

# Predicción de delitos en base a factores climaticos en Colombia

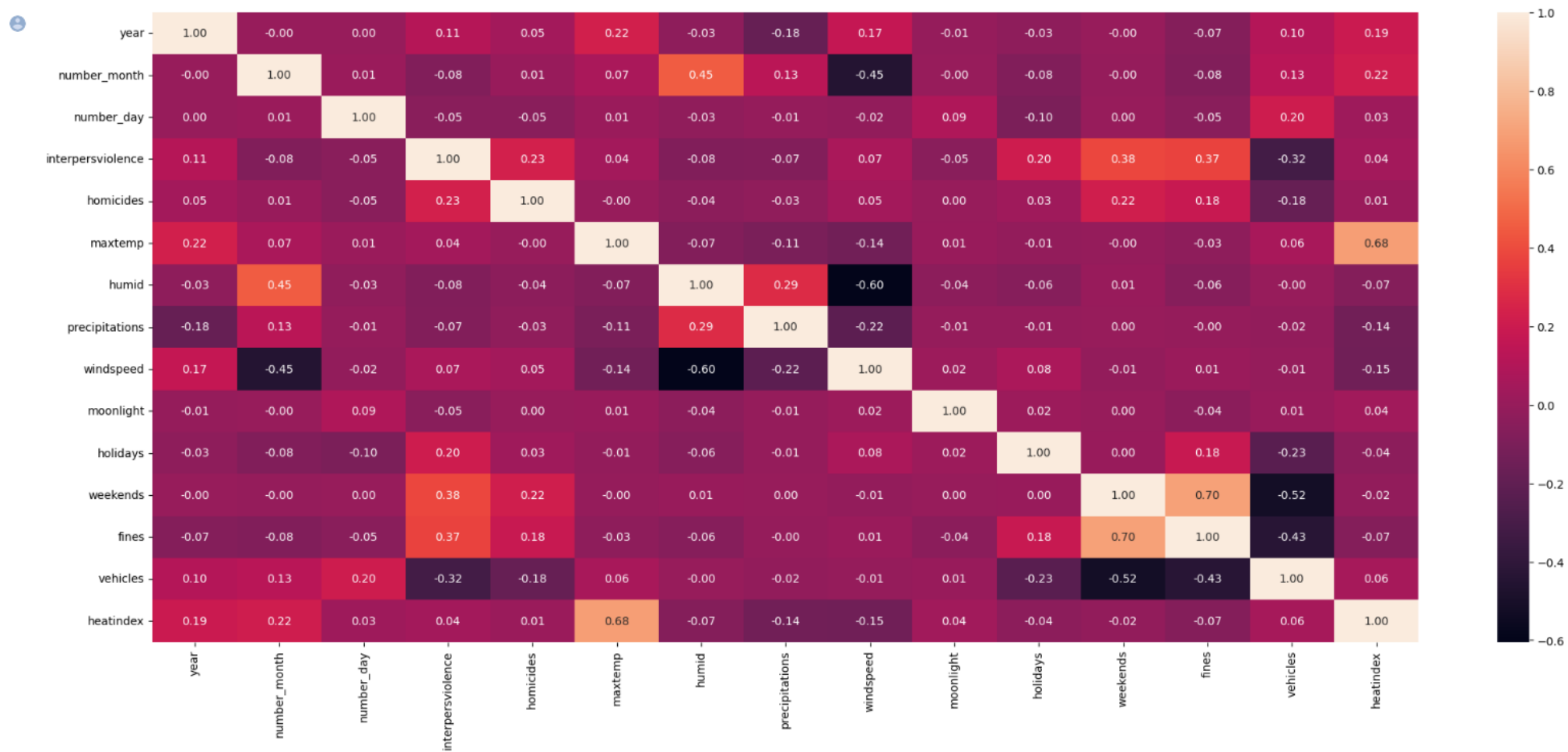
Juan Diego Sepulveda Herrera  
Juan Pablo Ramirez Ortega  
Carlos Arturo Meza Garci



# Dataset

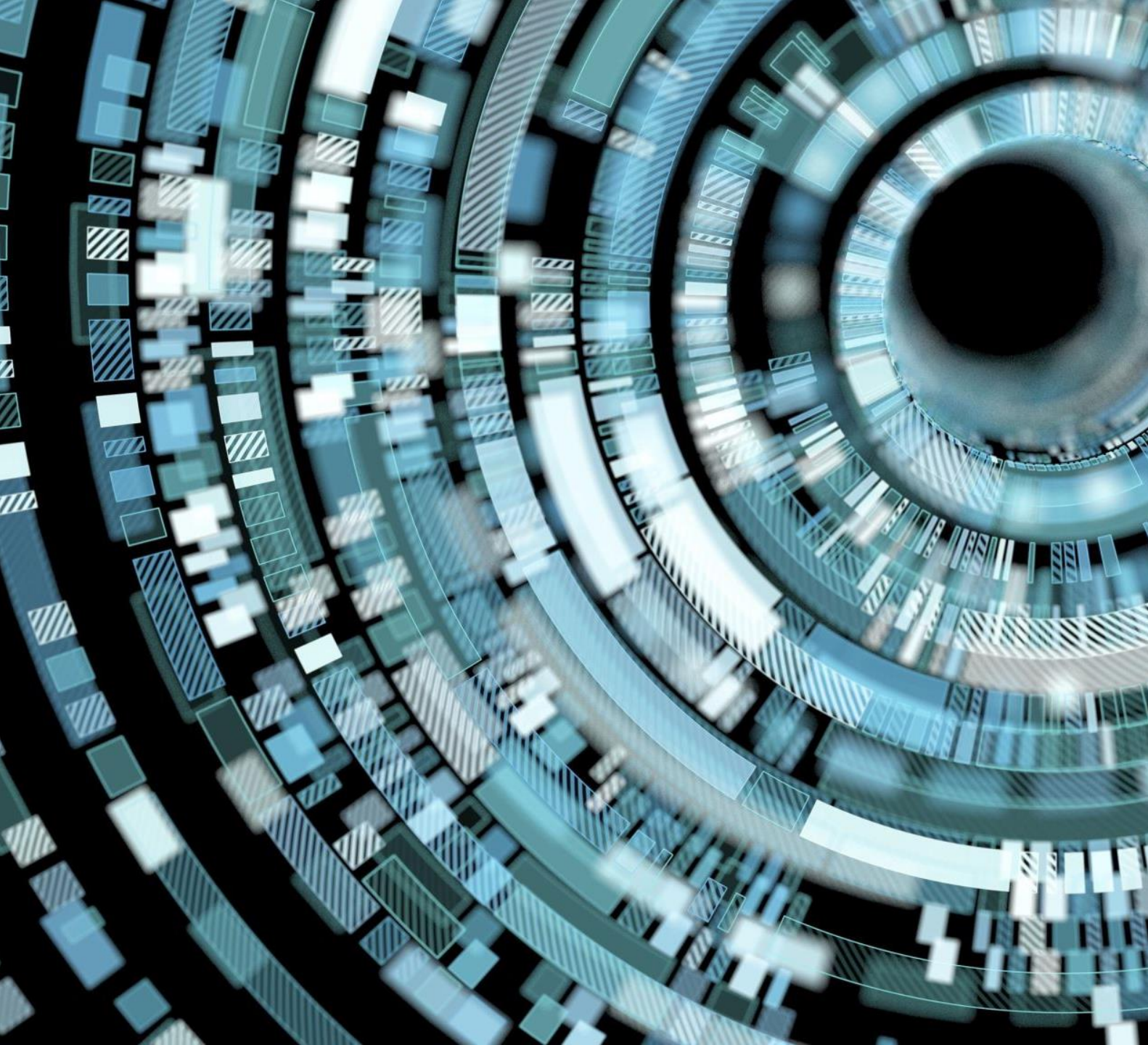


	date	year	month	number_month	day	number_day	interpersviolence	homicides	maxtemp	humid	precipitations	windspeed	moonlight	holidays	weekends	fines	vehicles	heatindex
0	01/01/2010	2010	January	1	Friday	1	31	2	33	78	0.0	15.4	1.000	1	0	2	0	26.2516
1	01/02/2010	2010	January	1	Saturday	2	6	0	32	82	0.0	16.5	0.987	0	1	13	0	30.4034
2	01/03/2010	2010	January	1	Sunday	3	8	1	32	76	0.0	14.8	0.944	0	1	11	0	29.7769
3	01/04/2010	2010	January	1	Monday	4	7	0	34	81	0.0	10.7	0.876	0	0	0	13	30.0638
4	01/05/2010	2010	January	1	Tuesday	5	3	2	35	83	0.0	13.5	0.787	0	0	0	1	30.0271



Matriz de coorelación





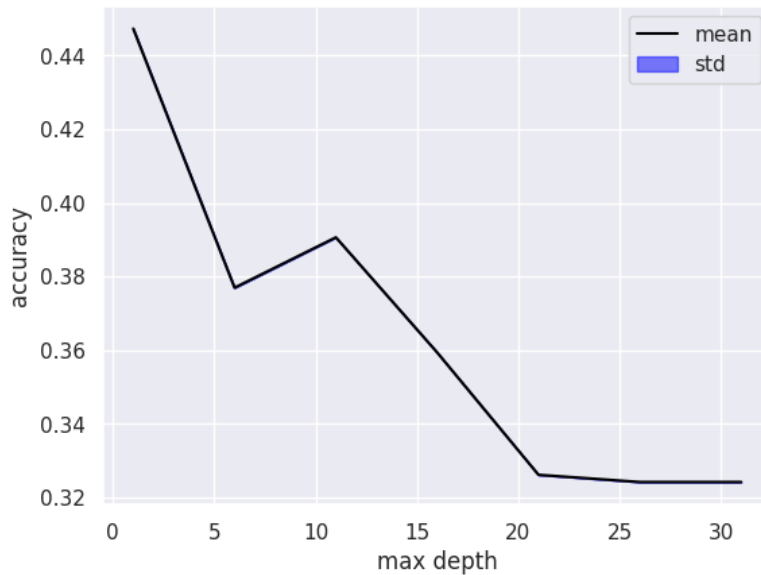
---

## Métodos de clasificación

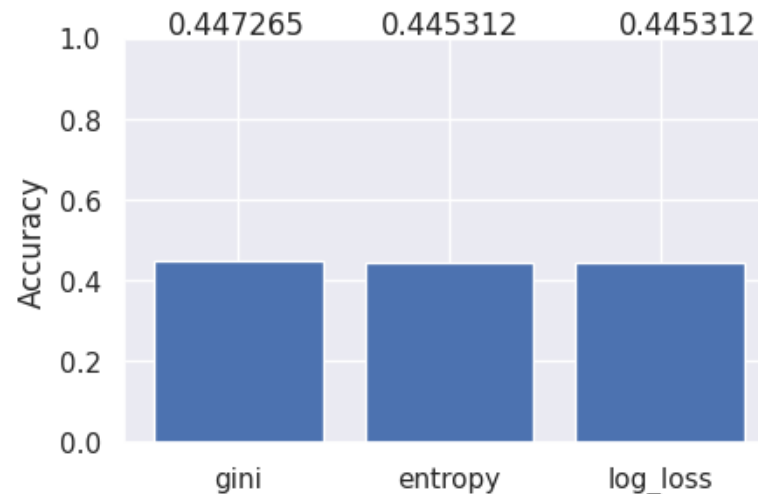
- Decision Tree classifier
- Random Forest classifier
- Support vector machine (SVM)

# DecisionTreeClassifier

Tunnig de parametros  
(max\_depth)



Tunnig de parametros  
(criterion)



Accuracy

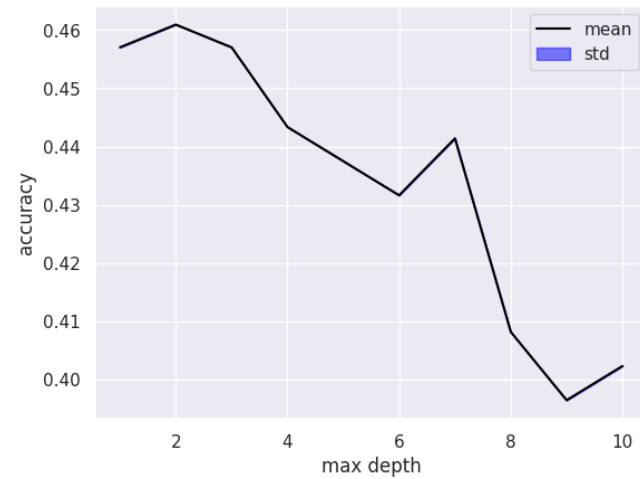
1. Decision Tree classifier (escogida)

```
1 #@title 1. Decision Tree classifier (escogida)
2 from sklearn.tree import DecisionTreeClassifier
3 from sklearn.model_selection import KFold
4 from sklearn.model_selection import cross_val_score
5 # Creamos una instancia del clasificador de árbol de decisión
6 est = DecisionTreeClassifier(criterion='gini', max_depth=1, random_state=21)
7 # Ajustamos el clasificador utilizando los datos de entrenamiento
8 est.fit(X_train,y_train)
9 print(accuracy_score(est.predict(X_test), y_test))
```

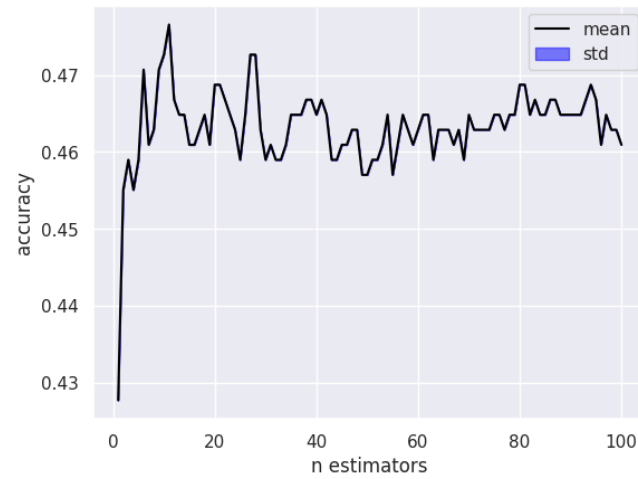
0.447265625

# RandomForestClassifier

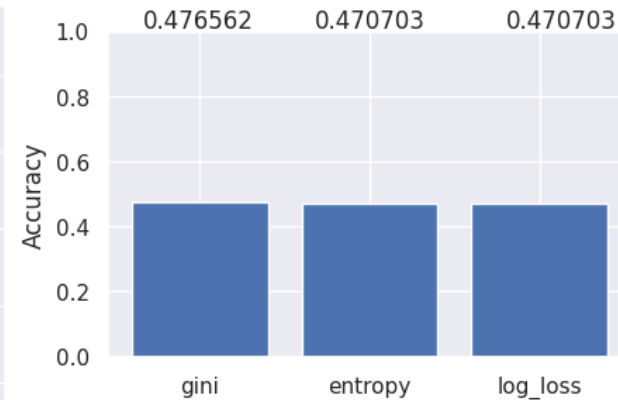
Tunnig de parametros  
(max\_depth)



Tunnig de parametros  
(n\_estimators)



Tunnig de parametros  
(criterion)



Accuracy

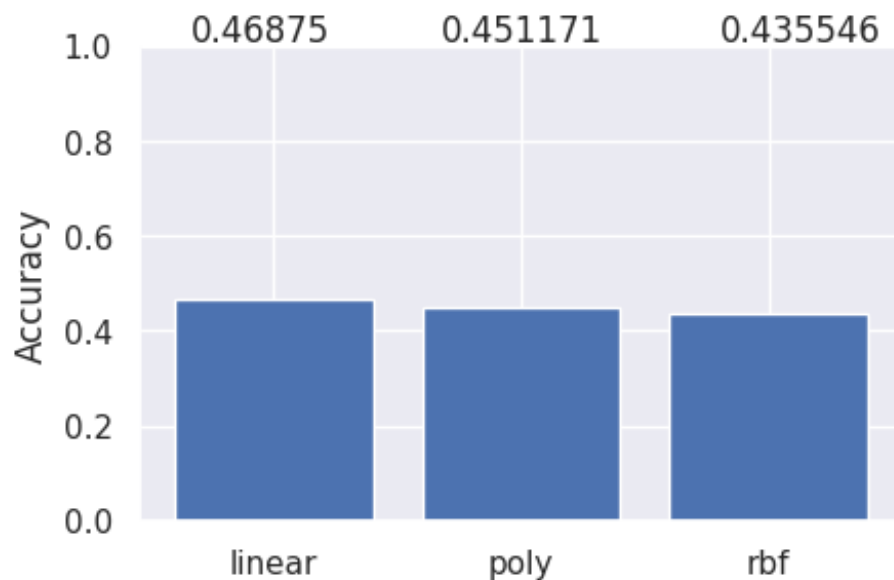
2. Random Forest classifier (escogida)

```
1 #@title 2. Random Forest classifier (escogida)
2 from sklearn.ensemble import RandomForestClassifier
3 est = RandomForestClassifier(criterion="gini",
4                             n_estimators=11,max_depth=2,
5                             random_state=21)
6 est.fit(X_train,y_train)
7 print(accuracy_score(est.predict(X_test), y_test))
```

0.4765625

# SupportVectorMachine

Tunnig de parametros (kernel)



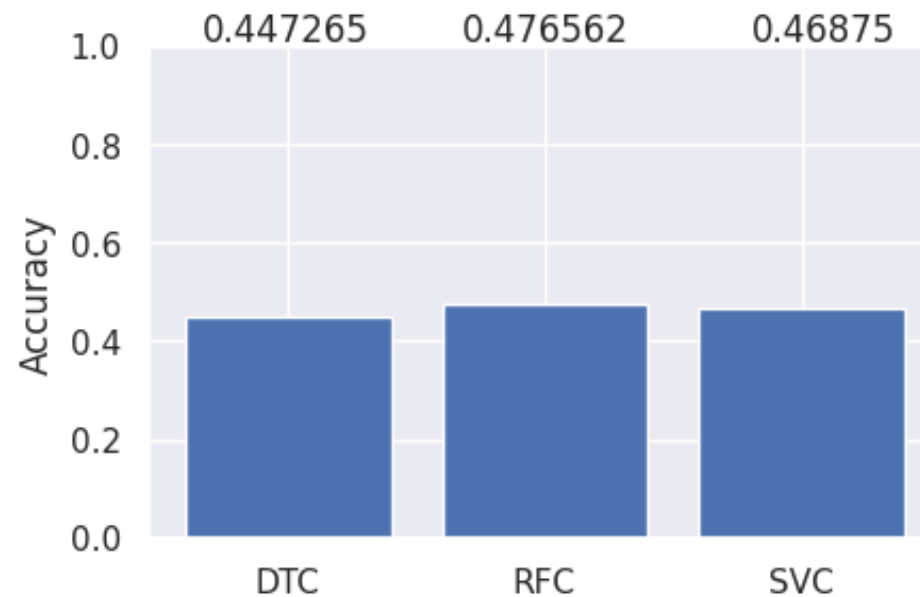
Accuracy

Support vector machine (SVM) (escogida)

```
[94] 1 #@title Support vector machine (SVM) (escogida)
      2 from sklearn.svm import SVC #c
      3 est = SVC(kernel="linear",C=51,random_state=21)
      4 est.fit(X_train,y_train)
      5 print(accuracy_score(est.predict(X_test), y_test))
```

0.46875

# Comparación de métodos para escoger el mejor



El mejor método de clasificación es RandomForestClassifier.



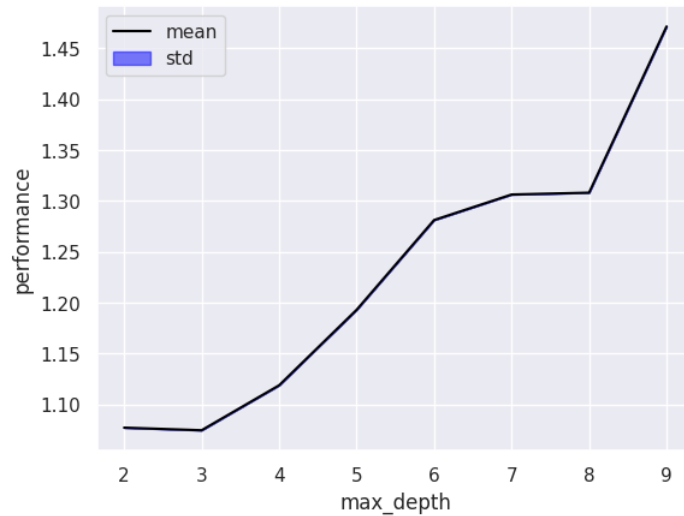
# Regresión



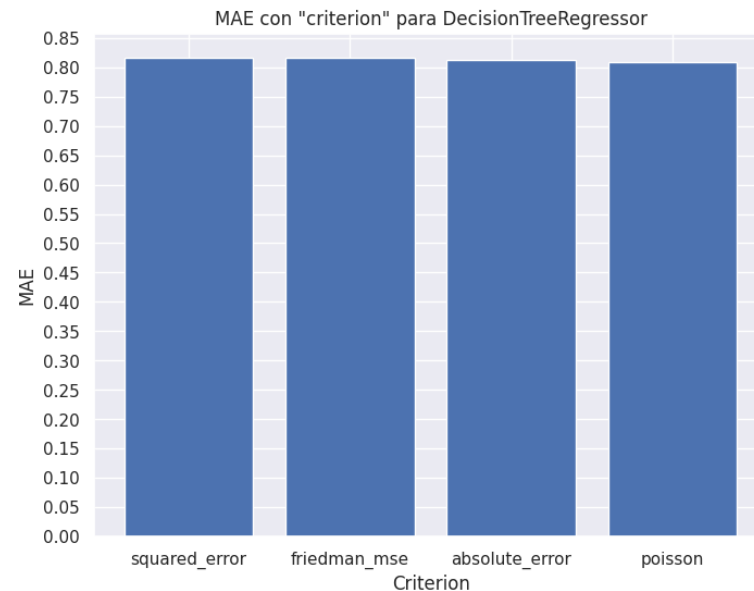
- Decision Tree Regressor
- Random Forest Regressor
- Support vector regressor (SVR)

# DecisionTreeRegressor

Tunnig de parametros  
(max\_depth)



Tunnig de parametros  
(criterion)



% de error

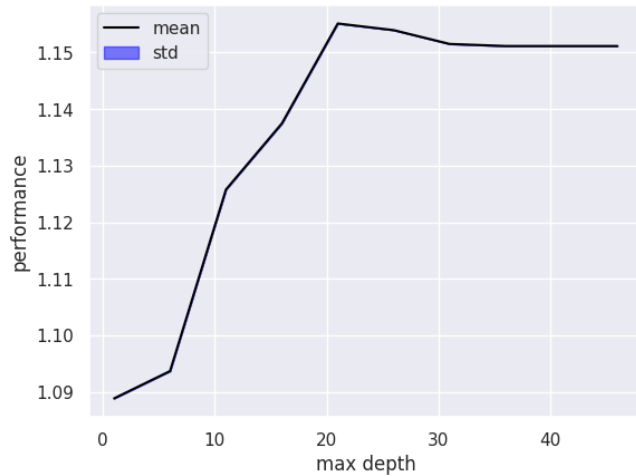
► Porcentajes de error

✓ 0 s [▶] [Mostrar código](#)

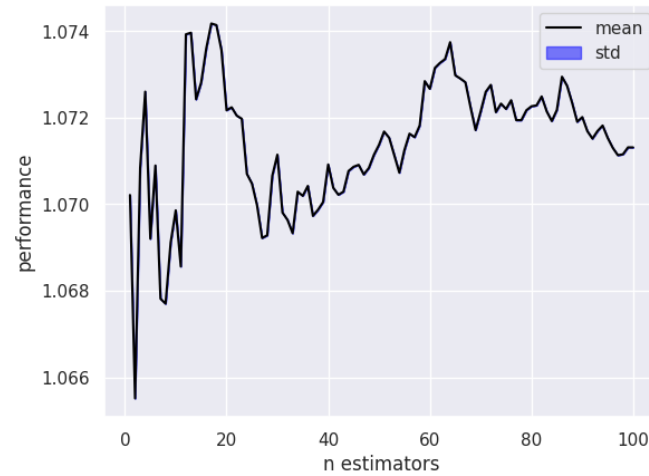
❏ MSE depth split data: 1.062  
RMSE depth split data: 1.031  
MAE depth split data (escogido): 0.809

# RandomForestRegressor

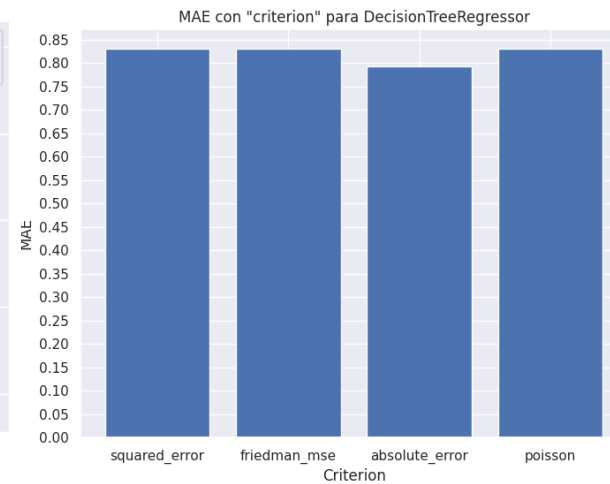
Tunnig de parametros  
(max\_depth)



Tunnig de parametros  
(n\_estimator)



Tunnig de parametros  
(criterion)



% de error

► Porcentajes de error

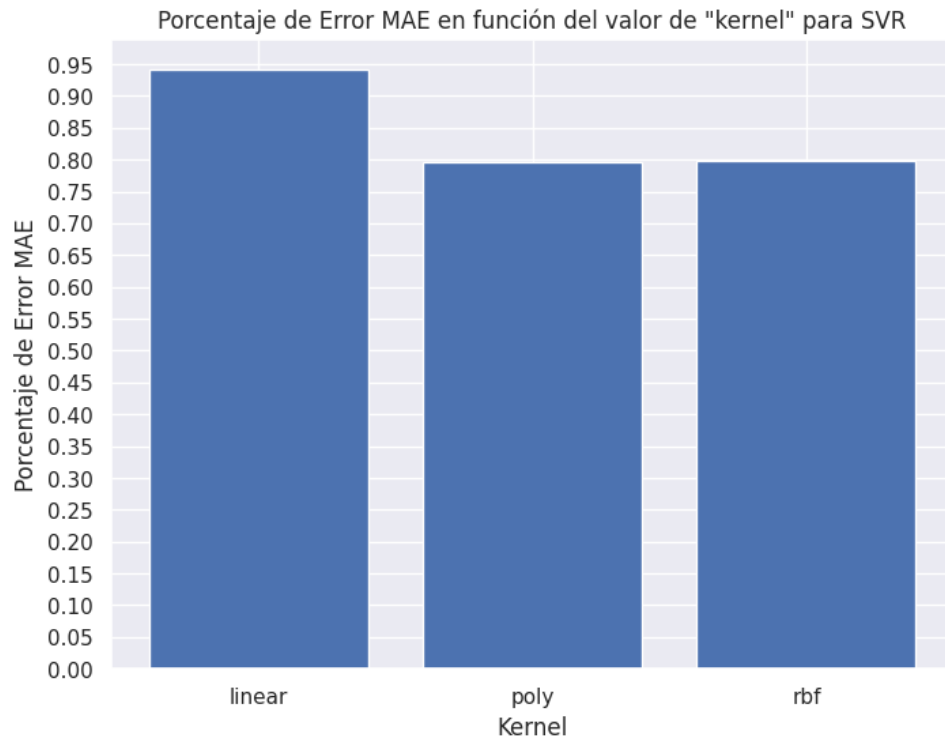
✓ [177]  
0s

[Mostrar código](#)

MSE depth split data: 1.145  
RMSE depth split data: 1.070  
MAE depth split data (escogida): 0.793

# SupportVectorRegression

Tunnig de parametros (kernel)



% de error

Porcentajes de error

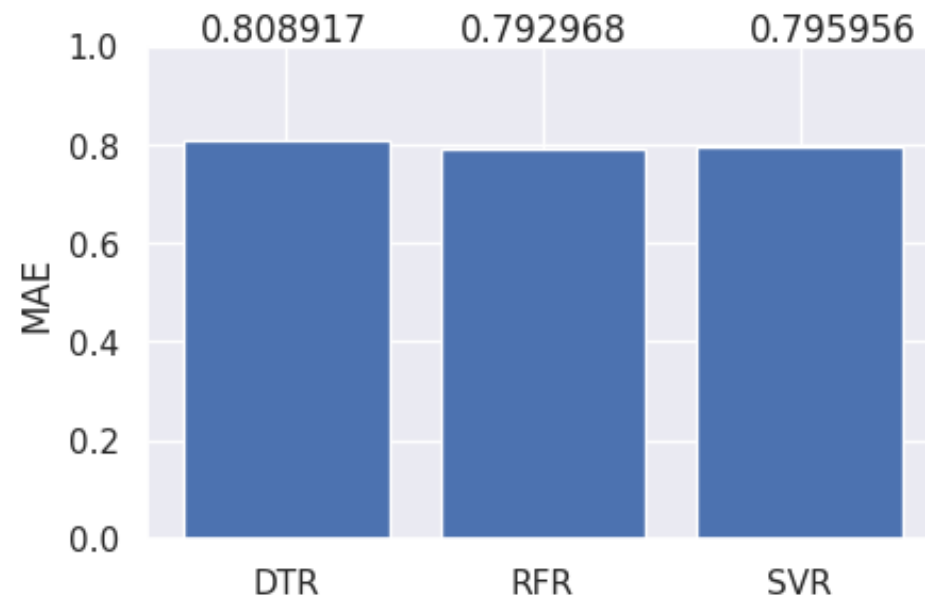
[178]

[Mostrar código](#)

```
➤ MSE depth split data: 1.078  
  RMSE depth split data: 1.038  
  MAE depth split data (escogida): 0.796
```



# Comparación de métodos para escoger el mejor



El mejor método de clasificación es RandomForestRegressor.



# Conclusiones

---