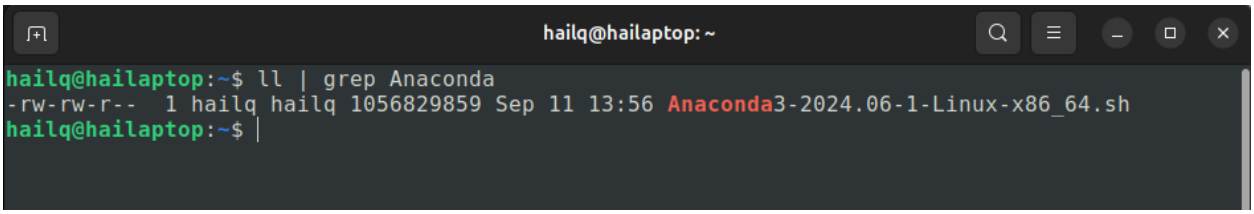
Tutorial 1. Getting Started

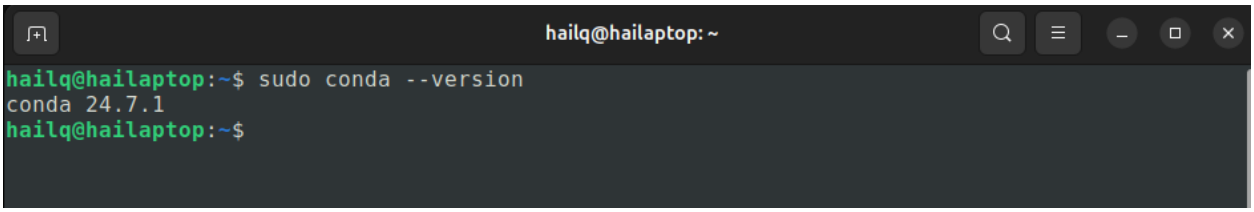
Getting started with Anaconda

Install conda

1. Downloaded the installer package from the website. The screenshot shows there is an executable file for installing conda in the directory.

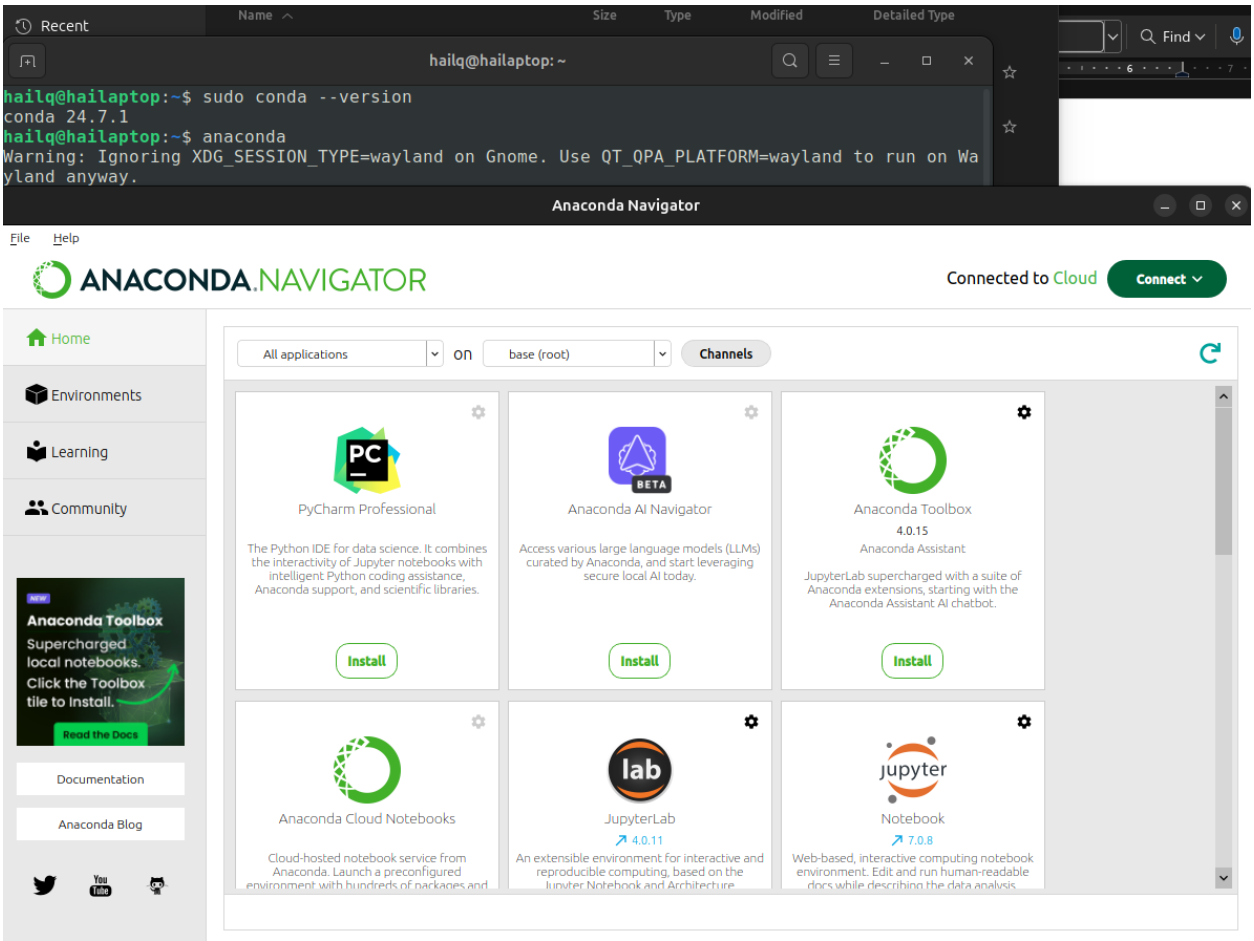


2. Execute the installer, the message below verifies that conda version 24.7.1 has been successfully installed.



Install anaconda

1. Install anaconda-navigator using conda. Having configured a symlink in environment variable to the /anaconda/bin/anaconda-navigator, open the anaconda-navigator via terminal. The screenshot below verifies successful installation



Install jupyter packages via conda

1. Install jupyter via terminal (ubuntu)

```
hailq@hailaptop:~$ conda install jupyter
Channels:
 - defaults
Platform: linux-64
```

2. Create symlink to the jupyter-notebook file in /usr/local/bin so that jupyter notebook can be opened via terminal (ubuntu)

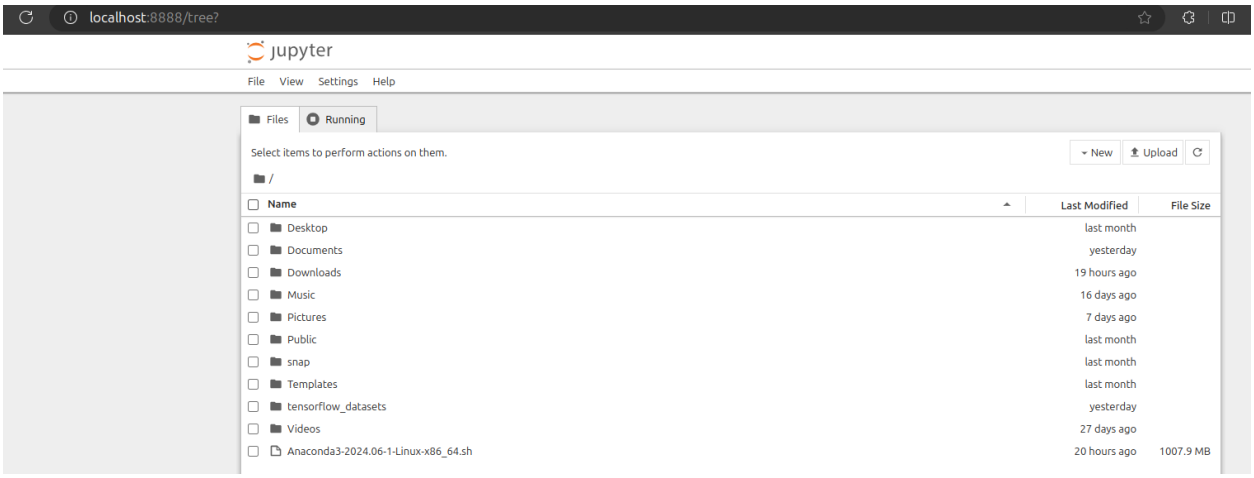
```
hailq@hailaptop:~$ ll /usr/local/bin/ | grep jupyter
lrwxrwxrwx  1 root root   35 Sep 12 10:31 jupyter -> /usr/local/src/anaconda/bin/jupyter*
lrwxrwxrwx  1 root root   44 Sep 12 10:33 jupyter-notebook -> /usr/local/src/anaconda/bin/jupyter-noteb
```

Run Jupyter Notebook

1. Execute jupyter-notebook package

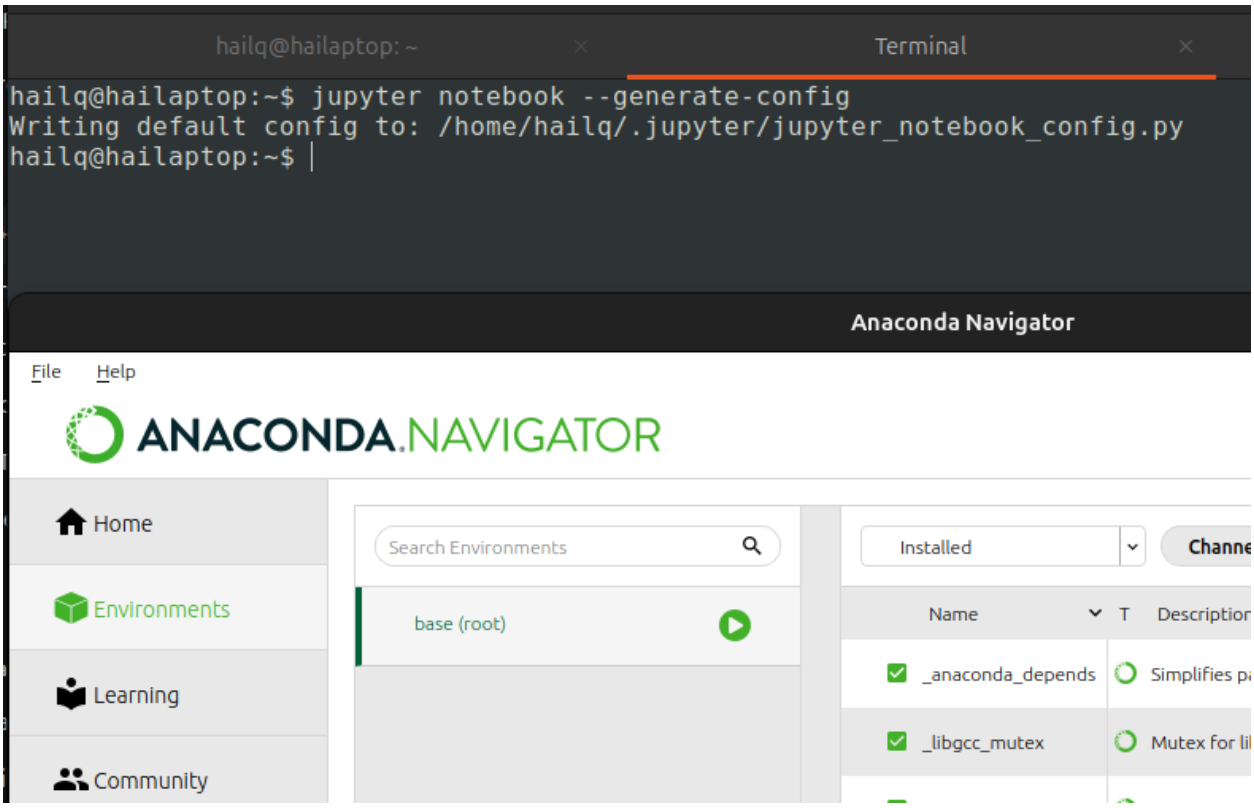
```
hailq@hailaptop:~$ jupyter notebook
[W 2024-09-12 10:53:30.315 ServerApp] A `_jupyter_server_extension_points` funct
ion was not found in jupyter_lsp. Instead, a `_jupyter_server_extension_paths` f
unction was found and will be used for now. This function name will be deprecate
d in future releases of Jupyter Server.
[W 2024-09-12 10:53:30.640 ServerApp] A `_jupyter_server_extension_points` funct
ion was not found in notebook_shim. Instead, a `_jupyter_server_extension_paths`
 function was found and will be used for now. This function name will be depreca
ted in future releases of Jupyter Server.
[I 2024-09-12 10:53:31.384 ServerApp] Extension package panel.io.jupyter_server_
extension took 0.7427s to import
[I 2024-09-12 10:53:31.384 ServerApp] jupyter_lsp | extension was successfully l
inked.
[I 2024-09-12 10:53:31.389 ServerApp] jupyter_server_terminals | extension was s
uccessfully linked.
[I 2024-09-12 10:53:31.394 ServerApp] jupyterlab | extension was successfully li
nked.
[I 2024-09-12 10:53:31.398 ServerApp] notebook | extension was successfully link
ed.
[I 2024-09-12 10:53:31.617 ServerApp] notebook_shim | extension was successfully
 linked.
[I 2024-09-12 10:53:31.617 ServerApp] panel.io.jupyter_server_extension | extens
ion was successfully linked.
```

2. Enter jupter notebook GUI via 127.0.0.1:8888 endpoint

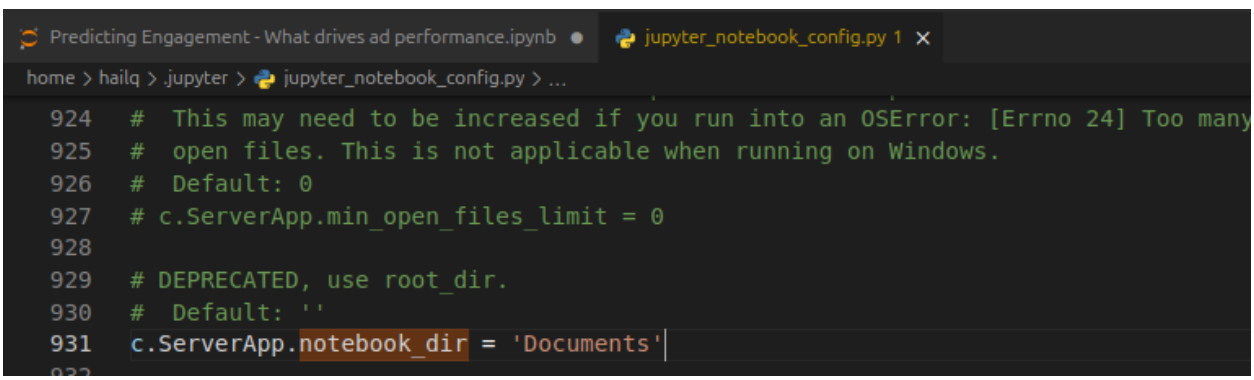


Change Jupyter Notebook start directory

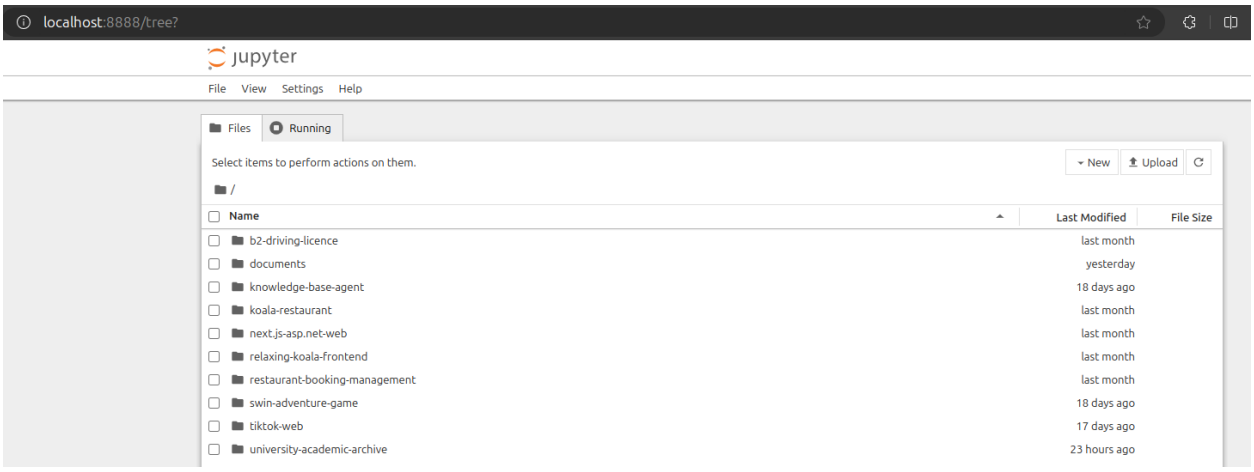
1. Open the Anaconda Navigator and click on Environments -> base(root) -> Open Terminal. Type the command jupyter notebook –generate-config. On ubuntu system, a configuration file is created at ~ which is /home/username/.jupyter/jupyter_notebook_config.py



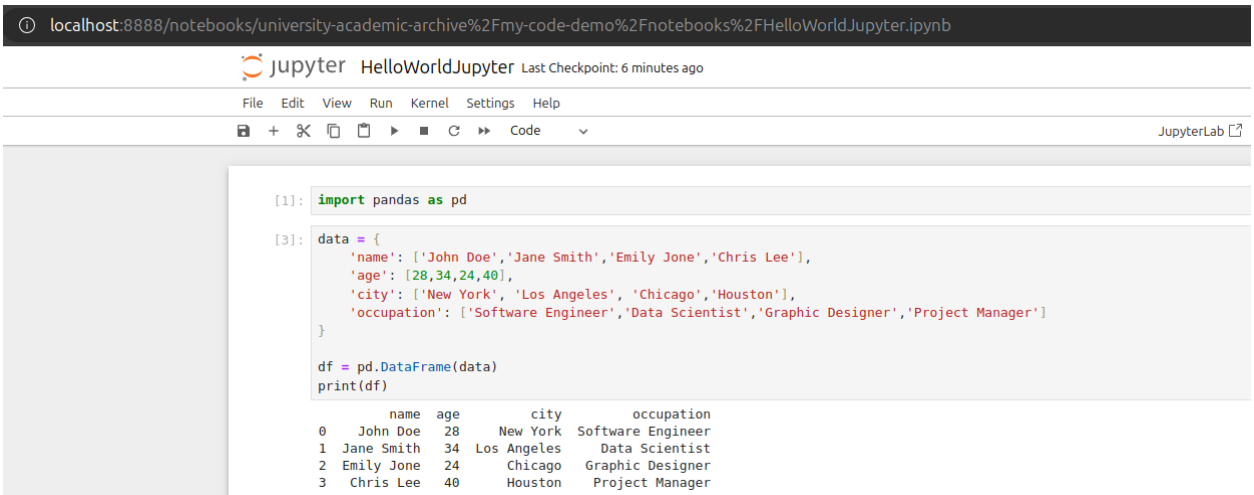
2. Modify the configuration in the jupyter_notebook_config.py to ~/Documents



3. Verify the new starting directory applied to jupyter notebook. We no longer access home directory by default (that contains Documents, Download, Pictures, ...)



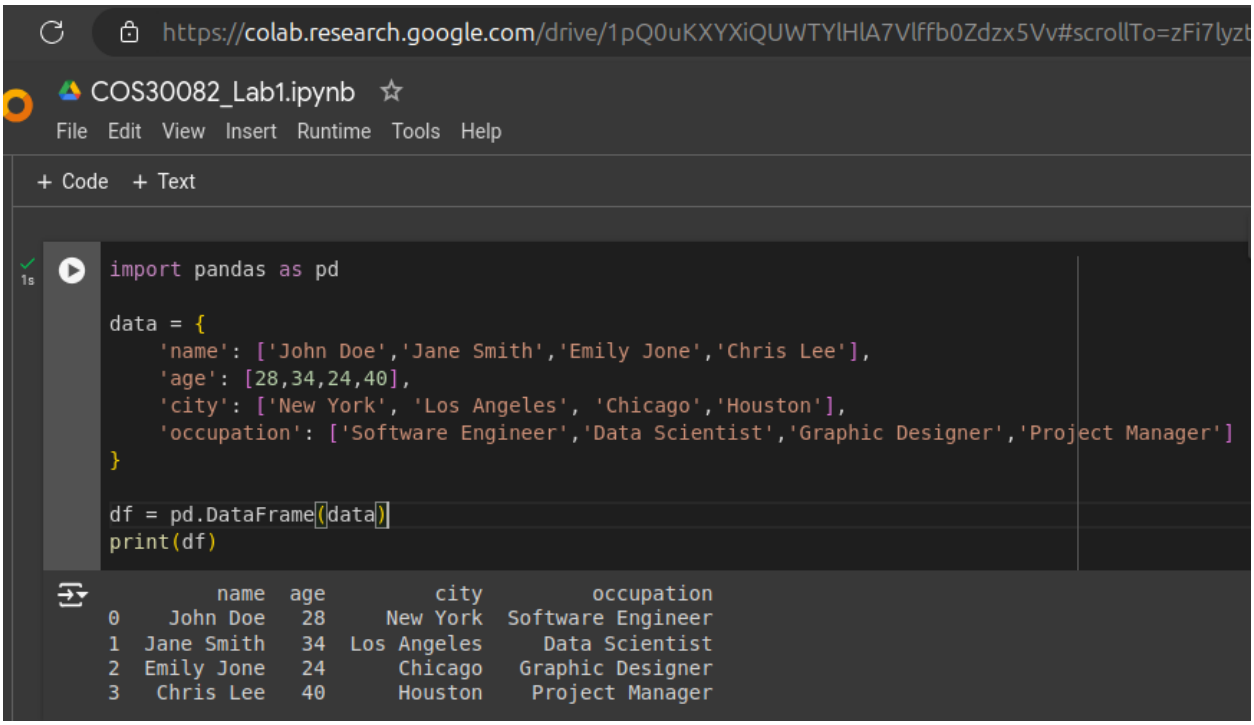
Run some Python code



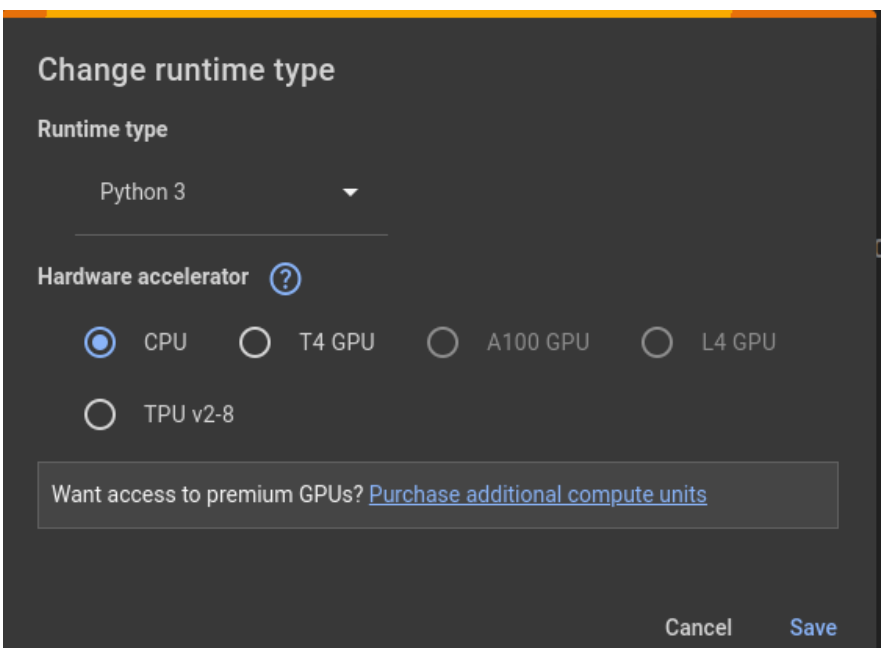
Getting started with Google Collab

Familiarize with the interface

1. Login to Google Collab, create a new Notebook, and write some code then execute the code



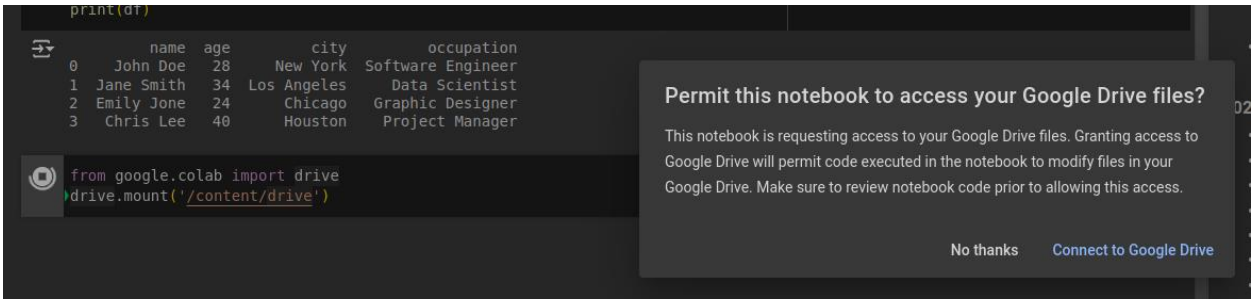
Accessing GPU and TPU resources



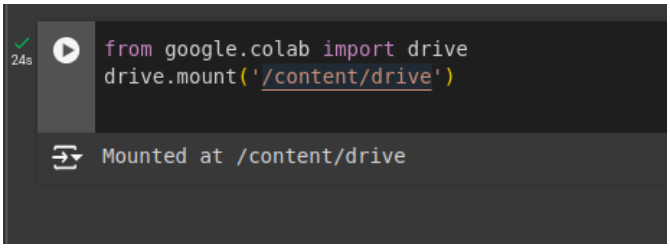
Mount Google Drive

1. Execute the code as instructed

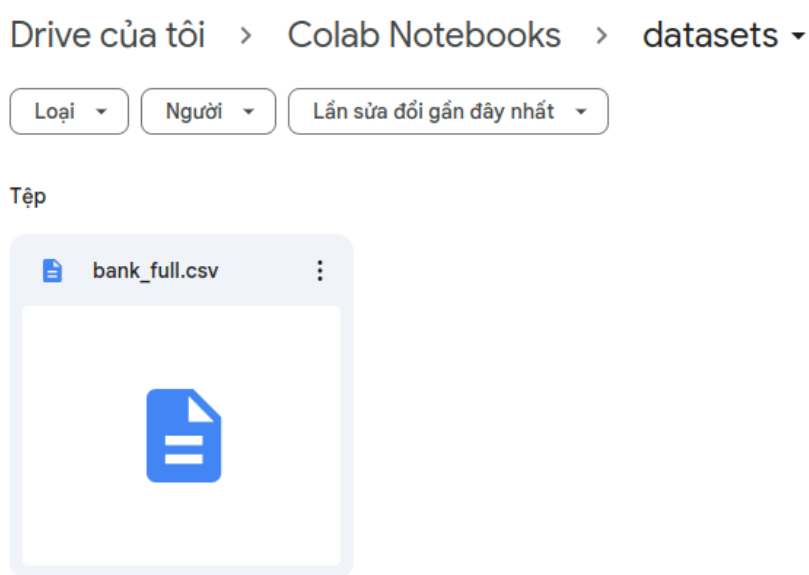
```
from google.colab import drive
drive.mount('/content/drive')
```



2. Verify credentials and authenticate. Verify mounted successfully



3. Now when upload a file to /Collab Notebook/datasets



4. The dataset can be accessed via pandas

```
[3] from google.colab import drive
    drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import pandas as pd

df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/datasets/bank_full.csv')
print(df)
```

```
       age           job   marital  education default  balance housing loan  \
0       58    management   married   tertiary      no     2143     yes   no
1       44    technician    single  secondary      no       29     yes   no
2       33  entrepreneur   married  secondary      no        2     yes  yes
3       47   blue-collar   married    unknown      no     1506     yes   no
4       33       unknown    single    unknown      no        1      no   no
...    ...           ...       ...        ...     ...      ...     ...  ...
45206   51    technician   married   tertiary      no      825      no   no
45207   71       retired  divorced    primary      no     1729      no   no
45208   72       retired   married  secondary      no     5715      no   no
45209   57   blue-collar   married  secondary      no      668      no   no
45210   37  entrepreneur   married  secondary      no     2971      no   no

         contact  day month  duration  campaign  pdays  previous poutcome    y
0        unknown    5   may       261         1     -1         0  unknown   no
1        unknown    5   may       151         1     -1         0  unknown   no
2        unknown    5   may        76         1     -1         0  unknown   no
3        unknown    5   may        92         1     -1         0  unknown   no
4        unknown    5   may       198         1     -1         0  unknown   no
...          ...  ...   ...       ...       ...    ...       ...      ...  ...
45206   cellular   17   nov       977         3     -1         0  unknown  yes
45207   cellular   17   nov       456         2     -1         0  unknown  yes
45208   cellular   17   nov      1127         5    184         3  success  yes
45209  telephone   17   nov       508         4     -1         0  unknown   no
45210   cellular   17   nov       361         2    188        11    other   no
```

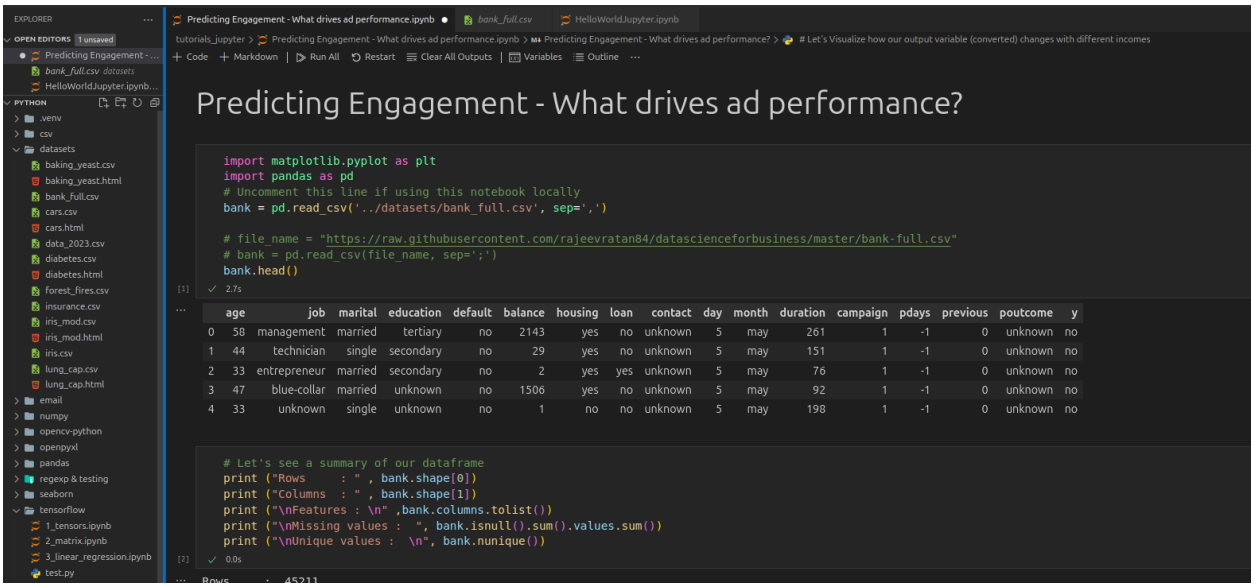
Run some Python code

Images above

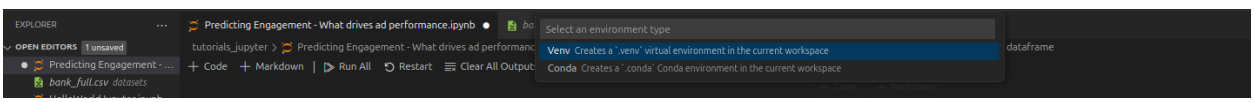
Self-Exploration

Another way of running Python – IDE

1. Is it possible to run a jupyter notebook that is integrated into Visual Studio Code via an extension Python language support.

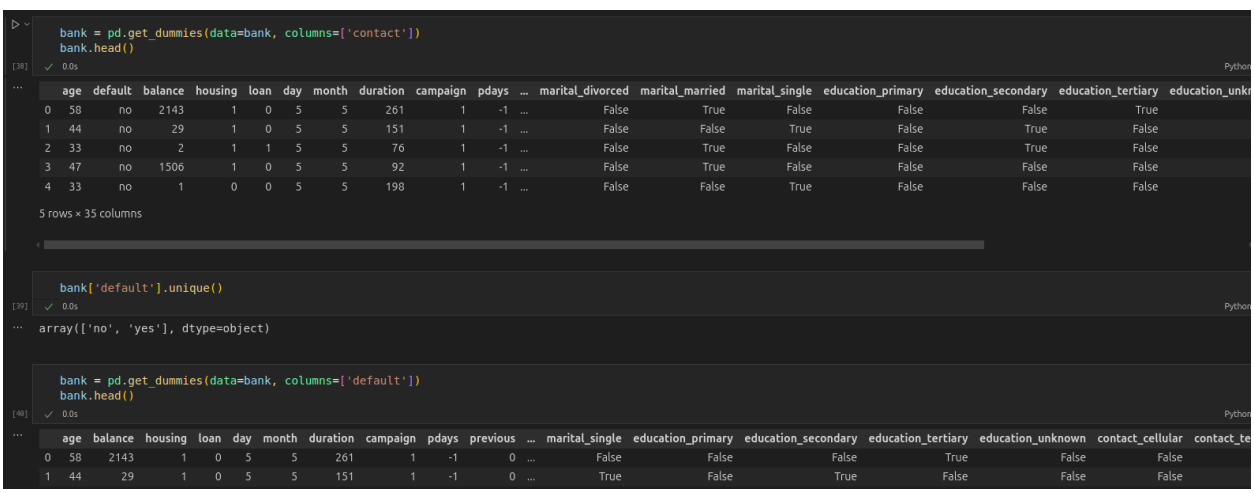


2. The extension allows creation of Python virtual environments with both conda and venv

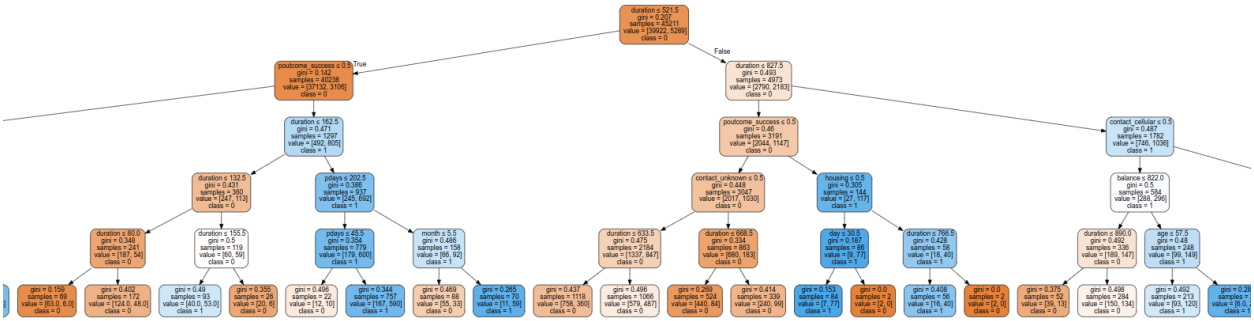


Predicting engagement

1. Run the script one by one to explore the data provided and transform wherever necessary.



2. Train a decision tree model with the refined dataset



Predicting Insurance Premiums

Predicting Insurance Premiums

- Our simple dataset contains a few attributes for each person such as
- Age, Sex, BMI, Children, Smoker, Region and their charges

Aim

- To use this info to predict charges for new customers

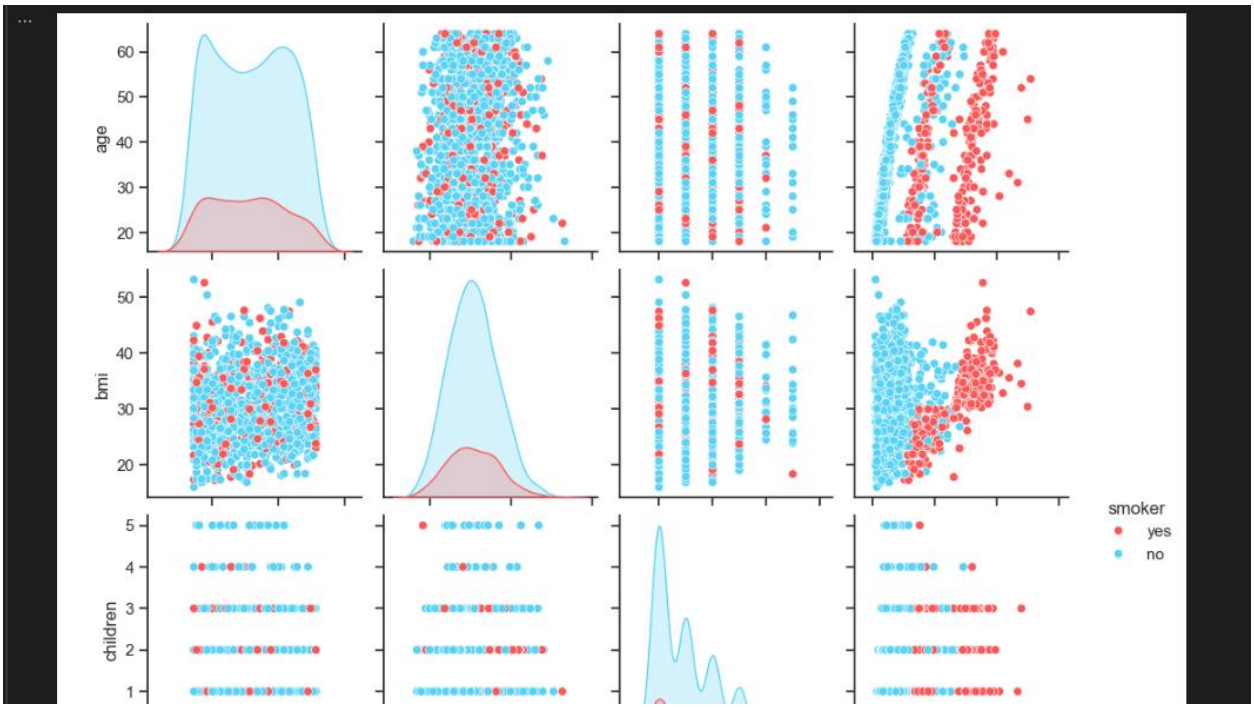
+ Code + Markdown

```python
import pandas as pd

# Uncomment this line if using this notebook locally
insurance = pd.read_csv('../datasets/insurance.csv')

# file_name = "https://raw.githubusercontent.com/rajeevratan84/datascienceforbusiness/master/insurance.csv"
# insurance = pd.read_csv(file_name)

# Preview our data
insurance.head()
```

[2] ✓ 0.0s Python

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200

Observe data features



Train models of different machine learning algorithms and observe effectiveness of each

```
# compare all results in one table
training_accuracies = [accuracy_MLR_train, accuracy_PR_train, accuracy_DTR_train, accuracy_RFR_train, accuracy_SVR_trai
testing_accuracies = [accuracy_MLR_test, accuracy_PR_test, accuracy_DTR_test, accuracy_RFR_test, accuracy_SVR_test]
training_RMSE = [RMSE_MLR_train, RMSE_PR_train, RMSE_DTR_train, RMSE_RFR_train, RMSE_SVR_train]
testing_RMSE = [RMSE_MLR_test, RMSE_PR_test, RMSE_DTR_test, RMSE_RFR_test, RMSE_SVR_test]
cv_accuracies = [accuracy_cv_MLR, accuracy_cv_PR, accuracy_cv_DTR, accuracy_cv_RFR, accuracy_cv_SVR]

parameters = ["fit_intercept=False", "fit_intercept=False", "max_depth=5", "n_estimators=400, max_depth=5", "kernel="li

table_data = {"Parameters": parameters, "Training Accuracy": training_accuracies, "Testing Accuracy": testing_accuracie
              "Training RMSE": training_RMSE, "Testing RMSE": testing_RMSE, "10-Fold Score": cv_accuracies}
model_names = ["Multiple Linear Regression", "Polynomial Regression", "Decision Tree Regression", "Random Forest Regres

table_dataframe = pd.DataFrame(data=table_data, index=model_names)
table_dataframe
```

[46] ✓ 0.0s

	Parameters	Training Accuracy	Testing Accuracy	Training RMSE	Testing RMSE	10-Fold Score
Multiple Linear Regression	fit_intercept=False	-0.489561	-0.324110	14589.307283	14438.166279	0.717113
Polynomial Regression	fit_intercept=False	0.835502	0.880879	4848.265611	4330.562228	0.839107
Decision Tree Regression	max_depth=5	0.869426	0.871194	4319.509663	4503.167202	0.849424
Random Forest Regression	n_estimators=400, max_depth=5	0.878632	0.896906	4164.457267	4028.712787	0.857379
Support Vector Regression	kernel="linear", C=1000	0.652218	0.734317	7049.511742	6467.427432	0.705813