
Predicció per 'A nice film but it could be better': ['negative']
Predicció per 'A nice film': ['positive']
Predicció per 'I was going to say something good, but I simply cannot because the film is so bad.': ['negative']
```

Les conclusions del model són positives: fins ara, ha encertat tot.

CONCLUSIÓ

Desequilibri de les classes:
Ha permés l'aplicació de tècniques com l'undersampling per abordar aquesta qüestió.

Importància de la representativitat en la divisió de les dades:
La divisió aleatòria de les dades ajuda a evitar l'overfitting del model.

Aprenentatge supervisat:
Support Vector Machines (SVM) i Logistic Regression han presentat els millors resultats, acurats en un 85%.

Avaluació del model:
F1 Score, Classification Report i Confusion Matrix, i finalment la maximització amb GridSearchCV, han arribat a una precisió del un 85,6 %.

Reptes i millores:
Tècniques avançades de processament de llenguatge natural i cross-validation per avaluacions més robustes.

Contacting? Pones edaimon en google y ya te salgo.

THE END

FIN

Pitch

Want to make a presentation like this one?

Start with a fully customizable template, create a beautiful deck in minutes, then easily share it with anyone.

Create a presentation (It's free)