COMPARACIÓN DE MODELOS CON DATASET LEMMATIZADO Y STEMMIZADO

Lemma – Naive Bayes

Naive Bayes - Modelo Benchmark

```
from sklearn.naive_bayes import GaussianNB
nb = GaussianNB()
nb.fit(xtrain,ytrain)

y_pred = nb.predict(xtest)
print("accuracy : ", accuracy_score(ytest,y_pred))
```

```
accuracy :  0.41009523809523807
```

```
print(color.BOLD + 'Reporte de clasificación : '+ color.
```

```
Reporte de clasificación :
```

	precision	recall	f1-score	support
1	0.45	0.72	0.55	12600
2	0.35	0.26	0.30	12600
3	0.35	0.19	0.25	12600
4	0.38	0.22	0.28	12600
5	0.43	0.67	0.52	12600
accuracy			0.41	63000
macro avg	0.39	0.41	0.38	63000
weighted avg	0.39	0.41	0.38	63000

Stem – Naive Bayes

Naive Bayes - Modelo Benchmark

```
from sklearn.naive_bayes import GaussianNB
nb = GaussianNB()
nb.fit(xtrain,ytrain)

y_pred = nb.predict(xtest)
print("accuracy : ", accuracy_score(ytest,y_pred))
```

```
accuracy :  0.4178095238095238
```

```
print(color.BOLD + 'Reporte de clasificación : '+ color.
```

```
Reporte de clasificación :
```

	precision	recall	f1-score	support
1	0.49	0.68	0.57	12600
2	0.36	0.27	0.31	12600
3	0.35	0.22	0.27	12600
4	0.37	0.22	0.27	12600
5	0.42	0.70	0.53	12600
accuracy			0.42	63000
macro avg	0.40	0.42	0.39	63000
weighted avg	0.40	0.42	0.39	63000

COMPARACIÓN DE MODELOS CON DATASET LEMMATIZADO Y STEMMIZADO

Lemma – Random Forest

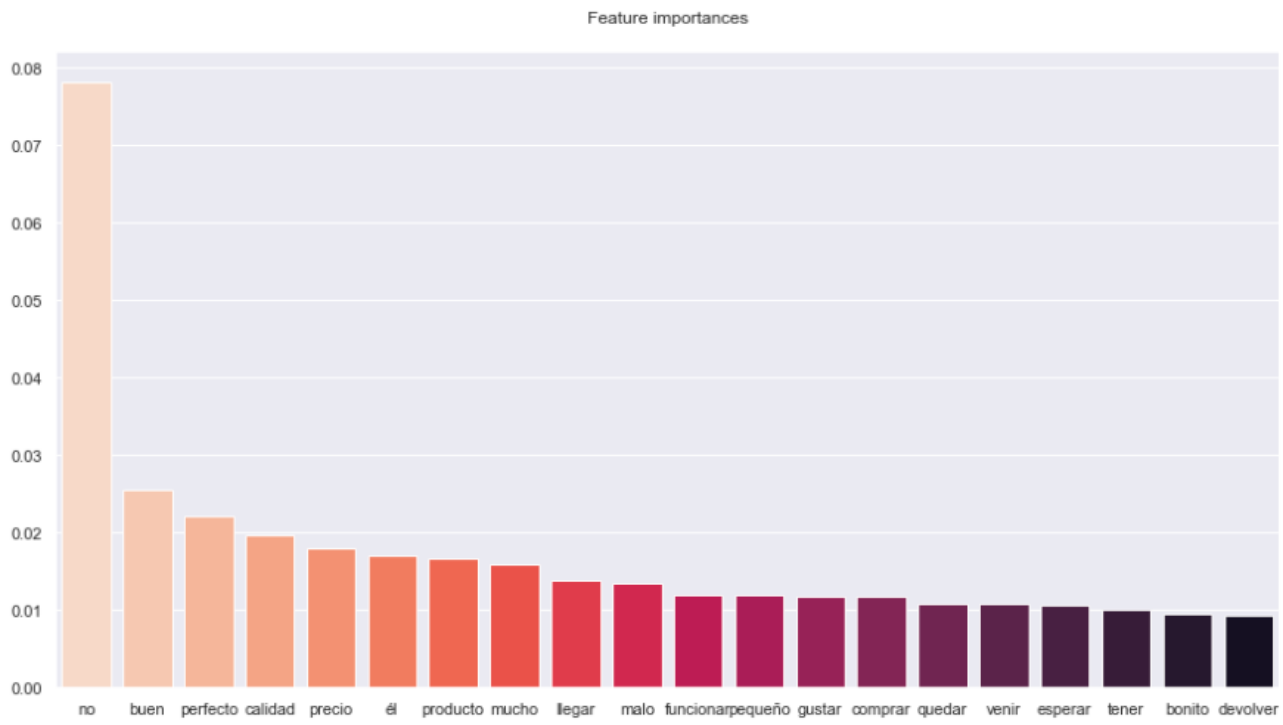
Random Forest

```
rf = RandomForestClassifier()
rf.fit(xtrain,ytrain)

y_pred = rf.predict(xtest)
print("accuracy : ", accuracy_score(ytest, y_pred))
```

```
accuracy :  0.4451428571428571
```

```
print(color.BOLD + 'Reporte de clasificación : '+ color.E
```

```
Reporte de clasificación :

              precision    recall  f1-score   support

           1       0.54      0.62      0.58     12600
           2       0.37      0.36      0.37     12600
           3       0.34      0.31      0.33     12600
           4       0.39      0.35      0.37     12600
           5       0.53      0.58      0.56     12600

    accuracy                           0.45     63000
   macro avg       0.44      0.45      0.44     63000
weighted avg       0.44      0.45      0.44     63000
```



COMPARACIÓN DE MODELOS CON DATASET LEMMATIZADO Y STEMMIZADO

Stem – Random Forest

Random Forest

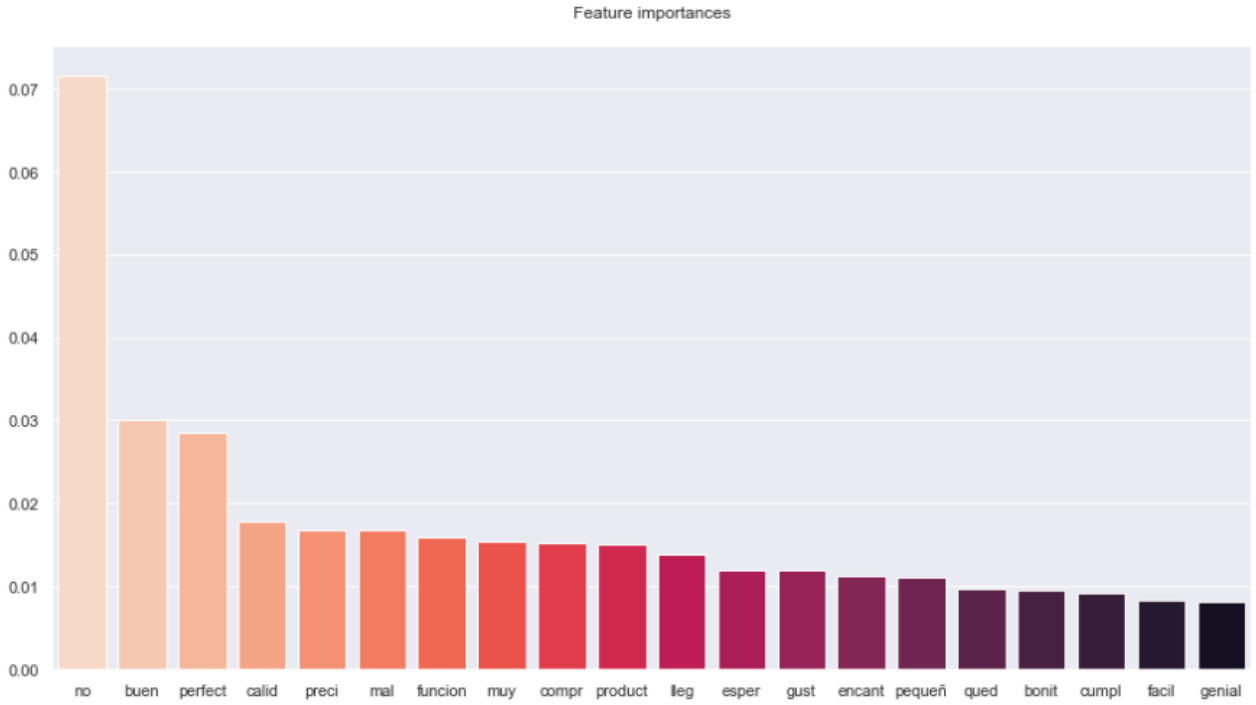
```
rf = RandomForestClassifier()
rf.fit(xtrain,ytrain)

y_pred = rf.predict(xtest)
print("accuracy : ", accuracy_score(ytest, y_pred))
```

```
accuracy :  0.45941269841269844
```

```
print(color.BOLD + 'Reporte de clasificación : '+ color
```

```
Reporte de clasificación :
```

	precision	recall	f1-score	support
1	0.56	0.64	0.60	12600
2	0.38	0.37	0.37	12600
3	0.36	0.31	0.33	12600
4	0.41	0.36	0.38	12600
5	0.56	0.61	0.58	12600
accuracy			0.46	63000
macro avg	0.45	0.46	0.45	63000
weighted avg	0.45	0.46	0.45	63000



COMPARACIÓN DE MODELOS CON DATASET LEMMATIZADO Y STEMMIZADO

Lemma - SVM

SVM

```
# En vez de utilizar SVC, vamos a usar LinearSVC,
# ya que para el Kernel Lineal esta función es MUCHO ma
from sklearn.svm import LinearSVC

svc = LinearSVC(C = 1)
svc.fit(xtrain,ytrain)

y_pred = svc.predict(xtest)
print("accuracy : ", accuracy_score(ytest, y_pred))
```

```
accuracy :  0.4608888888888889
```

```
print(color.BOLD + 'Reporte de clasificación : '+ color
```

```
Reporte de clasificación :

              precision    recall  f1-score   support

           1       0.52      0.71      0.60     12600
           2       0.39      0.32      0.35     12600
           3       0.37      0.28      0.32     12600
           4       0.43      0.33      0.38     12600
           5       0.52      0.66      0.58     12600

    accuracy                           0.46     63000
   macro avg       0.45      0.46      0.45     63000
weighted avg       0.45      0.46      0.45     63000
```

Stem - SVM

SVM

```
# En vez de utilizar SVC, vamos a usar LinearSVC,
# ya que para el Kernel Lineal esta función es MUCHO mas
from sklearn.svm import LinearSVC

svc = LinearSVC(C = 1)
svc.fit(xtrain,ytrain)

y_pred = svc.predict(xtest)
print("accuracy : ", accuracy_score(ytest, y_pred))
```

```
accuracy :  0.4737619047619048
```

```
print(color.BOLD + 'Reporte de clasificación : '+ color.
```

```
Reporte de clasificación :

              precision    recall  f1-score   support

           1       0.53      0.72      0.61     12600
           2       0.41      0.32      0.36     12600
           3       0.38      0.29      0.33     12600
           4       0.43      0.35      0.39     12600
           5       0.54      0.68      0.60     12600

    accuracy                           0.47     63000
   macro avg       0.46      0.47      0.46     63000
weighted avg       0.46      0.47      0.46     63000
```