

Paterno	Materno	Nombres	Carnet
Apaza	Colque	Miguel Fernando	5477212

- Del anterior problema en PYTHON y WEKA realice un árbol de decisión basado en entropía, compare los resultados, esto debe repetirse al menos 10 veces y comparar las matrices de confusión y obtener la confiabilidad. Se recomienda comparar media y mediana de los resultados.

WEKA

Classifier output

Time taken to test model on test split: 0.02 seconds

=== Summary ===

Correctly Classified Instances	182	91	%
Incorrectly Classified Instances	18	9	%
Kappa statistic	0.8752		
Mean absolute error	0.1352		
Root mean squared error	0.1865		
Relative absolute error	16.1723	%	
Root relative squared error	43.6412	%	
Total Number of Instances	200		

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1.000	0.033	0.986	1.000	0.990	0.936	0.990	0.941	E
	0.639	0.000	1.000	0.639	0.780	0.769	0.970	0.879	ET
	0.930	0.013	0.952	0.930	0.941	0.925	0.994	0.970	T
	0.973	0.087	0.866	0.973	0.916	0.867	0.987	0.967	T
Weighted Avg.	0.910	0.042	0.918	0.910	0.905	0.879	0.986	0.946	

=== Confusion Matrix ===

	a	b	c	d	<-- classified as
48	0	0	0	1	a = E
3	23	2	8	1	b = ET
0	0	40	3	1	c = T
2	0	0	71	1	d = T

PYTHON

```

from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_recall_fscore_support
from sklearn.metrics import roc_auc_score
from sklearn.metrics import prc_auc_score
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from sklearn.metrics import relative_absolute_error
from sklearn.metrics import root_mean_squared_error
from sklearn.metrics import root_relative_squared_error
from sklearn.metrics import total_number_of_instances

# Load data
X, y = datasets.load_heart().data, datasets.load_heart().target

# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train model
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier()
model.fit(X_train, y_train)

# Predict
y_pred = model.predict(X_test)

# Evaluate
confusion_matrix(y_test, y_pred)
accuracy_score(y_test, y_pred)
precision_recall_fscore_support(y_test, y_pred, average='weighted')
roc_auc_score(y_test, y_pred)
prc_auc_score(y_test, y_pred)
mean_absolute_error(y_test, y_pred)
mean_squared_error(y_test, y_pred)
relative_absolute_error(y_test, y_pred)
root_mean_squared_error(y_test, y_pred)
root_relative_squared_error(y_test, y_pred)
total_number_of_instances(y_test, y_pred)

```