

# Predicción precios vivienda Madrid

Proyecto de Productivización

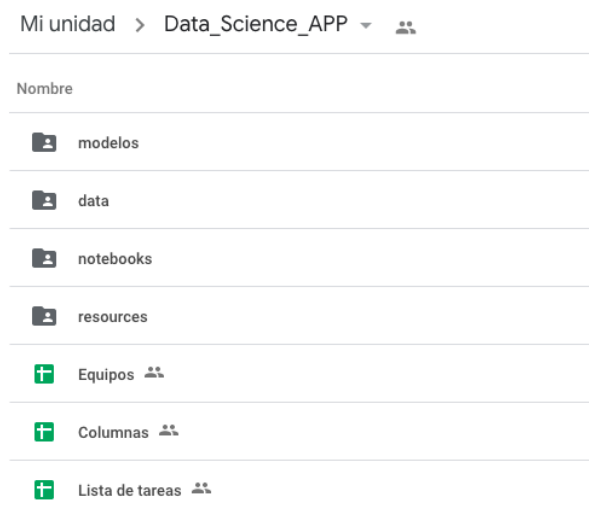
En un mercado tan extenso, con múltiples variables a tomar en cuenta y con amplios rangos de diferenciales en las características de sus bienes, como es el mercado inmobiliario; Siempre resulta difícil valorar viviendas o inmuebles. Por ello, nuestra aplicación da respuesta a este inconveniente de manera rápida y sencilla para todos los usuarios en la Comunidad de Madrid.

Con un solo click y sin intermediarios podrás hacer uso de nuestro producto para obtener respuesta inmediata a todas tus dudas sobre viviendas en Madrid. 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 tenemos datos actualizados diariamente, que actualmente contamos con unos 20000 registros que analizarán todas las características de la vivienda para dar un precio lo más real posible.

## 1. Organización

Se han dividido los grupos en las 3 vertientes que se presentan a continuación y se ha nombrado un manager por grupo que coordine la comunicación entre grupos.

Se ha creado una carpeta en Google Drive compartida con todos los usuarios para subir los recursos y documentos que se van realizando.



En primer lugar, para elegir la temática de la app a desarrollar se propuso un brainstorming conjunto con todos los miembros de los diferentes equipos para proponer temas y buscar datos durante el fin de semana.

El lunes a primera hora se realizó una reunión para votar y definir el tema. A partir de ese momento el trabajo se ha realizado por grupos.

## 2. Obtención de Datos (Grupo A)

Para el siguiente trabajo se ha obtenido un dataframe denominado “Madrid Houses Clean” de la página de Kaggle, mediante la siguiente dirección web: <https://www.kaggle.com/makofe/housesclean>. Este dataset consiste en 17 columnas con la siguiente descripción:

- id: identificador.
- sqmtbuilt: m2 contruidos.
- n\_rooms: número de cuartos.
- n\_bathrooms: número de baños.
- sqmtallotment: m2 de zonas exteriores.
- floor: número de piso.
- isrenewalneeded: booleano.
- has\_lift: booleano.
- is\_exterior: booleano.
- energy\_certificate: valores más altos significan más eficiente.
- has\_parking: booleano.
- neighborhood: barrios de Madrid.
- district: distritos de Madrid.
- house\_type: piso, casa, estudio, dúplex, ático.
- buy\_price: precio de la vivienda.

Por medio de pyspark, tras instalarlo e investigar cómo se debía desarrollar el código, se realizaron las siguientes tareas:

- Instalación pyspark.
- Apertura de sesión y lectura del csv.
- Creación de columna de Tiempo: año, mes, día, hora, minuto y segundos. En la cual se establecía la fecha de venta.
- Eliminación de la columna Unnamed 0.
- Comprobación de valores nulos (sin presencia de nulos).
- Inicialmente, se eliminaron una serie de valores del dataset para reducir ruido pero, posteriormente, pudimos comprobar, tras varios intentos, que al eliminar esos valores, las casas también se borraban del dataset ya que estaban asignadas, precisamente, a esos valores. En consecuencia, consideramos que lo más oportuno era dejar los valores iniciales para poder continuar con la variable “Casas”

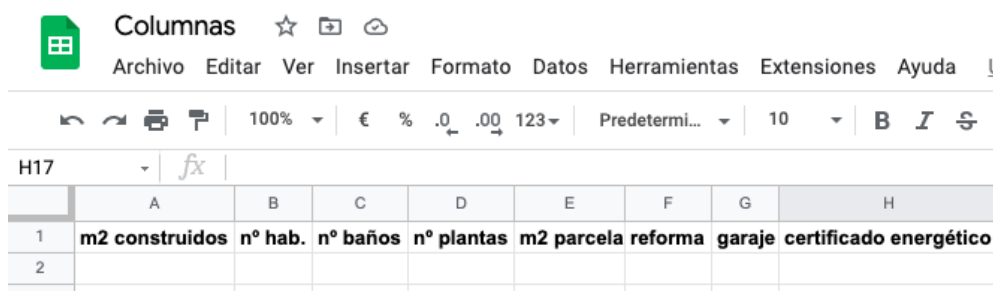
en los siguientes procesos (ya que era un objetivo de nuestro trabajo). De este modo, procedimos a realizar, nuevamente, el procesamiento de los datos.

- Los datos se dividieron en 2 datasets:
  - o Casas.
  - o Pisos: pisos, áticos, estudios y dúplex. Dentro de este dataset se procedió a efectuar una aplicación de get dummies para que se pudieran procesar, correctamente, los datos en la sección de Machine Learning.
- Creación de precio promedio de la vivienda de cada tipo y en cada distrito.
- Creación de columna de desviación del precio de la vivienda de cada tipo y en cada distrito.
- Transformación de los datos ya procesados y limpios a un archivo csv con el fin de que pudieran ser utilizados para las partes posteriores del trabajo.

### 3. Modelado y análisis Databricks (Grupo B)

El primer paso para abordar la parte del modelo ha sido la definición de las columnas necesarias para desarrollar el modelo. Para ello, se crearon dos modelos en XGBoost y Ramdon Forest con el .csv en bruto y los evaluamos. A partir de ahí se vio necesario dividir el dataset y crear dos modelos, uno para chalets y otro para pisos.

Se creó un Excel con la estructura de columnas para cada dataset pero tras hablarlo y coordinarse con el Grupo C, se ha tomado la decisión de simplificar y crear una X común con las mismas columnas para ambos modelos.



|   | A              | B       | C        | D          | E          | F       | G      | H                      |
|---|----------------|---------|----------|------------|------------|---------|--------|------------------------|
| 1 | m2 construidos | nº hab. | nº baños | nº plantas | m2 parcela | reforma | garaje | certificado energético |
| 2 |                |         |          |            |            |         |        |                        |

Después de tener los datos limpios por parte del equipo A, con idas y venidas de transformación necesarias para optimizar los modelos, se realizó un pipeline con varios modelos para evaluar y saber cuál era el mejor modelo de regresión a utilizar para este caso.

Por último, se ha realizado un Grid Search con el Random Forest (el mejor modelo según el pipeline) para mejorarlo, entrenarlo, guardarlo y compartirlo con el Grupo C.

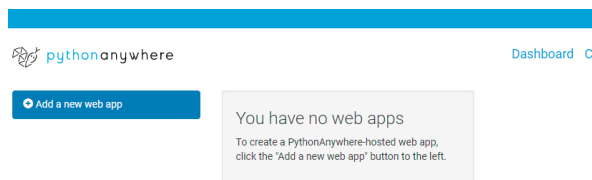
Finalmente, para el análisis de Databricks, se han subido a la nube tanto las bases de datos con las transformaciones y los notebooks para correrlos en la nube y poder dar respuesta a las preguntas:

- Hostname: ec2-54-244-203-15.us-west-2.compute.amazonaws.com
- La nube está en AWS
- Nº Workers: 1
- Tipo de usuario: root
- Ruta del drive en el nodo maestro: /databricks/driver
- Dónde está instalado Spark: /databricks/spark
- Base de datos relacional que usa el nodo maestro: Apache Parquet

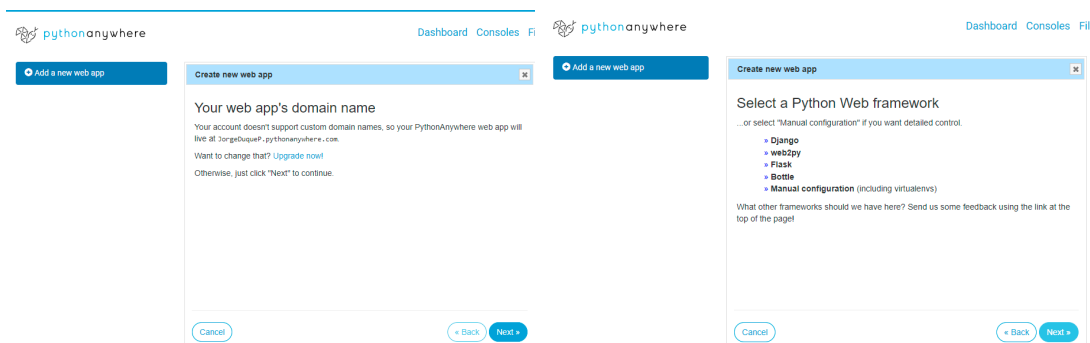
#### 4. USO de la API (Grupo C)

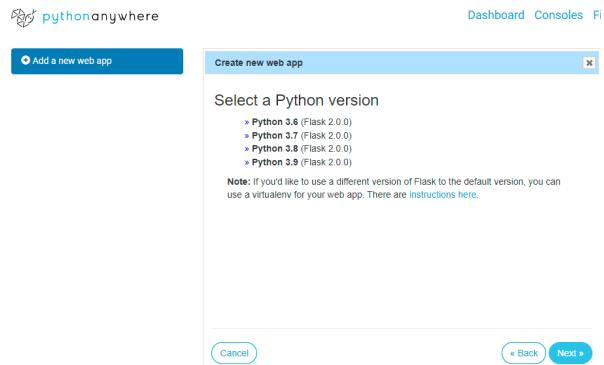
Para la creación de la API hemos seguido los siguientes pasos.

Se crea la web en Pythonanywhere:

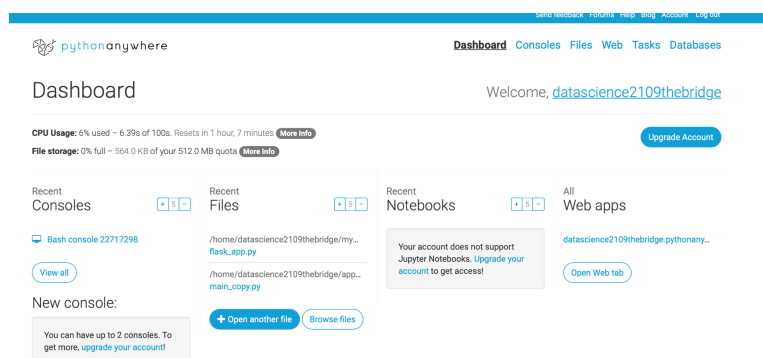
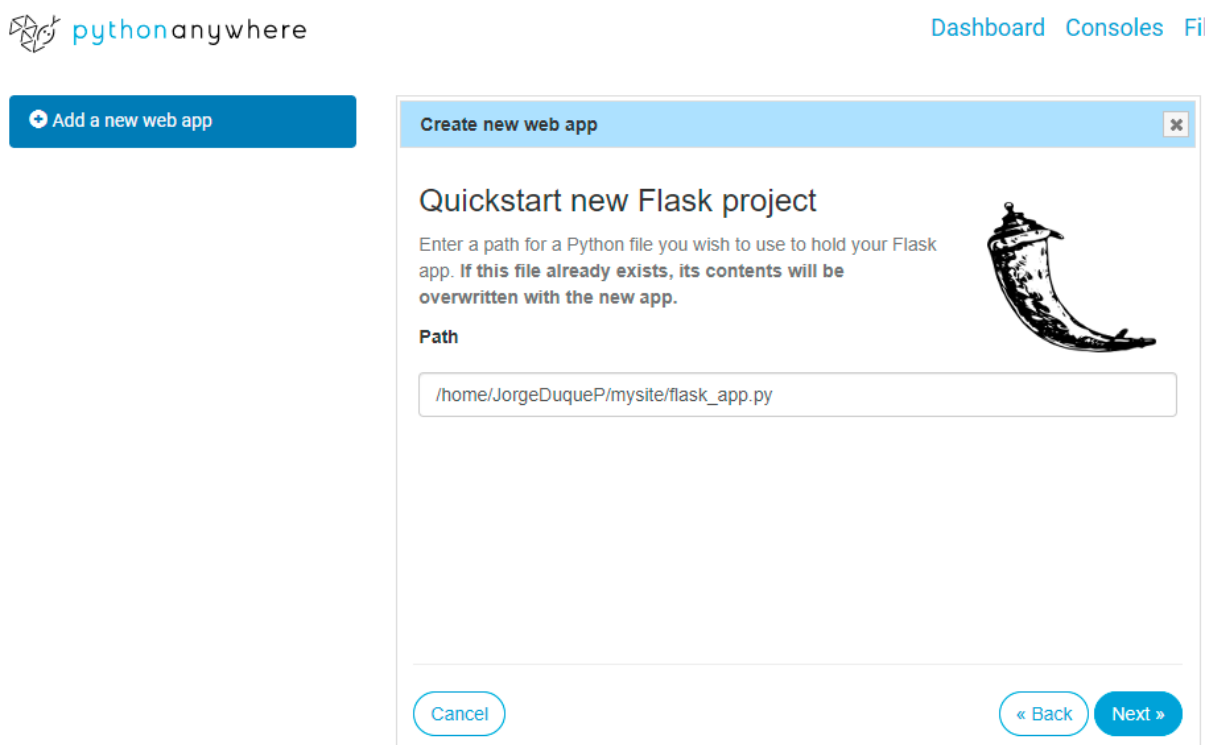


Se van eligiendo las opciones de Flask y Python (la versión que se tenga en local):





Se crea automáticamente una carpeta en la que iremos incluyendo los archivos del proyecto:



Send feedback Forums Help Blog Account Log out

pythonanywhere

Dashboard Consoles Files **Web** Tasks Databases

datascience2109thebridge.pytho...

Add a new web app

### Configuration for datascience2109thebridge.pythonanywhere.com

Reload:

Reload datascience2109thebridge.pythonanywhere.com

Best before date:

We're happy to host your free website – and keep it free – for as long as you want to keep it running, but you'll need to log in at least once every three months and click the "Run until 3 months from today" button below. We'll send you an email a week before the site is disabled so that you don't forget to do that. [See here for more details.](#)

This site will be disabled on **Monday 21 March 2022**

Run until 3 months from today

Paying users' sites stay up forever without any need to log in to keep them running.

Traffic:

How busy is your site?

|                             |           |
|-----------------------------|-----------|
| This month (previous month) | 20 (0)    |
| Traffic (last month)        | 0.0 (0.0) |

## Se crean las carpetas del proyecto:

Send feedback Forums Help Blog Account Log out

pythonanywhere

Dashboard Consoles **Files** Web Tasks Databases

/home/ datascience2109thebridge

Open Bash console here 21% full – 108.7 MB of your 512.0 MB quota [More Info](#)

Directories

Enter new directory name [New directory](#)

cache/

local/

virtualenvs/

mysite/

Files

Enter new file name, eg hello.py [New file](#)

bash\_history

bashrc

gitconfig

profile

pythonstartup.py

vimrc

README.txt

Upload a file

100MB maximum size

Send feedback Forums Help Blog Account Log out

pythonanywhere

Dashboard Consoles **Files** Web Tasks Databases

/home/datascience2109thebridge/ mysite

Open Bash console here 21% full – 108.7 MB of your 512.0 MB quota [More Info](#)

Directories

Enter new directory name [New directory](#)

git/

\_\_pycache\_\_/

model/

static/

templates/

Files

Enter new file name, eg hello.py [New file](#)

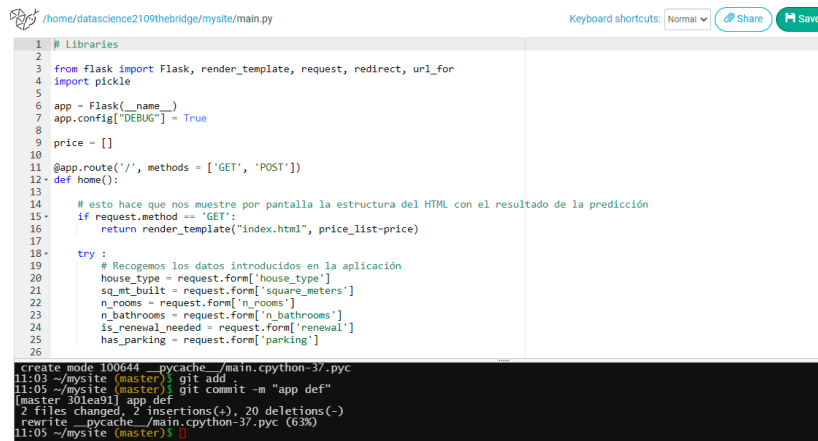
flask\_app.py

main.py

Upload a file

100MB maximum size

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

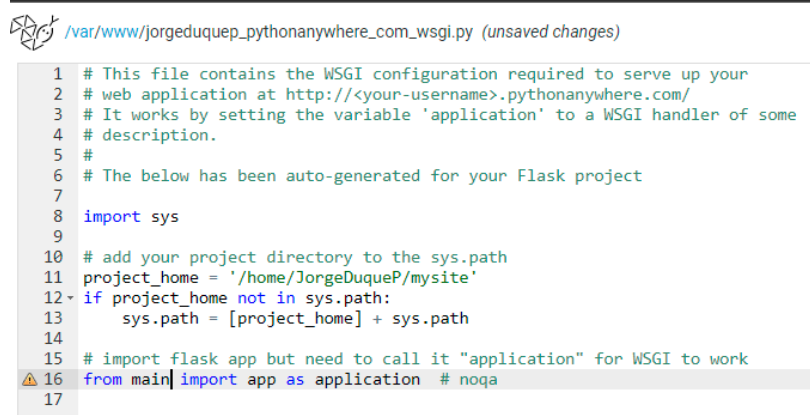


```

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
27 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) $

```

Y debemos de modificar en la última línea para que de importe el archivo app desde “main”.

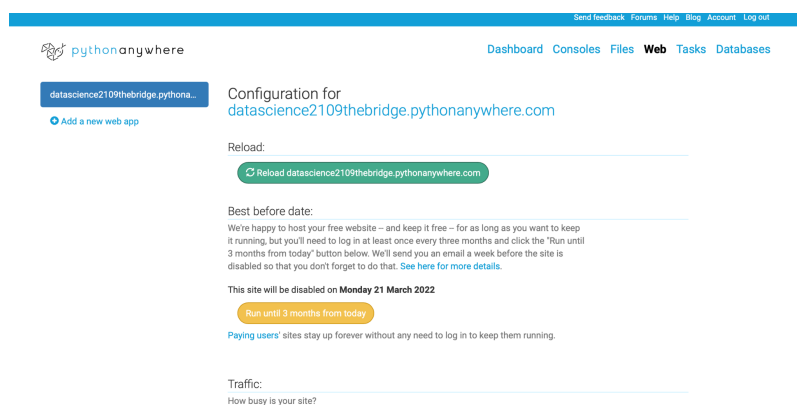


```

1 # This file contains the WSGI configuration required to serve up your
2 # web application at http://<your-username>.pythonanywhere.com/
3 # It works by setting the variable 'application' to a WSGI handler of some
4 # description.
5 #
6 # The below has been auto-generated for your Flask project
7
8 import sys
9
10 # add your project directory to the sys.path
11 project_home = '/home/JorgeDuqueP/mysite'
12 if project_home not in sys.path:
13     sys.path = [project_home] + sys.path
14
15 # import flask app but need to call it "application" for WSGI to work
16 from main| import app as application # noqa
17

```

Realizamos un “reload” para que nos actualice las modificaciones:



Send feedback · Forums · Help · Blog · Account · Log out

pythonanywhere Dashboard Consoles Files Web Tasks Databases

datascience2109thebridge.pythonanywhere.com

➤ Add a new web app

Configuration for  
datascience2109thebridge.pythonanywhere.com

Reload:

↻ Reload datascience2109thebridge.pythonanywhere.com

Best before date:

We're happy to host your free website – and keep it free – for as long as you want to keep it running, but you'll need to log in at least once every three months and click the "Run until 3 months from today" button below. We'll send you an email a week before the site is disabled so that you don't forget to do that. [See here for more details.](#)

This site will be disabled on **Monday 21 March 2022**

Run until 3 months from today

Paying users' sites stay up forever without any need to log in to keep them running.

Traffic:

How busy is your site?

Vamos a la web para ver si todo es correcto y lanzamos predicción:

## Calculador de precio de viviendas



Piso o Chalet

Metros cuadrados

Número de habitaciones

Número de baños

Necesita reforma

Parking

**PRECIO**

- Precio: 476705.6 €

### 5. Documentación y presentación.

Se han creado dos esquemas, uno para la documentación (Grupo C) y otro para la presentación (Grupo A&B) que se han ido completando de forma paralela gracias a la nube de Google Drive.

### 6. Enlaces

Github: [https://github.com/ds2109fulltime/App\\_Viviendas](https://github.com/ds2109fulltime/App_Viviendas)

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