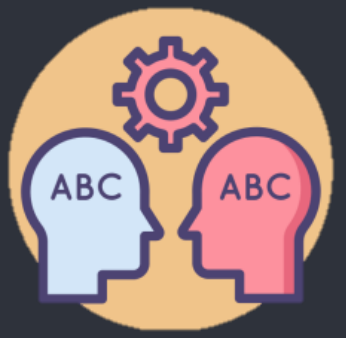
Procesamiento de Lenguaje Natural{



[NLP]

<Aprendizaje Supervisado | Grid Search & Cross-validation>

}

Ivan Rojas Gonzalez – Universidad Panamericana

La Agenda de hoy {

01 Que es Grid Search?

<Cómo funciona>

02 Que es Cross Validation?

<Cómo funciona>

03 Gridsearch & Cross Validation

<El futuro es hoy, oiste viejo?>

}

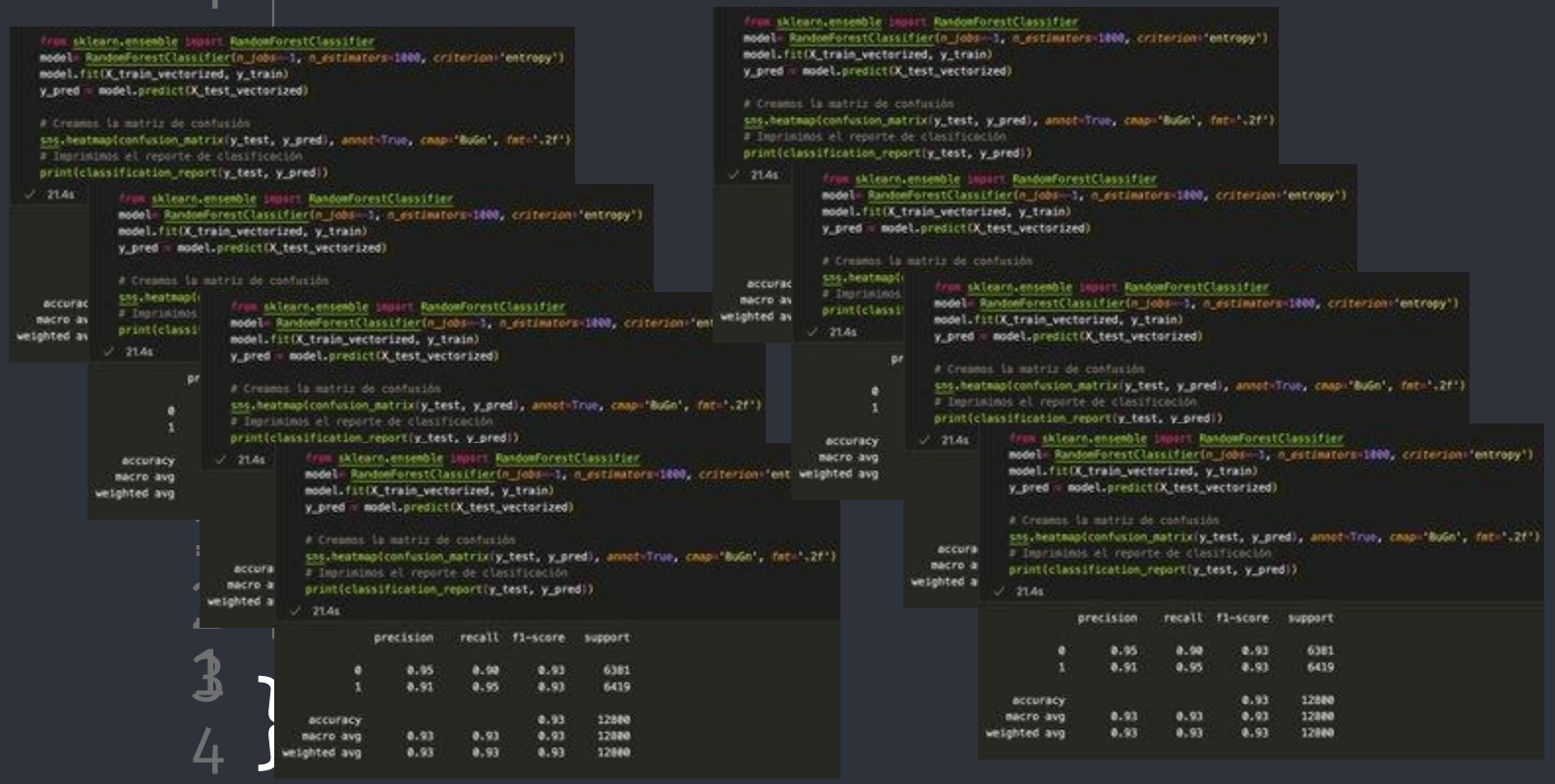
{Grid Search}

Ivan Rojas Gonzalez – Universidad Panamericana

Grid Search{

<Hasta ahora, cada que queremos probar una nueva combinacion de hiperparametros terminamos creando un modelo nuevo a mano>

}



Grid Search{

<Y si hubiera una forma de automatizar ese proceso?>

}

Grid Search{

<Gridsearch realiza una busqueda exhaustiva entre las posibles combinaciones de hiperparametros para determinar que combinacion da los mejores resultados>

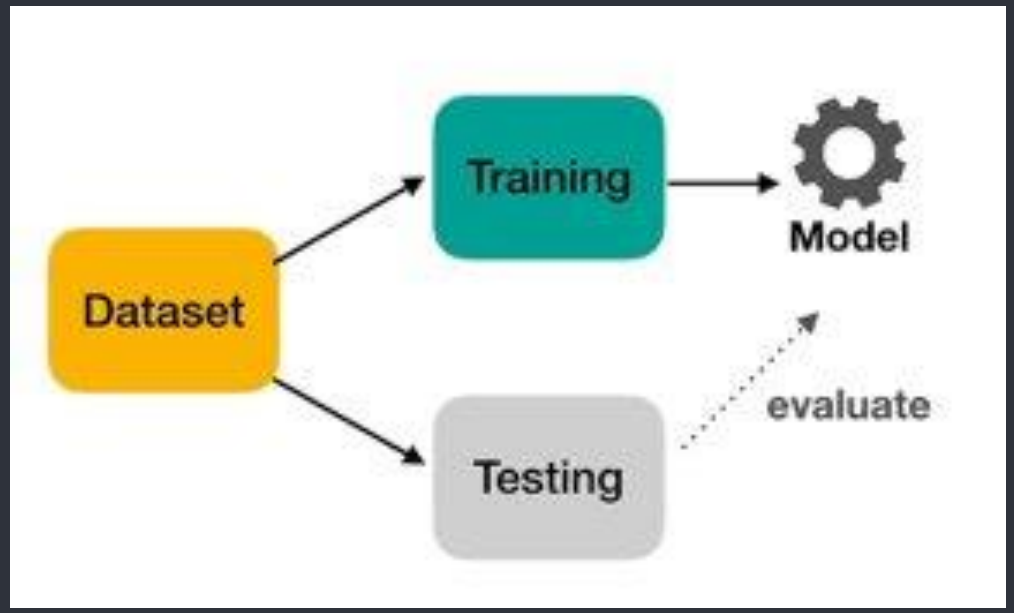
}

{Cross Validation}

Ivan Rojas Gonzalez – Universidad Panamericana

Cross Validation{

<Hasta ahora, al construir modelos hacemos lo siguiente: >

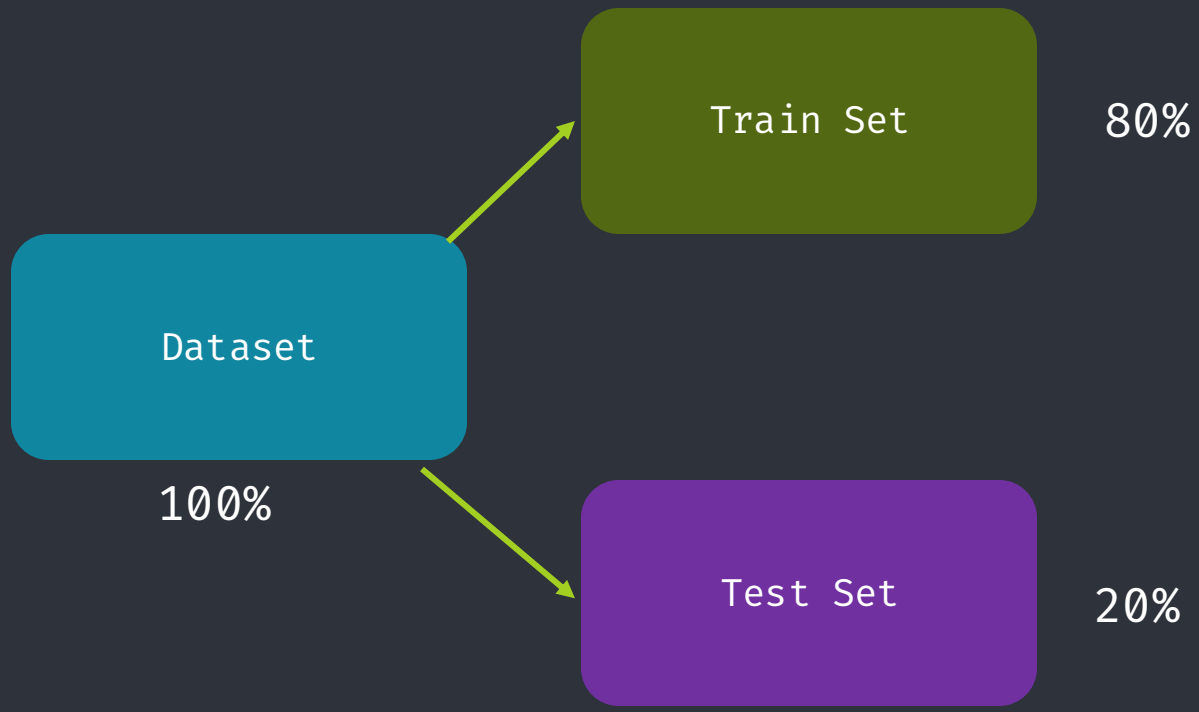


<Debemos esperar a que el modelo termine de entrenar para poder evaluar >

}

Cross Validation{

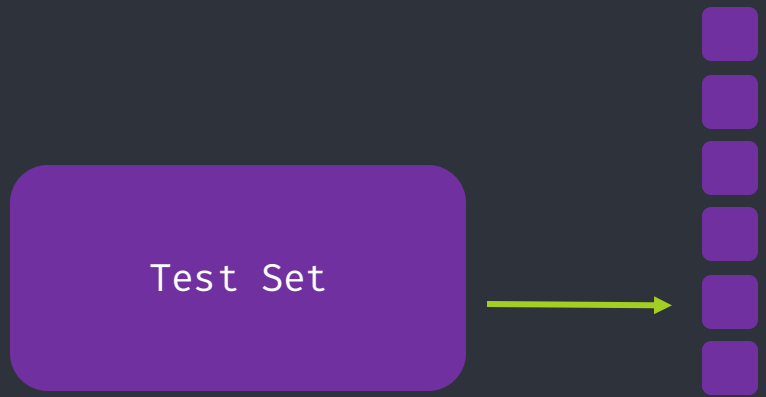
<Con cross-validation podemos evaluar desempeño durante entrenamiento>



}

Cross Validation{

<Con cross-validation podemos evaluar desempeño durante entrenamiento>



El 100% de ese 20%

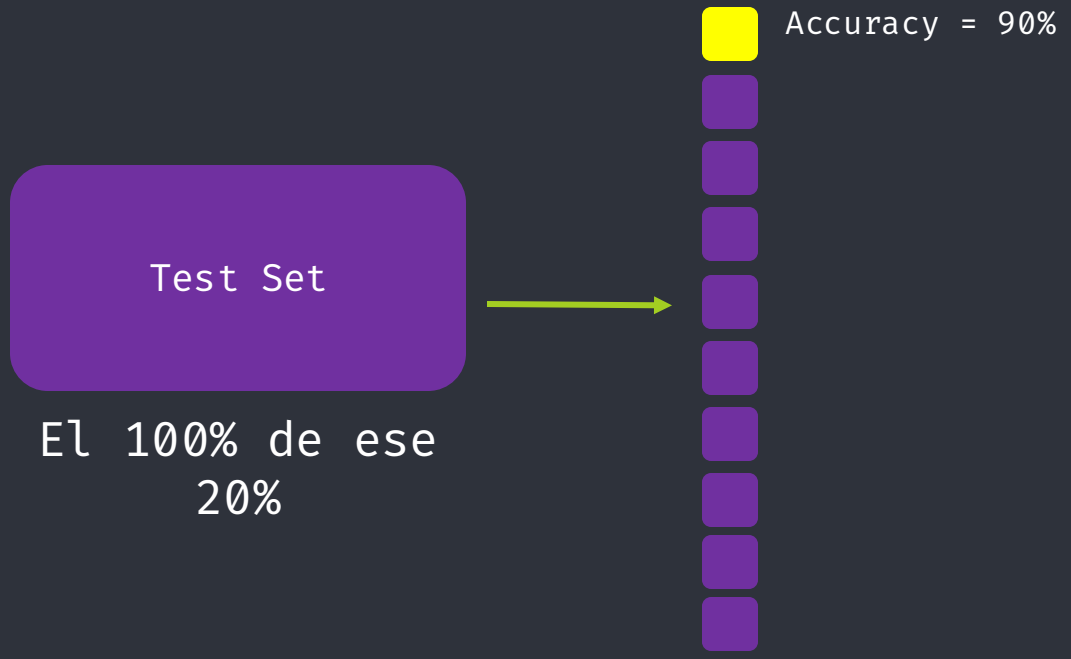
Es dividido en k-folds (un valor tipico es 10)

Y un modelo con una combinacion determinada de hiperparametros es evaluada con un fold a la vez

}

Cross Validation{

<Con cross-validation podemos evaluar desempeño durante entrenamiento>

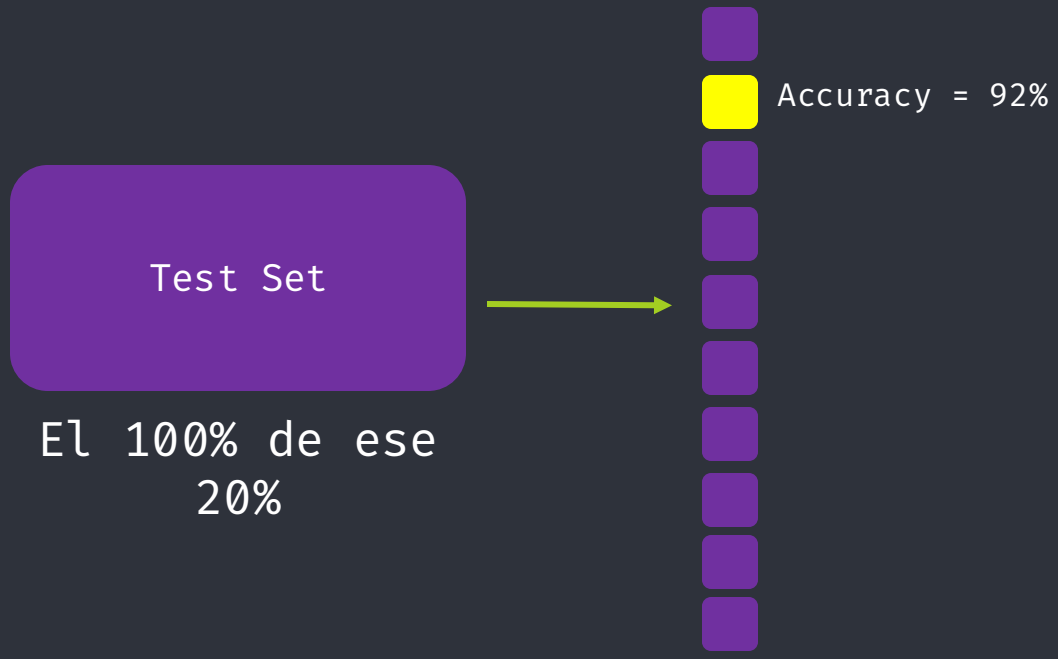


Y un modelo con una combinacion determinada de hiperparametros es evaluada con un fold a la vez

}

Cross Validation{

<Con cross-validation podemos evaluar desempeño durante entrenamiento>



El 100% de ese 20%

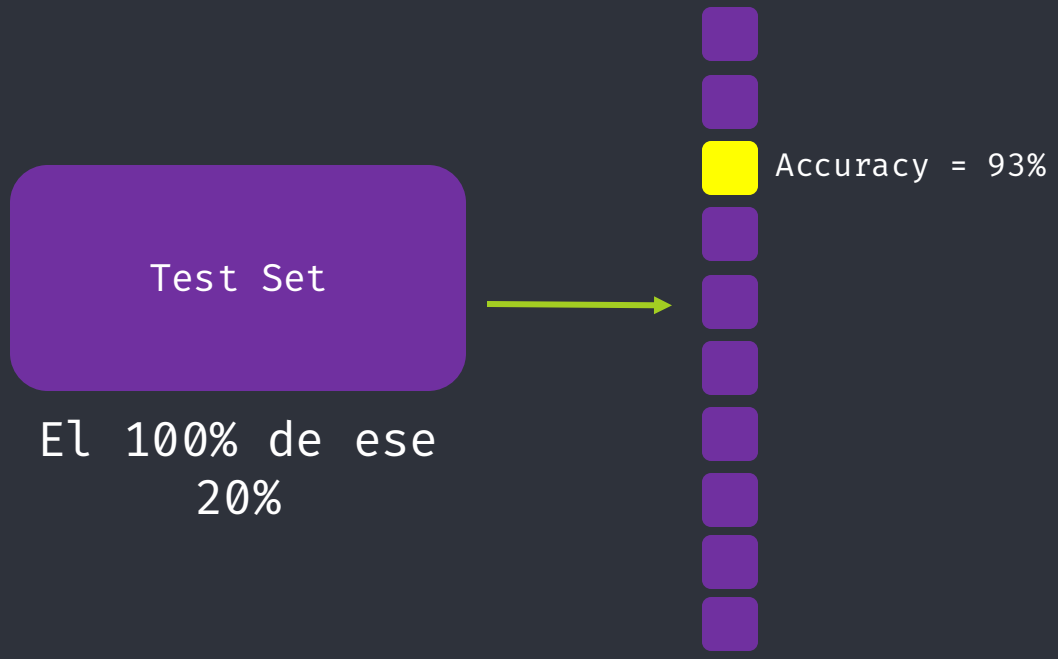
Es dividido en k-folds (un valor tipico es 10)

Y un modelo con una combinacion determinada de hiperparametros es evaluada con un fold a la vez

}

Cross Validation{

<Con cross-validation podemos evaluar desempeño durante entrenamiento>



El 100% de ese 20%

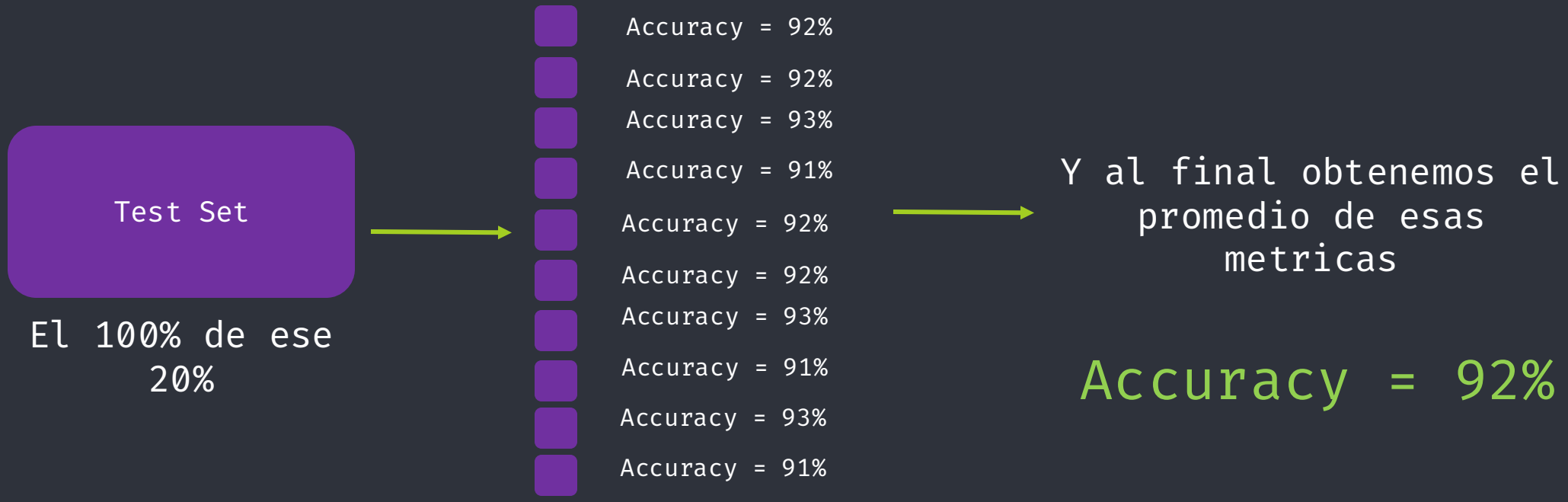
Es dividido en k-folds (un valor tipico es 10)

Y un modelo con una combinacion determinada de hiperparametros es evaluada con un fold a la vez

}

Cross Validation{

<Con cross-validation podemos evaluar desempeño durante entrenamiento>



}

{Implementando Grid Search & Cross Validation}

Ivan Rojas Gonzalez – Universidad Panamericana

Implementando GridSearch & Cross Validation{

<Primero importamos Pipeline y definimos un pipeline. (En NLP, estamos usando el vectorizador tfidf para preprocesar)>

```
from sklearn.pipeline import Pipeline
pipeline = Pipeline([
    ('tfidf', TfidfVectorizer()),
    ('clf', RandomForestClassifier()),
])
```

Importamos Pipeline

Paso 1 - Vectorizar

Paso 2 - Llevar esa informacion al modelo

}

Implementando GridSearch & Cross Validation{

<Ahora importamos GridSearchCV y definimos los parametros para testear>

```
from sklearn.model_selection import GridSearchCV
parameters = {
    'tfidf__ngram_range': [(1, 1), (1, 2)],
    'clf__n_estimators': [10,20],
    'clf__criterion':['gini']
}
```

Importamos GridSearchCV

Distintos ngram ranges para el vectorizador

Distinto parametros para random forest

}

Implementando GridSearch & Cross Validation{

<Instanciamos el grid y lo ajustamos a los datos de entrenamiento>

El numero de K-folds

-1 para paralelizar la creacion de modelos

Para imprimir en consola el progreso

```
grid = GridSearchCV(pipeline, parameters, cv=10, n_jobs=-1, verbose=3, scoring='accuracy')
grid.fit(X_train, y_train)
```

Metrica a monitorear

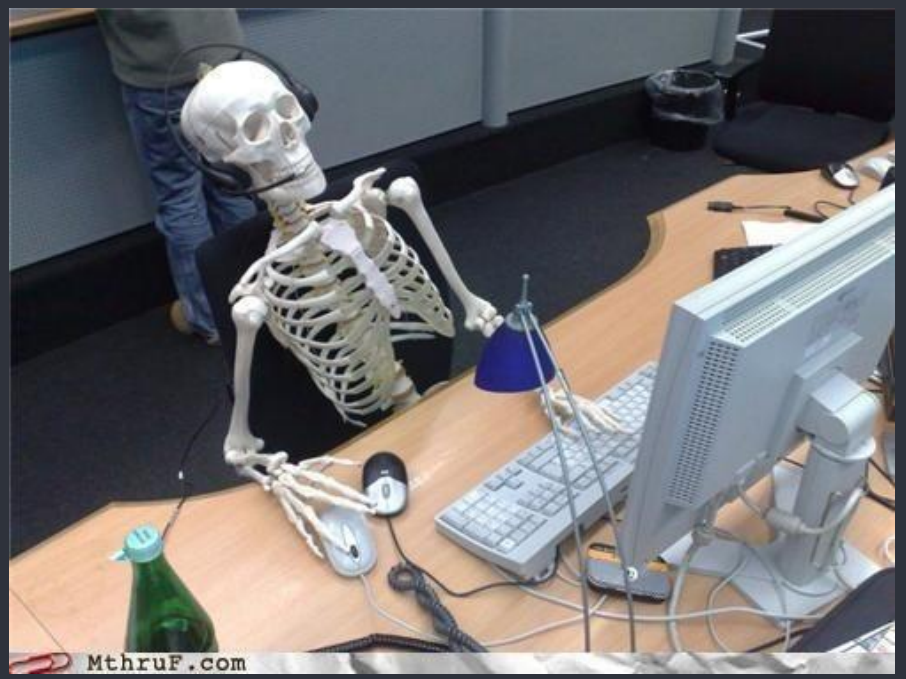
Estimador. En este caso el Pipeline

Los parametros para testear

}

Implementando GridSearch & Cross Validation{

<Y ahora esperamos>



}

Implementando GridSearch & Cross Validation{

<Une vez el proceso termine, podemos obtener el mejor estimador y sus metricas haciendo lo siguiente>

```
grid.best_params_
✓ 0.0s
{'clf__criterion': 'gini',
 'clf__n_estimators': 20,
 'tfidf__ngram_range': (1, 1)}
```

Combinacion de parametros que entregó el accuracy mas alto

```
grid.best_score_
✓ 0.0s
0.9153515624999999
```

Accuracy score del mejor estimador

}

Q&A