

Predicción precios vivienda Madrid

Proyecto de Productivización

Índice

Briefing

Procesamiento datos (ETL)

Demo

Cronograma

Modelado (ML)

Conclusiones

Equipos

Front-end App

Briefing

- Desarrollar una PoC (proof of concept) para un proyecto de Data Science.
- No se busca que esté perfecta, sino mostrar el potencial del proyecto.
- Es preciso trabajar con PySpark para la limpieza de los datos, con un poco de comprensión de la nube donde está el cluster así como alojar el modelo en alguna plataforma (Pythonanywhere).
- El modelado en sí no es clave y no es necesario obtener scorings excelentes.





Principales Hitos

1

17-19 dic.

Definición de equipos
Brainstorming
Búsqueda de datos
Elección tema

3

21 dic.

Intercambio de archivos
Validación de recursos
Testing

2

20-21 dic.

Procesamiento de datos con Spark
Modelado y análisis en la nube con Databricks
Desarrollo Front-end

4

22 dic.

Documentación
Presentación
Demo

Objetivo del proyecto

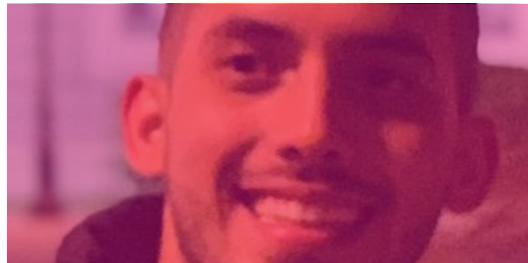


Siempre resulta difícil valorar nuestros bienes, y más si se trata de nuestro hogar.

Por eso te ofrecemos de manera inmediata y en un solo *click*, analizar las variables y características únicas de tu vivienda, que te ayudarán a tener la mejor tasación.



Grupo A





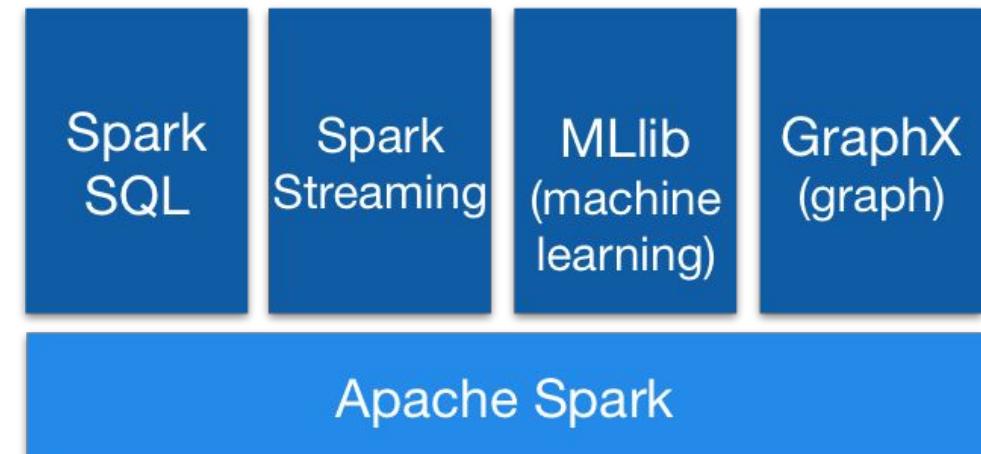
Procesamiento de datos con Spark



1. ¿Qué es Spark?



Apache Spark es una librería que se utiliza para procesamiento de datos en Big Data y Machine Learning ya que está diseñado para ser rápido y dividir las operaciones para poder distribuir los trabajos en un cluster de varios nodos.





2. Importación de datos

Importing pyspark and pandas

```
import pyspark
import pandas as pd
```

Importing SparkSession

```
from pyspark.sql import SparkSession

spark=SparkSession.builder.appName('Practise').getOrCreate()

#df_pyspark_madrid=spark.read.csv('madrid_houses_clean.csv')
```

Reading the CSV

```
df_pyspark_madrid=spark.read.option('header','true').csv('madrid_casas_con_fechas.csv')
```

Checking dtypes of the columns

```
# Checking dtypes of this dataframe
df_pyspark_madrid.dtypes
```

```
[('_c0', 'string'),
 ('id', 'string'),
 ('sq_mt_built', 'string'),
 ('n_rooms', 'string'),
 ('n_bathrooms', 'string'),
 ('n_floors', 'string'),
 ('sq_mt_allotment', 'string'),
 ('floor', 'string'),
 ('buy_price', 'string'),
 ('is_renewal_needed', 'string'),
 ('has_lift', 'string'),
 ('is_exterior', 'string'),
 ('energy_certificate', 'string'),
 ('has_parking', 'string'),
 ('neighborhood', 'string'),
 ('district', 'string'),
 ('house_type', 'string'),
 ('fecha_venta', 'string')]
```

"Madrid Houses Clean": <https://www.kaggle.com/makofe/housesclean>

17 columnas

3.Transformaciones: limpieza

Checking null values

```
# Checking null values
df_pyspark_madrid.printSchema()

root
|-- _c0: string (nullable = true)
|-- id: string (nullable = true)
|-- sq_mt_built: string (nullable = true)
|-- n_rooms: string (nullable = true)
|-- n_bathrooms: string (nullable = true)
|-- n_floors: string (nullable = true)
|-- sq_mt_allotment: string (nullable = true)
|-- floor: string (nullable = true)
|-- buy_price: string (nullable = true)
|-- is_renewal_needed: string (nullable = true)
|-- has_lift: string (nullable = true)
|-- is_exterior: string (nullable = true)
|-- energy_certificate: string (nullable = true)
|-- has_parking: string (nullable = true)
|-- neighborhood: string (nullable = true)
|-- district: string (nullable = true)
|-- house_type: string (nullable = true)
|-- fecha_venta: string (nullable = true)
```

Dropping column called 'Unnamed: 0'

```
df_pyspark_madrid = df_pyspark_madrid.drop('Unnamed: 0')
```

```
#counting the number of rows
```

```
df_pyspark_madrid.count()
```

21739



4.Transformaciones: división (I)

Checking for number of rows in floor values -5, -3, -2, -1

```
#checking out houses that has floor of -5  
df_pyspark_madrid.select('floor').where(df_pyspark_madrid.floor == -5).count()  
  
#there are 1938 number of rows with a floor value of -5. Equates to 8% of total dataframe. In this case our suggestion is to delete them.
```

1938

```
#checking out houses that has floor of -3  
df_pyspark_madrid.select('floor').where(df_pyspark_madrid.floor == -3).count()
```

32

```
#checking out houses that has floor of -2  
df_pyspark_madrid.select('floor').where(df_pyspark_madrid.floor == -2).count()
```

92

```
#checking out houses that has floor of -1  
df_pyspark_madrid.select('floor').where(df_pyspark_madrid.floor == -1).count()
```

2175

Checking for Distinct Values in 'floor' column

```
df_pyspark_madrid.select('floor').distinct().collect()
```

```
[Row(floor='7'),  
 Row(floor='-1'),  
 Row(floor='3'),  
 Row(floor='8'),  
 Row(floor='0'),  
 Row(floor='5'),  
 Row(floor='6'),  
 Row(floor='9'),  
 Row(floor='1'),  
 Row(floor='10'),  
 Row(floor='3'),  
 Row(floor='4'),  
 Row(floor='2'),  
 Row(floor='5'),  
 Row(floor='2')]
```

4.Transformaciones: división (II)

Creating a new Dataframe where housetype is only 2, Equates to Single Family homes.

```
df_pyspark_madrid_only_house_type2 = updated_df_madrid_spark.where("house_type==2")
```

```
df_pyspark_madrid_only_house_type2.show()
```

_c0	id	sq_mt_built	n_rooms	n_bathrooms	n_floors	sq_mt_allotment	floor	buy_price	is_renewal_needed	has_lift	fecha_venta
85	21657	150.0	2	1	2	0.0	-5	159000	True	False	01-17 20:38:31
378	21364	240.0	4	3	3	0.0	-5	580000	False	False	

4.Transformaciones: convertir a pandas

From Spark to Pandas

```
df_pyspark_madrid_only_house_type2_pandas = df_pyspark_madrid_only_house_type2.toPandas()
```

```
df_pyspark_madrid_only_house_type2_pandas.head()
```

_c0	id	sq_mt_built	n_rooms	n_bathrooms	n_floors	sq_mt_allotment	floor	buy_price	is_renewal_needed
0	85	21657	150.0	2	1	2	0.0	-5	159000
1	378	21364	240.0	4	3	3	0.0	-5	580000
2	402	21340	242.0	4	3	4	22.0	-5	380000
3	403	21339	166.0	3	1	3	166.0	-5	300000
4	531	21211	274.0	8	3	3	157.0	-5	495000

4.Transformaciones: precio medio por distrito

Mapping districts with std deviation in prices

```
# Mapping districts with std deviation in prices
pisos_df['std_deviation_price_district'] = pisos_df['district'].map(dict(pisos_df.groupby('district')['buy_price'].std()))
pisos_df.head()
```

C:\Users\FZ\AppData\Local\Programs\Python37\lib\site-packages\ipykernel_launcher.py:2: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

	sq_mt_built	n_rooms	n_bathrooms	n_floors	sq_mt_allotment	floor	buy_price	is_renewal_needed	has_lift	is_exterior	energy_certificate	has_parking
0	64.000	2	1	1	0.000	3	85000		0	0	1	4
1	70.000	3	1	1	0.000	4	129900		1	1	1	0
2	94.000	2	2	1	0.000	1	144247		0	1	1	0
3	64.000	2	1	1	0.000	-1	109900		0	1	1	0
4	108.000	2	2	1	0.000	4	260000		0	1	1	0

4.Transformaciones: dummies tipo vivienda

Pisos

```
# Dataframe 1 - Pisos
pisos = [1,3,4,5]
pisos_df = df[df.house_type.isin(pisos)]  
  
# Mapping districts with mean prices
pisos_df['mean_price_district'] = pisos_df['district'].map(dict(pisos_df.groupby('district')['buy_price'].mean()))
pisos_df
```

```
# Dummies
pisos_dummies = pd.get_dummies(pisos_df['house_type'])
pisos_con_dummies = pd.concat([pisos_df, pisos_dummies], axis=1)
pisos_con_dummies.drop(['house_type'], axis=1, inplace=True)
pisos_con_dummies
```



Grupo B





Modelado y análisis en la nube con Databricks

1. ¿Qué es Databricks?



Herramienta online conectada con servidores de tipo cloud (Azure, AWS, Google, etc.) que se utilizan para procesar y realizar transformaciones de grandes volúmenes de datos en tiempo real, **Big Data**.





2. Importar datos

The screenshot displays three panels of the Databricks web interface. The left panel shows a sidebar with options like Machine Learning, Create, Workspace, Recents, Search, Data, Compute, Jobs, and Experiments. The middle panel shows a sub-menu for 'Create' with options Notebook, Table, Cluster, Experiment, and Model. The right panel shows a 'Workspace' page with options for Workspace, Shared, and Users. A context menu is open over the 'Create' button in the workspace header, listing options: Create, Import (which is highlighted), Export, Permissions, Copy Link Address, and Sort.

Machine Learning

Provide Feedback

Notebook

Create a notebook for querying, transforming, and analyzing data.

Table

Cluster

Experiment

Model

Machine Learning

Home

Workspace

Shared

Users

Guide: Training

with a tutorial on tuning ML models.

Create

Import

Export

Permissions

Copy Link Address

Sort

Release notes



3. Ejecutar transformaciones y entrenar modelos

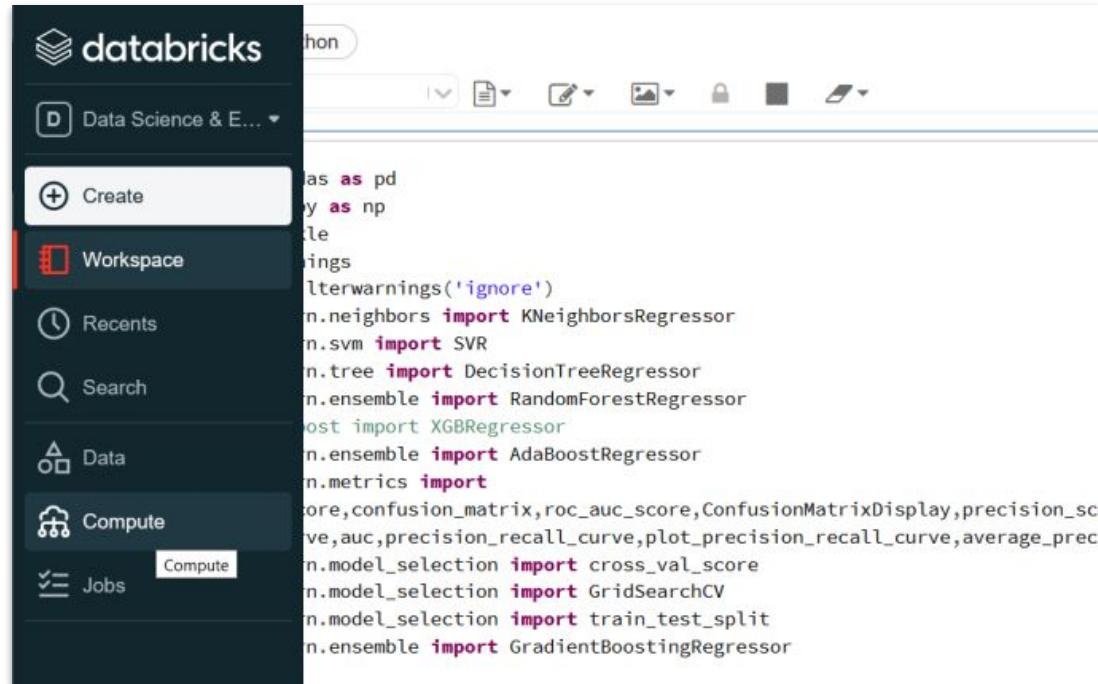
The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** Shows the notebook name "prov_pisos" and the language "Python".
- Toolbar:** Includes icons for file operations (New, Open, Save, etc.), cluster management ("My Cluster"), and other notebook-specific functions.
- Code Cell:** Contains Python code for importing various machine learning and evaluation modules from the "sklearn" library, including KNeighborsRegressor, SVR, DecisionTreeRegressor, RandomForestRegressor, XGBRegressor, AdaBoostRegressor, and GradientBoostingRegressor. It also imports metrics like accuracy_score, confusion_matrix, roc_auc_score, etc., and cross-validation tools like GridSearchCV and train_test_split.
- Cell Status:** The cell status bar indicates "Cmd 1" and "Python".
- Execution Buttons:** A prominent orange arrow points to the "Run all cells in this notebook" button in the toolbar.

```
import pandas as pd
import numpy as np
import pickle
import warnings
warnings.filterwarnings('ignore')
from sklearn.neighbors import KNeighborsRegressor
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
# from xgboost import XGBRegressor
from sklearn.ensemble import AdaBoostRegressor
from sklearn.metrics import
accuracy_score,confusion_matrix,roc_auc_score,ConfusionMatrixDisplay,precision_score,recall_score,f1_score,classification_report,roc_curve,p
lot_roc_curve,auc,precision_recall_curve,plot_precision_recall_curve,average_precision_score
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import train_test_split
from sklearn.ensemble import GradientBoostingRegressor
import os
```

4. Analizar el entorno de producción (I)

Dentro del área **Compute** del menú lateral podremos ver las características de la instancia del nodo maestro de nuestro cluster donde se encuentra toda la información sobre su hostname, trabajadores, stages, etc. En definitiva, cómo organiza y distribuye el trabajo.



The screenshot shows the Databricks workspace interface. The left sidebar has a dark theme with white text. The 'Compute' section is currently selected, indicated by a blue background. Other sections like 'Create', 'Workspace', 'Recents', 'Search', 'Data', and 'Jobs' are visible but not selected. The main area shows a snippet of Python code for machine learning imports:

```
import pandas as pd
import numpy as np
from sklearn import neighbors
from sklearn import svm
from sklearn import tree
from sklearn import ensemble
from sklearn import metrics
from sklearn import cross_validation
from sklearn import grid_search
from sklearn import model_selection
from sklearn import train_test_split
from sklearn import gradient_boosting_regressor
```



4. Analizar el entorno de producción (II)

Compute

All-purpose clusters Job clusters

+ Create Cluster

Name	State	Nodes	Runtime	Driver	Worker	Creator	Actions
My Cluster	Pending	0	9.1 LTS (includes Apache Spark 3.1.2, Scal...	Communi...	Communi...	outronomillo...	0
My Cluster	Terminated	-	9.1 LTS (includes Apache Spark 3.1.2, Scal...	Communi...	Communi...	outronomillo...	0
My Cluster	Terminated	-	9.1 LTS (includes Apache Spark 3.1.2, Scal...	Communi...	Communi...	outronomillo...	0

1 - 3 of 3 < > 20 / Page Go

Clusters / My Cluster

My Cluster

Edit Clone Restart Terminate Delete

Configuration Notebooks Libraries Event log Spark UI Driver Logs Metrics Apps Spark cluster UI - Master

Detach

Name	Status	Last Command Run	Location
prov_pisos	Running	Tue, Dec 21, 2021, 20:11:52 GMT+1 by outronomillo@gmail.com	/Users/outronomillo@gmail.com/pisos_pr/prov_pisos

4. Analizar el entorno de producción (III)

The screenshot shows the Apache Spark Cluster UI interface. On the left is a sidebar with icons for Clusters, Jobs, Stages, Storage, Environment, Executors, SQL, JDBC/ODBC Server, and Structured Streaming. The main area displays the following information:

- Clusters / My Cluster**: Shows a green dot next to "My Cluster". Action buttons include Edit, Clone, Restart, Terminate, and Delete.
- Configuration**, **Notebooks**, **Libraries**, **Event log**, **Spark UI**, **Driver Logs**, **Metrics**, **Apps**, and **Spark cluster UI - Master** (selected).
- Hostname: ec2-52-42-170-95.us-west-2.compute.amazonaws.com** and **Spark Version:9.1.x-scala2.12** are highlighted with red boxes.
- Storage** tab selected. A sub-section titled **Parquet IO Cache** is highlighted with a red box. Below it is a table with the following data:

Data Read from External Filesystem (All Formats)	Data Read from IO Cache (Cache Hits)	Data Written to IO Cache (Cache Misses)	Others
0.0 B	0.0 B	0.0 B	0.0 B



4. Analizar el entorno de producción (IV)

The screenshot shows the Databricks Cluster UI for the cluster "My Cluster". The left sidebar contains navigation icons for Clusters, Notebooks, Libraries, Event log, Spark UI, Driver Logs, Metrics, Apps, and a plus sign for creating new clusters. The main header shows "Clusters / My Cluster" and the cluster name "My Cluster" with actions: Edit, Clone, Restart, Terminate, and Delete.

The top navigation bar includes Configuration, Notebooks, Libraries, Event log, Spark UI, Driver Logs, Metrics, Apps, and "Spark cluster UI - Master". Below this, the hostname "Hostname: ec2-52-42-170-95.us-west-2.compute.amazonaws.com" and spark version "Spark Version: 9.1.x-scala2.12" are displayed.

The "Environment" tab is selected in the sub-navigation bar, which also includes Jobs, Stages, Storage, Executors, SQL, JDBC/ODBC Server, and Structured Streaming. The table lists environment variables:

Variable	Value
libraryDownload.timeoutSeconds	180
spark.akka.frameSize	256
spark.app.id	local-1640114148259
spark.app.name	Databricks Shell
spark.app.startTime	1640114143583
spark.cleaner.referenceTracking.blocking	false
spark.databricks.acl.client	com.databricks.spark.sql.acl.client.SparkSqlAclClient
spark.databricks.acl.provider	com.databricks.sql.acl.ReflectionBackedAclProvider
spark.databricks.acl.scim.client	com.databricks.spark.sql.acl.client.DriverToWebappScimClient
spark.databricks.cloudProvider	AWS



4. Analizar el entorno de producción (V)

En este apartado podemos ver todas las características de sus nodos trabajadores.



Clusters / My Cluster

My Cluster

Hostname: ec2-52-42-170-95.us-west-2.compute.amazonaws.com Spark Version: 9.1.x-scala2.12

Jobs Stages Storage Environment Executors SQL JDBC/ODBC Server Structured Streaming

Executors

Show Additional Metrics

Summary

RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	S	V
Active(1) 0	0.0 B / 3.9 GB	0.0 B	8	0	0	0	0	0.0 ms (0.0 ms)	0.0 B	0.0 B	0	0
Dead(0) 0	0.0 B / 0.0 B	0.0 B	0	0	0	0	0	0.0 ms (0.0 ms)	0.0 B	0.0 B	0	0
Total(1) 0	0.0 B / 3.9 GB	0.0 B	8	0	0	0	0	0.0 ms (0.0 ms)	0.0 B	0.0 B	0	0

Executors

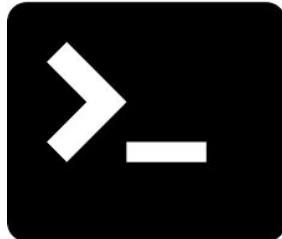
Show 20 entries Search:

Executor ID	Address	Status	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write
driver	10.172.244.3:40591	Active	0	0.0 B / 3.9 GB	0.0 B	8	0	0	0	0	0.0 ms (0.0 ms)	0.0 B	0.0 B	0.0 B

Showing 1 to 1 of 1 entries



5. Terminal Databricks (I)



Para acceder a la terminal online que ofrece databricks, dentro del área de trabajo o workspace, en el desplegable superior, pulsaremos “Terminal”

The screenshot shows the Databricks workspace interface. At the top, there's a navigation bar with a dropdown menu, a file icon, an edit icon, and a view standard icon. Below the navigation bar, the title "prov_pisos Python" is displayed, followed by a cluster selection dropdown set to "My Cluster". A red box highlights this dropdown. To the right of the dropdown is a "File" button, an "Edit" button, and a "View: Standard" button. Below the title, the text "Attached cluster:" is shown, followed by a list of attached clusters: "My Cluster" (15.25 GB | 2 Cores | DBR 9.1 LTS | Spark 3.1.2 | Scala 2.12), "Detach", "Restart Cluster", "Detach & Re-attach", "Spark UI", "Driver Logs", and a red box highlighting the "Terminal" link. Under the "Detach & Attach:" section, there are two entries: "My Cluster" (DBR 9.1 LTS | Spark 3.1.2 | Scala 2.12) and another entry for "My Cluster" (DBR 9.1 LTS | Spark 3.1.2 | Scala 2.12). The main area shows a code editor with the following Python imports:

```
10 # from xgboost import XGBRegressor
11 from sklearn.ensemble import AdaBoostRegressor
12 from sklearn.metrics import
accuracy_score,confusion_matrix,roc_auc_score,ConfusionMatrixDisp
13 from sklearn.model_selection import cross_val_score
14 from sklearn.model_selection import GridSearchCV
15 from sklearn.model_selection import train_test_split
16 from sklearn.ensemble import GradientBoostingRegressor
17 import os
```

At the bottom of the code editor, the text "... Waiting to run..." is visible.

5. Terminal Databricks (II)

Welcome to Databricks web terminal! Please read the following instructions:

- * This terminal session is **ephemeral**, so it will go away if you close or refresh the browser tab.
- * If you want to have a persistent terminal session on this cluster, please use '**tmux**'.
- * There is an idle timeout if no client- or server-side changes are made to the terminal session. Refreshing the tab would launch a new session.

```
root@1222-082914-uzx1dy61-10-172-182-131:/databricks/driver# ls
conf  eventlogs  ganglia  logs  metastore_db  preload_class.lst
root@1222-082914-uzx1dy61-10-172-182-131:/databricks/driver# 
```

```
root@1222-082914-uzx1dy61-10-172-182-131:/databricks# cd spark/
root@1222-082914-uzx1dy61-10-172-182-131:/databricks/spark# ls
HADOOP_VERSION  R  VERSION  assembly  bin  conf  dbconf  python  sbin  scripts
root@1222-082914-uzx1dy61-10-172-182-131:/databricks/spark# 
```

6. Instalar librerías

The screenshot shows the Databricks UI interface for managing clusters. On the left, there's a sidebar with various icons. The main area shows a cluster named "My Cluster". The "Libraries" tab is selected. A modal window titled "Install library" is open, with the "PyPI" tab selected under "Library Source". The "Package" field contains "PyPi package (simplejson or simplejson==3.8.0)". A red box highlights this section. Below it, a message says "Package is a required field". The modal has an "Optional" repository field. In the background, the main cluster page shows a table of installed libraries. One entry, "xgboost", is highlighted with a red box. The table columns are Name, Type, Status, and Source. The "Status" column for "xgboost" shows "Installing".

Clusters / My Cluster

My Cluster

Edit Clone Restart Terminate Delete

Configuration Notebooks Libraries

Uninstall Install new

Name

Install library

Library Source

Upload DBFS/S3 PyPI Maven CRAN Workspace

PyPi package (simplejson or simplejson==3.8.0)

Package is a required field

Repository

Optional

No lib

Clusters / My Cluster

My Cluster

Edit Clone Restart Terminate Delete

Configuration Notebooks Libraries Event log Spark UI Driver Logs Metrics Apps Spark cluster UI - Master

Uninstall Install new

Name	Type	Status	Source
xgboost	PyPI	Installing	



Grupo C





Desarrollo Front-end de la App en Flask



1. ¿Qué es Flask?

Flask es un “micro” Framework escrito en Python y concebido para facilitar el desarrollo de Aplicaciones Web bajo el patrón **MVC** (forma de trabajar que permite diferenciar y separar lo que es el modelo de datos, datos que normalmente están guardados en BD, la vista - página HTML y el controlador, donde se gestionan las peticiones de la app web).



Ventajas

- Proporciona una estructura del proyecto, es decir, todas las Apps que estén construidas con Flask van a tener los mismos elementos y los mismos ficheros.
- Facilita la colaboración.
- Es fácil encontrar bibliotecas adaptadas al Framework.



Aplicación web basada en Flask

Archivos:

- 01 | main.py
- 02 | index.html
- 03 | main.css
- 04 | modelos de predicción



Main.py

Lo primero se bajan las librerías necesarias y se define el alcance global del archivo

```
# Libraries

from flask import Flask, render_template, request, redirect, url_for
import pickle

app = Flask(__name__)
app.config["DEBUG"] = True
```



```
@app.route('/', methods = ['GET', 'POST'])
def home():

    if request.method == 'GET':
        return render_template("index.html", price_list=price)

    try :
        house_type = request.form['house_type']
        sq_mt_built = request.form['square_meters']
        n_rooms = request.form['n_rooms']
        n_bathrooms = request.form['n_bathrooms']
        is_renewal_needed = request.form['renewal']
        has_parking = request.form['parking']

        if house_type == "Piso":
            model = pickle.load(open('./model/pisos_Rand_For.model', 'rb'))
            print('piso')
            prediction = model.predict([[sq_mt_built, n_rooms, n_bathrooms, is_renewal_needed, has_parking]])
            print(prediction)
            pred = ["Precio: " + str(round(prediction[0], 2))+" €"]

        elif house_type == "Chalet":
            model = pickle.load(open('./model/chalets_Rand_For.model', 'rb'))
            print('chale')
            prediction = model.predict([[sq_mt_built, n_rooms, n_bathrooms, is_renewal_needed, has_parking]])
            pred = ["Precio: " + str(round(prediction[0], 2))+" €"]

    except:
        pred = ["Valores erróneos o falta alguno de ellos."]

    try:
        price.pop()
    except:
        pass

    price.append(pred)
    return redirect(url_for('home'))
```

Se definen los diferentes métodos para recibir la data

Se inicializa el código

```
if __name__ == '__main__':
    app.run()
```





HTML

Se crea la cabecera y se definen las fuentes

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Precios de vivienda</title>

  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.2/dist/sandstone/bootstrap.min.css"
integrity="sha384-zEpAL7W11eTKeoBJK1g79kg19qjP7g84KfK3AZsuonx38n8ad+f5ZgXtoSDxPOh" crossorigin="anonymous">

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KyZXEAg3QhqLMpG8r+8fhAXLRk2vvoc2f3B09zVXn8CA5QIVfZOJ3BCsw2P0p/We" crossorigin="anonymous">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Dosis:wght@700&display=swap" rel="stylesheet">

  <link rel="stylesheet" href="{{url_for('static', filename='main.css')}}">

</head>
```

En el cuerpo se indican las variables a desplegar

```
<body>
  <main class = "container p-4"> <!--construye una estructura de 4 columnas-->
    <div class="d-grid gap-4">
      <h1 class="display-4 text-center mt-4 titulo">Calculador de precio de viviendas</h1>
      <div class="row row-cols-1 row-cols-md-2">
        <div class="col-md-4 offset-md-4 my-auto" style="text-align: center;"> <!--ocupará 4 columnas del espacio, centrado-->
          <div class="card" style="width: 180px; margin: auto;"> <!--creacion del objeto card-->
            <div class="card-header" style="background-color: #f0f0f0; border-bottom: 1px solid #ccc; padding: 5px; text-align: center;">
              <form action"." method="post">

                <div class="form-group" style="margin-bottom: 10px;">
                  <input type="text" name = "house_type" class="form-control" placeholder="Piso o Chalet">
                </div>

                <div class="form-group" style="margin-bottom: 10px;">
                  <input type="number" name = "square_meters" class="form-control" placeholder="Metros cuadrados">
                </div>

                <div class="form-group" style="margin-bottom: 10px;">
                  <input type="number" name="n_rooms" class="form-control" placeholder="Número de habitaciones">
                </div>

                <div class="form-group" style="margin-bottom: 10px;">
                  <input type="number" name="n_bathrooms" class="form-control" placeholder="Número de baños">
                </div>
```



Se crea el botón de Precio para la predicción y se cierra el cuerpo

```
<button type="submit" class="btn btn-dark btn-block">Precio
</button>

</form>

</div>

<div class="card-body">

    <ul>
        {% for price in price_list: %}
            <li>
                {{price[0]}}
            </li>
        {% endfor %}
    </ul>
</div>
</div>
</div>

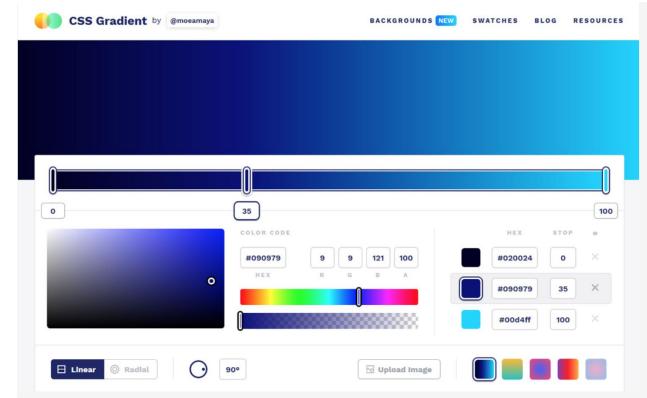
</main>
</body>
</html>
```

CSS

Es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado.

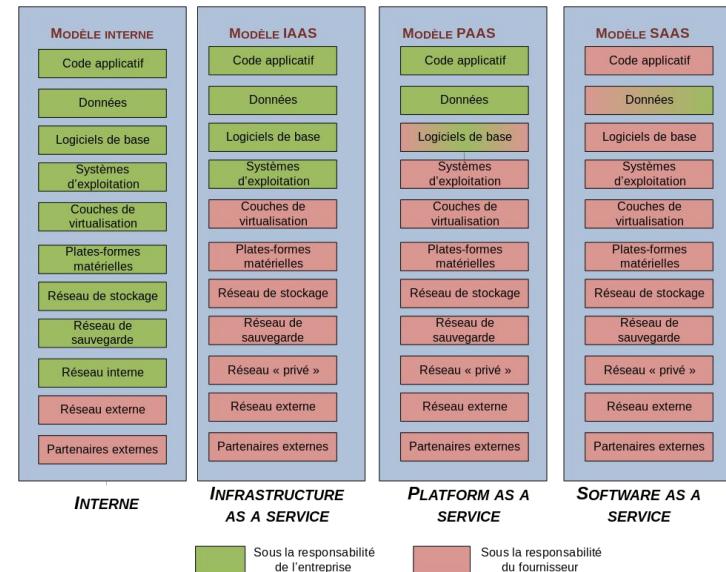
```
.titulo{  
font-family: 'Dosis', sans-serif;  
}
```

```
body{  
background: #FDC830; /* fallback for old browsers */  
background: -webkit-linear-gradient(to right, #F37335, #FDC830); /* Chrome 10-25, Safari 5.1-6 */  
background: linear-gradient(to right, #F37335, #FDC830); /* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+ */  
}
```



1. ¿Qué es Pythonanywhere?

Es un entorno de desarrollo integrado en línea (IDE) y un servicio de alojamiento web (PaaS) basado en el lenguaje de programación Python.





The screenshot shows the PythonAnywhere dashboard. At the top, there are links for Dashboard, Consoles, Files, Web, Tasks, and Databases. Below this, the "Recent" section includes "Consoles" (with a link to "Bash console 22717298" and a "View all" button), "Files" (with a link to "/home/datasience2109thebridge/my_flask_app.py" and a "View all" button), "Notebooks" (with a message: "Your account does not support Jupyter Notebooks. Upgrade your account to get access!" and a "Open Web tab" button), and "Web apps" (with a link to "datasience2109thebridge.pythonanywhere.com" and a "Open Web tab" button). A "New console:" section allows adding another file or browsing files. A note at the bottom says: "You can have up to 2 consoles. To get more, upgrade your account!"

Luego vamos a la sección de Web, donde podremos especificar el nombre de host en la URL que las personas ingresarán para ver el sitio web. Seleccionamos el framework que queremos usar.

<https://www.pythonanywhere.com/user/datasience2109thebridge/files/home/datasience2109thebridge/app>

Creamos una cuenta en [www_pythonanywhere.com](https://www.pythonanywhere.com),
(gratuita - “Beginner”).

The screenshot shows the configuration page for a new web app named "datasience2109thebridge.pythonanywhere.com". It includes a "Reload" button and a "Configuration for datasience2109thebridge.pythonanywhere.com" section. Below this, there's a "Best before date:" field set to "Monday 21 March 2022" with a "Run until 3 months from today" button. A note states: "Paying users' sites stay up forever without any need to log in to keep them running." The "Traffic:" section shows "This month (previous month) 20 (0)".



The screenshot shows the pythonanywhere file manager interface. At the top, there is a navigation bar with links for Send feedback, Forums, Help, Blog, Account, and Log out. Below the navigation bar, the main header includes the pythonanywhere logo, the current path /home/datasience2109thebridge, and tabs for Dashboard, Consoles, Files, Web, Tasks, and Databases. A message indicates that the account is 21% full with 108.7 MB used out of 512.0 MB quota, with a link to More Info.

The interface is divided into two main sections: Directories and Files. In the Directories section, there is a text input field "Enter new directory name" and a button "New directory". Below this, a list of existing directories is shown, including ".cache/", ".local/", ".virtualenvs/", and "mysite/". There are also icons for creating a new directory and deleting existing ones. A yellow "Upload a file" button with a 100MB maximum size limit is located at the bottom of this section.

In the Files section, there is a text input field "Enter new file name, eg hello.py" and a button "New file". Below this, a list of files is shown, including ".bash_history", ".bashrc", ".gitconfig", ".profile", ".pythonstartup.py", ".vimrc", and "README.txt", each with download and delete icons. The file sizes and last modified dates are also listed.

A large, semi-transparent modal window is overlaid on the main interface, showing a detailed view of the "mysite" directory. This modal has its own navigation bar and file list, mirroring the structure of the main page but focusing on the "mysite" folder. It also features a "Upload a file" button with a 100MB maximum size limit.

Se crearon las mismas carpetas y se suben los archivos en Upload a file



Modificamos las rutas de los modelos en el archivo “main” y lo subimos a Github

The screenshot shows a code editor with Python code for a Flask application and a terminal window below it.

Code Editor (main.py):

```
1 # Libraries
2
3 from flask import Flask, render_template, request, redirect, url_for
4 import pickle
5
6 app = Flask(__name__)
7 app.config["DEBUG"] = True
8
9 price = []
10
11 @app.route('/', methods = ['GET', 'POST'])
12 def home():
13
14     # esto hace que nos muestre por pantalla la estructura del HTML con el resultado de la predicción
15     if request.method == 'GET':
16         return render_template("index.html", price_list=price)
17
18     try :
19         # Recogemos los datos introducidos en la aplicación
20         house_type = request.form['house_type']
21         sq_mt_built = request.form['square_meters']
22         n_rooms = request.form['n_rooms']
23         n_bathrooms = request.form['n_bathrooms']
24         is_renewal_needed = request.form['renewal']
25         has_parking = request.form['parking']
26
create mode 100644 __pycache__/_main.cpython-37.pyc
11:03 ~/mysite (master)$ git add .
11:05 ~/mysite (master)$ git commit -m "app def"
[master 301ea91] app def
 2 files changed, 2 insertions(+), 20 deletions(-)
 rewrite __pycache__/_main.cpython-37.pyc (63%)
11:05 ~/mysite (master)$
```

Terminal:

```
create mode 100644 __pycache__/_main.cpython-37.pyc
11:03 ~/mysite (master)$ git add .
11:05 ~/mysite (master)$ git commit -m "app def"
[master 301ea91] app def
 2 files changed, 2 insertions(+), 20 deletions(-)
 rewrite __pycache__/_main.cpython-37.pyc (63%)
11:05 ~/mysite (master)$
```



Reload y hacemos click en Configuration for datascience2109thebridge.pythonanywhere.com

The screenshot shows the PythonAnywhere web interface. At the top, there's a navigation bar with links for Send feedback, Forums, Help, Blog, Account, and Log out. Below the navigation is the PythonAnywhere logo and a dashboard menu with links for Dashboard, Consoles, Files, Web, Tasks, and Databases.

The main content area is titled "Configuration for datascience2109thebridge.pythonanywhere.com". It features a "Reload:" button with a circular arrow icon and a green "Reload datascience2109thebridge.pythonanywhere.com" button. Below this, there's a "Best before date:" section with a note about hosting the site free-of-charge until March 21, 2022, and a "Run until 3 months from today" button. A note about paying users staying up forever without logging in is also present. The "Traffic:" section at the bottom indicates the site is not busy.

<http://datascience2109thebridge.pythonanywhere.com/>

<http://datascience2109thebridge.pythonanywhere.com/>

Cómo usar la App?

Para realizar la tasación de la vivienda solo será necesario introducir 6 parámetros, y hacer click en el botón de Precio.

- 01 | Tipo de Vivienda (Pisos / Chalet)
- 02 | Metros cuadrados
- 03 | Número de habitaciones
- 04 | Número de cuartos de baños
- 05 | Necesita reforma
- 06 | Tiene Parking



Calculador de precio de viviendas

Piso o Chalet

Metros cuadrados

Número de habitaciones

Número de baños

Necesita reforma

Parking

PRECIO

Precio: 221099.04 €



Esta aplicación da respuesta a las dudas de cara a vender la vivienda de un propietario.

Es una aproximación al precio de venta con calidad, ya que actualmente contamos con unos 20.000 registros que analizarán todas las características de la vivienda para dar un precio lo más real posible.

Por ello, nuestra aplicación da respuesta a este inconveniente de manera rápida y sencilla para todos los usuarios en la Comunidad de Madrid.

Por qué nuestra App

Competencias Adquiridas

Hard Skills

- Spark
- Gestión de Github
- Creación de modelos ML en Databricks
- Desarrollo en Flask y Pythonanywhere
- Debuguear

Soft Skills

- Búsqueda de información
- Resolución de problemas
- Trabajo con deadlines
- Pensamiento analítico
- Trabajo en equipo (coordinación, organización, empatía y liderazgo)





Gracias.

