7506R_TP2_GRUPO16_ENTREGA_ENSAMBLES

December 8, 2022

[1]:
```
!pip install numpy==1.21
!pip install matplotlib==3.1.3
!pip install dtreeviz
!pip install kneed
!pip install pyclustertend
!pip install geopandas
!pip install pyreadstat
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting numpy==1.21
  Downloading
numpy-1.21.0-cp38-cp38-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (15.7 MB)
     |                  | 15.7 MB 389 kB/s
Installing collected packages: numpy
  Attempting uninstall: numpy
    Found existing installation: numpy 1.21.6
    Uninstalling numpy-1.21.6:
      Successfully uninstalled numpy-1.21.6
Successfully installed numpy-1.21.0
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting matplotlib==3.1.3
  Downloading matplotlib-3.1.3-cp38-cp38-manylinux1_x86_64.whl (13.1 MB)
     |                  | 13.1 MB 5.7 MB/s
Requirement already satisfied: python-dateutil>=2.1 in
/usr/local/lib/python3.8/dist-packages (from matplotlib==3.1.3) (2.8.2)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.8/dist-packages (from matplotlib==3.1.3) (1.4.4)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in
/usr/local/lib/python3.8/dist-packages (from matplotlib==3.1.3) (3.0.9)
Requirement already satisfied: numpy>=1.11 in /usr/local/lib/python3.8/dist-
packages (from matplotlib==3.1.3) (1.21.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.8/dist-
packages (from matplotlib==3.1.3) (0.11.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.8/dist-
packages (from python-dateutil>=2.1->matplotlib==3.1.3) (1.15.0)
```

```
Installing collected packages: matplotlib
  Attempting uninstall: matplotlib
    Found existing installation: matplotlib 3.2.2
    Uninstalling matplotlib-3.2.2:
      Successfully uninstalled matplotlib-3.2.2
Successfully installed matplotlib-3.1.3
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting dtreeviz
  Downloading dtreeviz-1.4.1-py3-none-any.whl (72 kB)
     |                 | 72 kB 854 kB/s
Requirement already satisfied: matplotlib in
/usr/local/lib/python3.8/dist-packages (from dtreeviz) (3.1.3)
Requirement already satisfied: pytest in /usr/local/lib/python3.8/dist-packages
(from dtreeviz) (3.6.4)
Collecting colour
  Downloading colour-0.1.5-py2.py3-none-any.whl (23 kB)
Requirement already satisfied: pandas in /usr/local/lib/python3.8/dist-packages
(from dtreeviz) (1.3.5)
Requirement already satisfied: numpy in /usr/local/lib/python3.8/dist-packages
(from dtreeviz) (1.21.0)
Requirement already satisfied: graphviz>=0.9 in /usr/local/lib/python3.8/dist-
packages (from dtreeviz) (0.10.1)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.8/dist-
packages (from dtreeviz) (1.0.2)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.8/dist-packages (from matplotlib->dtreeviz) (1.4.4)
Requirement already satisfied: python-dateutil>=2.1 in
/usr/local/lib/python3.8/dist-packages (from matplotlib->dtreeviz) (2.8.2)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in
/usr/local/lib/python3.8/dist-packages (from matplotlib->dtreeviz) (3.0.9)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.8/dist-
packages (from matplotlib->dtreeviz) (0.11.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.8/dist-
packages (from python-dateutil>=2.1->matplotlib->dtreeviz) (1.15.0)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.8/dist-
packages (from pandas->dtreeviz) (2022.6)
Requirement already satisfied: setuptools in /usr/local/lib/python3.8/dist-
packages (from pytest->dtreeviz) (57.4.0)
Requirement already satisfied: pluggy<0.8,>=0.5 in
/usr/local/lib/python3.8/dist-packages (from pytest->dtreeviz) (0.7.1)
Requirement already satisfied: py>=1.5.0 in /usr/local/lib/python3.8/dist-
packages (from pytest->dtreeviz) (1.11.0)
Requirement already satisfied: attrs>=17.4.0 in /usr/local/lib/python3.8/dist-
packages (from pytest->dtreeviz) (22.1.0)
Requirement already satisfied: atomicwrites>=1.0 in
/usr/local/lib/python3.8/dist-packages (from pytest->dtreeviz) (1.4.1)
```

```
Requirement already satisfied: more-itertools>=4.0.0 in
/usr/local/lib/python3.8/dist-packages (from pytest->dtreeviz) (9.0.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/usr/local/lib/python3.8/dist-packages (from scikit-learn->dtreeviz) (3.1.0)
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.8/dist-
packages (from scikit-learn->dtreeviz) (1.2.0)
Requirement already satisfied: scipy>=1.1.0 in /usr/local/lib/python3.8/dist-
packages (from scikit-learn->dtreeviz) (1.7.3)
Installing collected packages: colour, dtreeviz
Successfully installed colour-0.1.5 dtreeviz-1.4.1
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting kneed
  Downloading kneed-0.8.1-py2.py3-none-any.whl (10 kB)
Requirement already satisfied: numpy>=1.14.2 in /usr/local/lib/python3.8/dist-
packages (from kneed) (1.21.0)
Requirement already satisfied: scipy>=1.0.0 in /usr/local/lib/python3.8/dist-
packages (from kneed) (1.7.3)
Installing collected packages: kneed
Successfully installed kneed-0.8.1
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting pyclustertend
  Downloading pyclustertend-1.8.2-py3-none-any.whl (7.2 kB)
Collecting scikit-learn<2.0.0,>=1.1.2
  Downloading
scikit_learn-1.2.0-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (9.7
MB)
     |                  | 9.7 MB 4.4 MB/s
Collecting numba<0.55.0,>=0.54.1
  Downloading
numba-0.54.1-cp38-cp38-manylinux2014_x86_64.manylinux_2_17_x86_64.whl (3.3 MB)
     |                  | 3.3 MB 42.0 MB/s
Collecting numpy==1.20.3
  Downloading
numpy-1.20.3-cp38-cp38-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (15.4 MB)
     |                  | 15.4 MB 51.4 MB/s
Requirement already satisfied: pandas<2.0.0,>=1.2.0 in
/usr/local/lib/python3.8/dist-packages (from pyclustertend) (1.3.5)
Collecting matplotlib<4.0.0,>=3.3.3
  Downloading
matplotlib-3.6.2-cp38-cp38-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (9.4
MB)
     |                  | 9.4 MB 40.1 MB/s
Requirement already satisfied: pillow>=6.2.0 in
/usr/local/lib/python3.8/dist-packages (from
matplotlib<4.0.0,>=3.3.3->pyclustertend) (7.1.2)
Requirement already satisfied: kiwisolver>=1.0.1 in
```

```
/usr/local/lib/python3.8/dist-packages (from
matplotlib<4.0.0,>=3.3.3->pyclustertend) (1.4.4)
Requirement already satisfied: pyparsing>=2.2.1 in
/usr/local/lib/python3.8/dist-packages (from
matplotlib<4.0.0,>=3.3.3->pyclustertend) (3.0.9)
Collecting fonttools>=4.22.0
  Downloading fonttools-4.38.0-py3-none-any.whl (965 kB)
     |                  | 965 kB 33.4 MB/s
Requirement already satisfied: cycler>=0.10 in
/usr/local/lib/python3.8/dist-packages (from
matplotlib<4.0.0,>=3.3.3->pyclustertend) (0.11.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.8/dist-
packages (from matplotlib<4.0.0,>=3.3.3->pyclustertend) (21.3)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.8/dist-packages (from
matplotlib<4.0.0,>=3.3.3->pyclustertend) (2.8.2)
Collecting contourpy>=1.0.1
  Downloading
contourpy-1.0.6-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (295
kB)
     |                  | 295 kB 85.7 MB/s
Collecting llvmlite<0.38,>=0.37.0rc1
  Downloading llvmlite-0.37.0-cp38-cp38-manylinux2014_x86_64.whl (26.3 MB)
     |                  | 26.3 MB 1.4 MB/s
Requirement already satisfied: setuptools in
/usr/local/lib/python3.8/dist-packages (from
numba<0.55.0,>=0.54.1->pyclustertend) (57.4.0)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.8/dist-
packages (from pandas<2.0.0,>=1.2.0->pyclustertend) (2022.6)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.8/dist-
packages (from python-dateutil>=2.7->matplotlib<4.0.0,>=3.3.3->pyclustertend)
(1.15.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/usr/local/lib/python3.8/dist-packages (from scikit-
learn<2.0.0,>=1.1.2->pyclustertend) (3.1.0)
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.8/dist-
packages (from scikit-learn<2.0.0,>=1.1.2->pyclustertend) (1.7.3)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.8/dist-
packages (from scikit-learn<2.0.0,>=1.1.2->pyclustertend) (1.2.0)
Installing collected packages: numpy, llvmlite, fonttools, contourpy, scikit-
learn, numba, matplotlib, pyclustertend
  Attempting uninstall: numpy
    Found existing installation: numpy 1.21.0
    Uninstalling numpy-1.21.0:
      Successfully uninstalled numpy-1.21.0
  Attempting uninstall: llvmlite
    Found existing installation: llvmlite 0.39.1
    Uninstalling llvmlite-0.39.1:
```

```
      Successfully uninstalled llvmlite-0.39.1
  Attempting uninstall: scikit-learn
    Found existing installation: scikit-learn 1.0.2
    Uninstalling scikit-learn-1.0.2:
      Successfully uninstalled scikit-learn-1.0.2
  Attempting uninstall: numba
    Found existing installation: numba 0.56.4
    Uninstalling numba-0.56.4:
      Successfully uninstalled numba-0.56.4
  Attempting uninstall: matplotlib
    Found existing installation: matplotlib 3.1.3
    Uninstalling matplotlib-3.1.3:
      Successfully uninstalled matplotlib-3.1.3
ERROR: pip's dependency resolver does not currently take into account all
the packages that are installed. This behaviour is the source of the following
dependency conflicts.
cmdstanpy 1.0.8 requires numpy>=1.21, but you have numpy 1.20.3 which is
incompatible.
Successfully installed contourpy-1.0.6 fonttools-4.38.0 llvmlite-0.37.0
matplotlib-3.6.2 numba-0.54.1 numpy-1.20.3 pyclustertend-1.8.2 scikit-
learn-1.2.0

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting geopandas
  Downloading geopandas-0.12.1-py3-none-any.whl (1.1 MB)
     |                  | 1.1 MB 4.9 MB/s
Collecting fiona>=1.8
  Downloading
Fiona-1.8.22-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (16.6 MB)
     |                  | 16.6 MB 51.0 MB/s
Requirement already satisfied: packaging in /usr/local/lib/python3.8/dist-
packages (from geopandas) (21.3)
Collecting pyproj>=2.6.1.post1
  Downloading
pyproj-3.4.0-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (7.8 MB)
     |                  | 7.8 MB 51.6 MB/s
Requirement already satisfied: shapely>=1.7 in
/usr/local/lib/python3.8/dist-packages (from geopandas) (1.8.5.post1)
Requirement already satisfied: pandas>=1.0.0 in /usr/local/lib/python3.8/dist-
packages (from geopandas) (1.3.5)
Requirement already satisfied: click>=4.0 in /usr/local/lib/python3.8/dist-
packages (from fiona>=1.8->geopandas) (7.1.2)
Collecting cligj>=0.5
  Downloading cligj-0.7.2-py3-none-any.whl (7.1 kB)
Collecting munch
```

```
  Downloading munch-2.5.0-py2.py3-none-any.whl (10 kB)
Requirement already satisfied: attrs>=17 in /usr/local/lib/python3.8/dist-
packages (from fiona>=1.8->geopandas) (22.1.0)
Collecting click-plugins>=1.0
  Downloading click_plugins-1.1.1-py2.py3-none-any.whl (7.5 kB)
Requirement already satisfied: six>=1.7 in /usr/local/lib/python3.8/dist-
packages (from fiona>=1.8->geopandas) (1.15.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.8/dist-
packages (from fiona>=1.8->geopandas) (57.4.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.8/dist-packages
(from fiona>=1.8->geopandas) (2022.9.24)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.8/dist-
packages (from pandas>=1.0.0->geopandas) (2022.6)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.8/dist-
packages (from pandas>=1.0.0->geopandas) (1.20.3)
Requirement already satisfied: python-dateutil>=2.7.3 in
/usr/local/lib/python3.8/dist-packages (from pandas>=1.0.0->geopandas) (2.8.2)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/usr/local/lib/python3.8/dist-packages (from packaging->geopandas) (3.0.9)
Installing collected packages: munch, cligj, click-plugins, pyproj, fiona,
geopandas
Successfully installed click-plugins-1.1.1 cligj-0.7.2 fiona-1.8.22
geopandas-0.12.1 munch-2.5.0 pyproj-3.4.0
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting pyreadstat
  Downloading
pyreadstat-1.2.0-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.7
MB)
     |                  | 2.7 MB 5.3 MB/s
Requirement already satisfied: pandas>=1.2.0 in
/usr/local/lib/python3.8/dist-packages (from pyreadstat) (1.3.5)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.8/dist-
packages (from pandas>=1.2.0->pyreadstat) (2022.6)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.8/dist-
packages (from pandas>=1.2.0->pyreadstat) (1.20.3)
Requirement already satisfied: python-dateutil>=2.7.3 in
/usr/local/lib/python3.8/dist-packages (from pandas>=1.2.0->pyreadstat) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.8/dist-
packages (from python-dateutil>=2.7.3->pandas>=1.2.0->pyreadstat) (1.15.0)
Installing collected packages: pyreadstat
Successfully installed pyreadstat-1.2.0
```

[2]:
```
#Data sets
import pandas as pd
import numpy as np
import csv
```

```
import geopandas as gp
from collections import defaultdict

#Visualización
import matplotlib.pyplot as plt
import seaborn as sns
import dtreeviz.trees as dtreeviz
import plotly.graph_objects as go
%matplotlib inline

#modelos y métricas
import statistics
from sklearn.neighbors import KNeighborsRegressor
from sklearn.linear_model import LogisticRegressionCV, Lasso
from sklearn import tree, metrics
from sklearn.ensemble import RandomForestRegressor, RandomForestClassifier,␣
 ↪ExtraTreesClassifier, RandomForestClassifier, AdaBoostClassifier,␣
 ↪VotingClassifier, StackingRegressor, GradientBoostingRegressor
from sklearn.model_selection import RandomizedSearchCV, GridSearchCV,␣
 ↪cross_val_score, RepeatedStratifiedKFold
from sklearn.metrics import confusion_matrix, precision_recall_curve,␣
 ↪roc_curve, recall_score, accuracy_score,f1_score, precision_score, auc,␣
 ↪roc_auc_score, mean_squared_error, silhouette_score,␣
 ↪classification_report,mean_absolute_error, max_error, median_absolute_error,␣
 ↪r2_score, explained_variance_score
import xgboost as xgb
from sklearn.svm import SVR
#preprocesamiento
from sklearn.preprocessing import MinMaxScaler, StandardScaler

#configuración warnings
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
warnings.simplefilter(action='ignore', category=UserWarning)
```

0.0.1 Data sets

[3]:
```
url = 'https://drive.google.com/file/d/1ziV8Kmuw_Vtv7aX1zGnfMdoLMdOnfoOr/view?
 ↪usp=share_link'
path = 'https://drive.google.com/uc?export=download&id='+url.split('/')[-2]

ds_train = pd.read_csv(path)

url = 'https://drive.google.com/file/d/1j34_hlvFOJV_8TN2jdnV2GXSArfWlUUK/view?
 ↪usp=share_link'
path = 'https://drive.google.com/uc?export=download&id='+url.split('/')[-2]
```

```
ds_test = pd.read_csv(path)
```

Ponemos valores numéricos para trabajar con los tipos de propiedades

[4]:
```
de_temp_train = ds_train.copy()

de_temp_train.loc[ds_train["tipo_precio"]=="bajo", "target"]= 0
de_temp_train.loc[ds_train["tipo_precio"]=="medio", "target"]= 1
de_temp_train.loc[ds_train["tipo_precio"]=="alto", "target"]= 2
```

[5]:
```
de_temp_test = ds_test.copy()

de_temp_test.loc[ds_test["tipo_precio"]=="bajo", "target"]= 0
de_temp_test.loc[ds_test["tipo_precio"]=="medio", "target"]= 1
de_temp_test.loc[ds_test["tipo_precio"]=="alto", "target"]= 2
```

[6]:
```
ds_train_x = de_temp_train.drop(['id', 'tipo_precio', 'property_price',␣
 ↪'target'], axis='columns', inplace=False)
ds_test_x = de_temp_test.drop(['id', 'tipo_precio', 'property_price',␣
 ↪'target'], axis='columns', inplace=False)
```

[7]:
```
ds_train_x = pd.get_dummies(ds_train_x, columns=["barrio","property_type"],␣
 ↪drop_first=True)
ds_test_x = pd.get_dummies(ds_test_x, columns=["barrio","property_type"],␣
 ↪drop_first=True)
```

0.0.2 Funciones generales

[8]:
```
#Funcion para plotear la performance
def plot_results(model_scores, name):

    model_names = list(model_scores.keys())
    results = [model_scores[model] for model in model_names]
    fig = go.Figure()
    for model, result in zip(model_names, results):
        fig.add_trace(go.Box(
            y=result,
            name=model,
            boxpoints='all',
            jitter=0.5,
            whiskerwidth=0.2,
            marker_size=2,
            line_width=1)
        )

    fig.update_layout(
```

```
    title='Performance of Different Models Using 5-Fold Cross-Validation',
    paper_bgcolor='rgb(243, 243, 243)',
    plot_bgcolor='rgb(243, 243, 243)',
    xaxis_title='Model',
    yaxis_title='Accuracy',
    showlegend=False)
    fig.show()
```

[9]:
```
#Funcion para evaluar los modelos 5fold -CV
def evaluate_model(model, x, y):
    cv = RepeatedStratifiedKFold(n_splits=5, n_repeats=2, random_state=1)
    scores = cross_val_score(model, x, y, cv=cv, verbose=1, n_jobs=3,␣
 ↪error_score='raise')
    return scores
```

0.0.3 Voting

[10]:
```
ds_train_y = de_temp_train['target'].copy()
ds_test_y = de_temp_test['target'].copy()
```

[11]:
```
sscaler=StandardScaler()
sscaler.fit(pd.DataFrame(ds_train_x))
```

[11]:
```
StandardScaler()
```

[12]:
```
x_train_transform=sscaler.transform(pd.DataFrame(ds_train_x))
x_test_transform=sscaler.transform(pd.DataFrame(ds_test_x))
```

Usamos DecisionTree, RandomForest y AdaBoost

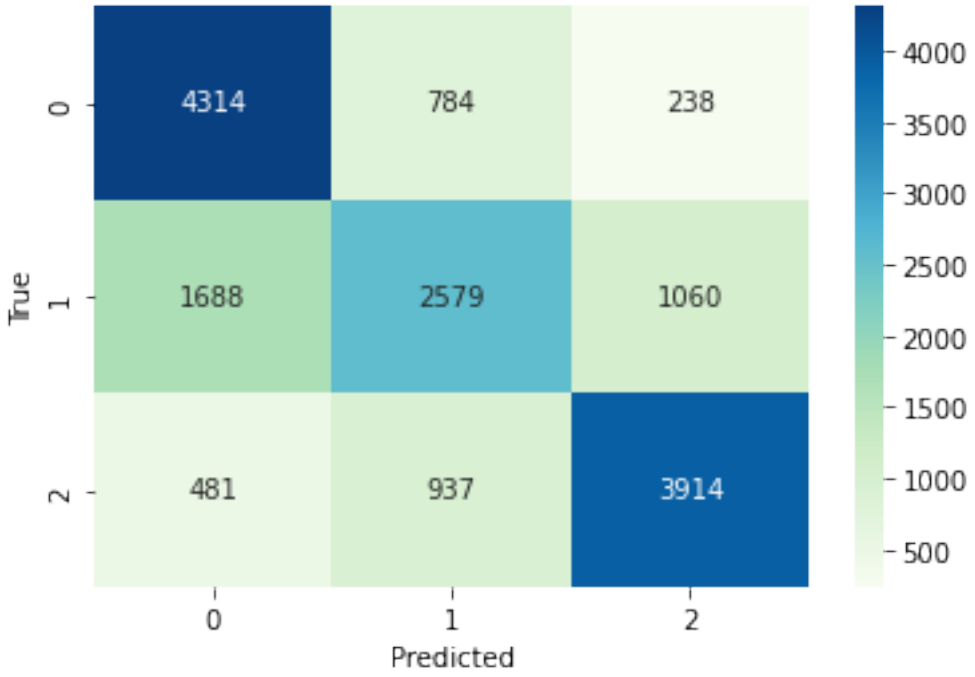
[13]:
```
#Creamos los clasificadores
tree_classifier = tree.DecisionTreeClassifier()
rf = RandomForestClassifier()
adaboost = AdaBoostClassifier()

#Entreno clasificadores
voting = VotingClassifier(estimators=[('tree', tree_classifier), ('rf', rf),␣
 ↪('adaboost', adaboost)], voting='hard')

#Entreno el ensemble
voting.fit(x_train_transform, ds_train_y)
```

[13]:
```
VotingClassifier(estimators=[('tree', DecisionTreeClassifier()),
                             ('rf', RandomForestClassifier()),
                             ('adaboost', AdaBoostClassifier())])
```

[14]:
```
y_pred = voting.predict(x_test_transform)
```

[15]:
```
#Calculo las métricas en el conjunto de evaluación
accuracy=accuracy_score(ds_test_y,y_pred)

print(classification_report(ds_test_y,y_pred))
print("Accuracy: "+str(accuracy))
print(" ")

#Creo la matriz de confusión
tabla=confusion_matrix(ds_test_y, y_pred)

#Grafico la matriz de confusión
sns.heatmap(tabla,cmap='GnBu',annot=True,fmt='g')
plt.xlabel('Predicted')
plt.ylabel('True')
plt.show()
```

```
              precision    recall  f1-score   support

         0.0       0.67      0.81      0.73      5336
         1.0       0.60      0.48      0.54      5327
         2.0       0.75      0.73      0.74      5332

    accuracy                           0.68     15995
   macro avg       0.67      0.68      0.67     15995
weighted avg       0.67      0.68      0.67     15995

Accuracy: 0.6756486402000625
```



[16]:
```
# Guardamos el modelo
import pickle

filename = 'voting.sav'
pickle.dump(voting, open(filename, 'wb'))
```

####Conclusiones voting

El accuracy es aceptable, consideramos que no se sobreajusta al modelo, pero nos gustaría que se ajustara un poco más a los datos de entrenamiento.

0.0.4 Staking

[17]:
```
ds_train_y = de_temp_train['property_price'].copy()
ds_test_y = de_temp_test['property_price'].copy()
```

[18]:
```
base_models = estimators = [('xgboost',xgb.XGBRegressor()),('knnr',␣
 ↪KNeighborsRegressor()),('gradient_boosting_regressor',␣
 ↪GradientBoostingRegressor())]

#Modelo para definir que modelo usar
meta_model = RandomForestRegressor(n_estimators=5,random_state=42)

#Guarda los scores de cada modelo
model_scores = defaultdict()
```

```
for base_model in base_models:
    model_name = base_model[0]
    model = base_model[1]

    print('Evaluating {}'.format(model_name))
    scores = evaluate_model(model, ds_train_x, ds_train_y)
    model_scores[model_name] = scores

reg = StackingRegressor(
     estimators=base_models,
     final_estimator = meta_model, passthrough=True, cv=5, verbose=2)


stacking_scores = evaluate_model(reg, ds_train_x, ds_train_y)

#Agregamos stacking para poder graficar luego
model_scores['stacking'] = stacking_scores
```

```
Evaluating xgboost

[Parallel(n_jobs=3)]: Using backend LokyBackend with 3 concurrent workers.
[Parallel(n_jobs=3)]: Done  10 out of  10 | elapsed:  1.6min finished
[Parallel(n_jobs=3)]: Using backend LokyBackend with 3 concurrent workers.

Evaluating knnr

[Parallel(n_jobs=3)]: Done  10 out of  10 | elapsed:  1.4min finished
[Parallel(n_jobs=3)]: Using backend LokyBackend with 3 concurrent workers.

Evaluating gradient_boosting_regressor

[Parallel(n_jobs=3)]: Done  10 out of  10 | elapsed:  2.0min finished
[Parallel(n_jobs=3)]: Using backend LokyBackend with 3 concurrent workers.
[Parallel(n_jobs=3)]: Done  10 out of  10 | elapsed: 22.4min finished
```

[19]:
```
reg.fit(ds_train_x, ds_train_y)
y_pred = reg.predict(ds_test_x)
```

```
[23:26:17] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear
is now deprecated in favor of reg:squarederror.

[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.

[23:26:48] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear
is now deprecated in favor of reg:squarederror.

[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    9.8s remaining:    0.0s

[23:26:58] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear
is now deprecated in favor of reg:squarederror.
[23:27:07] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear
```

```
is now deprecated in favor of reg:squarederror.
[23:27:17] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear
is now deprecated in favor of reg:squarederror.
[23:27:27] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear
is now deprecated in favor of reg:squarederror.

[Parallel(n_jobs=1)]: Done   5 out of   5 | elapsed:   48.6s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    5.4s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   5 out of   5 | elapsed:   37.2s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:   14.9s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   5 out of   5 | elapsed:  1.3min finished
```

[20]:
```
mse = metrics.mean_squared_error(y_true = ds_test_y, y_pred  = y_pred, squared␣
 ↪= True)
rmse = metrics.mean_squared_error(y_true = ds_test_y, y_pred = y_pred, squared␣
 ↪= False)

print(f"El error (mse) de test es: {mse}")
print(f"El error (rmse) de test es: {rmse}")
```

```
El error (mse) de test es: 11906547693.986822
El error (rmse) de test es: 109117.12832542296
```

[21]:
```
#Grafico performance
plot_results(model_scores, name='stacking_model_cv')
```

[23]:
```
# Guardamos el modelo
filename = 'stacking.sav'
pickle.dump(reg, open(filename, 'wb'))
```

####Conclusiones stacking

Obtuvimos buenos resultados de los modelos particulares, exceptuando KNeighbors. Aprendiendo de lo visto entre todos los modelos pudimos generar resultados finales mejores a los obtenidos por cualquier modelo de manera independiente.

###Conclusiones generales de ensambles de modelos

El stacking muestra claramente que para los datos que tenemos se puede ajustar mucho mejor con un ensamble de modelos, que Voting. De igual manera, hay que tener en cuenta que para hacer stacking usamos métodos que nos dieron muy buenos resultados en el TP1, principalmente los de gradient boosting.

En general, en comparacion con los demas modelos de regresion, Stacking tiene un RMSE mucho menor a XGBoost y a Redes Neuronales ya sean multicapa o de una sola capa.

Y en comparacion a los demas modelos de clasificacion, Voting tiene un accuracy superior pero no por un margen muy grande.

En conclusion nos sorprendio los resultados de Stacking para realizar regresiones, pero al mismo tiempo es bastante logico teniendo en cuenta el enfoque que tiene para el entrenamiento.