ARCHITECTURE DESIGN DOCUMENT

(BANK MARKETING ANALYTICS – BI PROJECT)

MADHAV KHURANA
MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY, NEW DELHI

VERSION: 1.0
DATED: 07/09/2021

Document Version Control:

Bank Marketing Analytics - Business Intelligence Project

Version	Date	Author	Change
1.0	07/09/21	Madhav Khurana	First version of complete Architecture Design Document

Abstract:

The data is related to direct marketing campaigns (phone calls) of a Portuguese banking institution. The classification goal is to predict if the client will subscribe to a term deposit. The data is related to direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be subscribed or not.

Contents:

1. Introduction:

1.1. Why this Low-Level Design Document?

Any software needs the architectural design to represents the design of software. IEEE defines architectural design as “the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system.” The software that is built for computer-based systems can exhibit one of these many architectures.

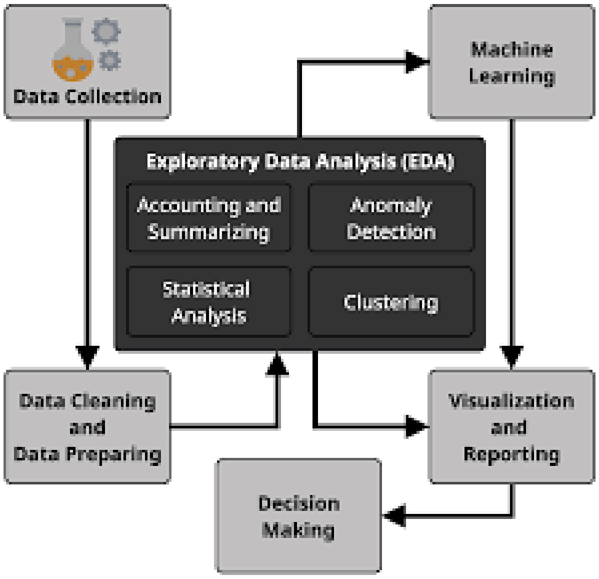
Each style will describe a system category that consists of:

- A set of components (e.g. a database, computational modules) that will perform a function required by the system.
- The set of connectors will help in coordination, communication, and cooperation between the components.
- Conditions that how components can be integrated to form the system.
- Semantic models that help the designer to understand the overall properties of the system.

1.2. Scope

Architecture Design Document (ADD) is an architecture design process that follows a step-by-step refinement process. The process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the design principles may be defined during requirement analysis and then refined during architectural design work.

2. Architecture:

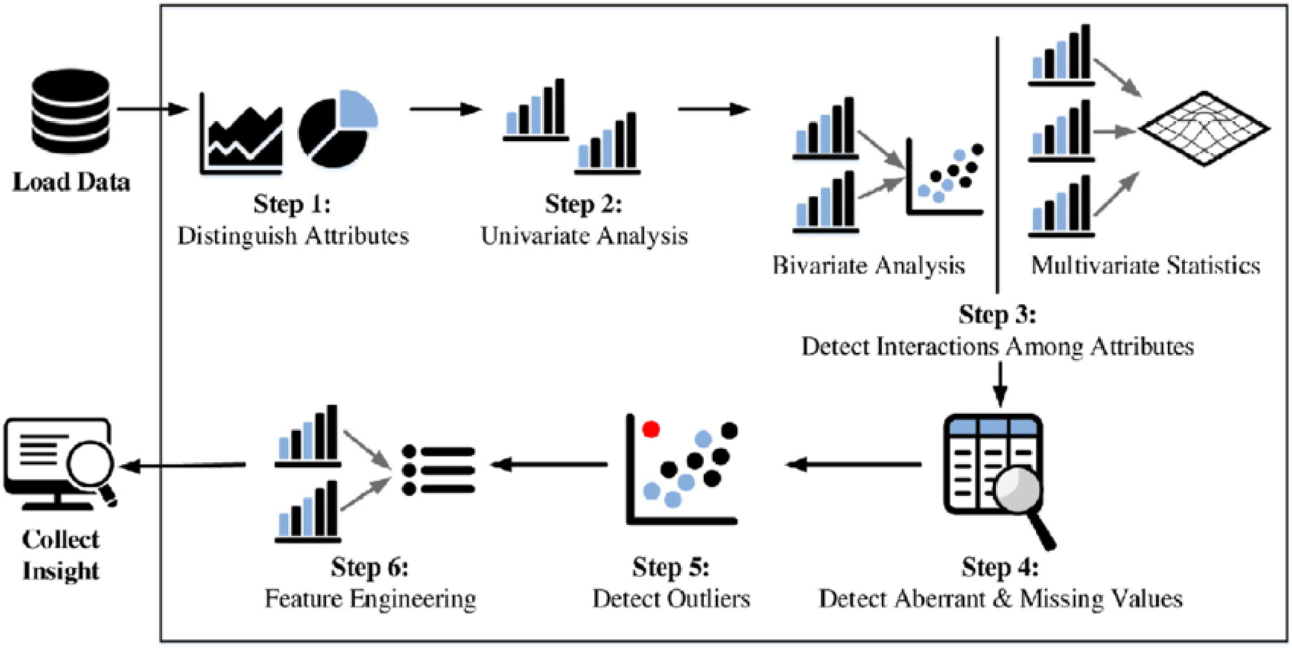

Data Collection
Machine Learning
Exploratory Data Analysis (EDA)
Accounting and Summarizing
Anomaly Detection
Statistical Analysis
Clustering
Data Cleaning and Data Preparing
Visualization and Reporting
Decision Making

EDA in Python uses data visualization to draw meaningful patterns and insights. It also involves the preparation of data sets for analysis by removing irregularities in the data.
Based on the results of EDA, companies also make business decisions, which can have repercussions later.

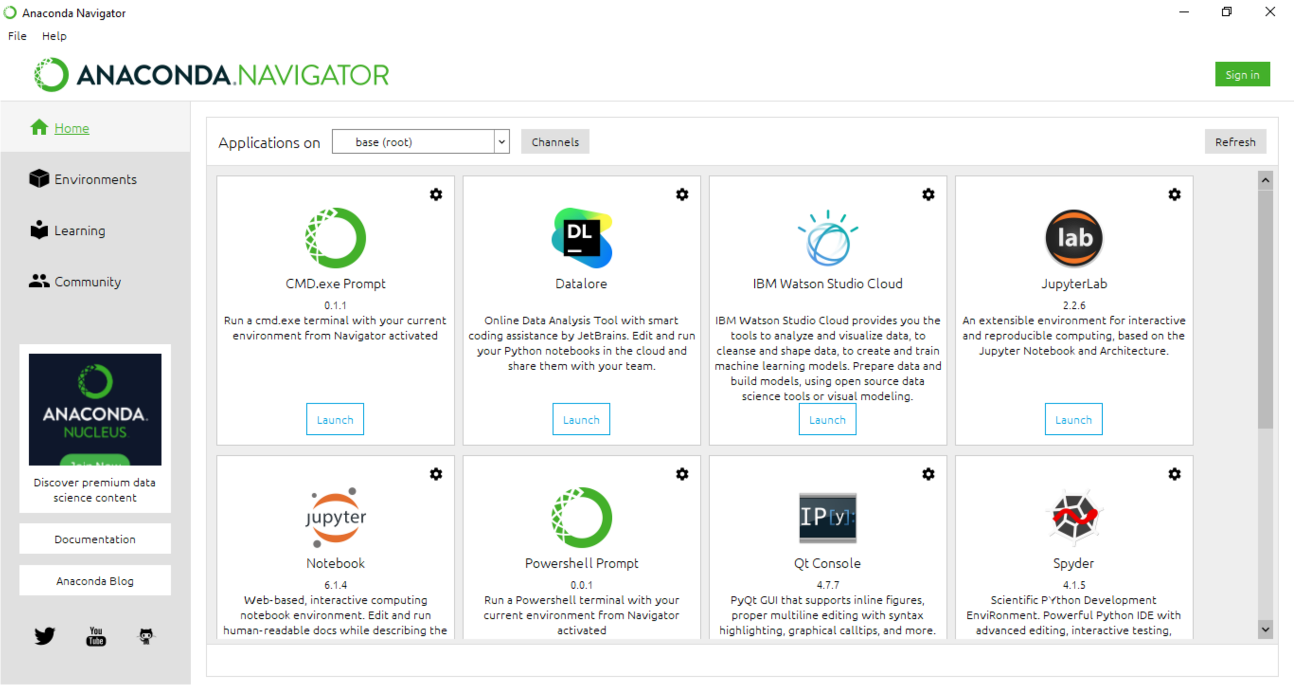
- If EDA is not done properly then it can hamper the further steps in the machine learning model building process.
- If done well, it may improve the efficacy of everything we do next.

Below are following steps to follow for EDA:

1. Data Sourcing
2. Data Cleaning
3. Univariate analysis
4. Bivariate analysis
5. Multivariate analysis



3. Architecture Description:



CMD.exe Prompt
0.1.1
Run a cmd.exe terminal with your current environment from Navigator activated

Datalore
Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.

IBM Watson Studio Cloud
IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

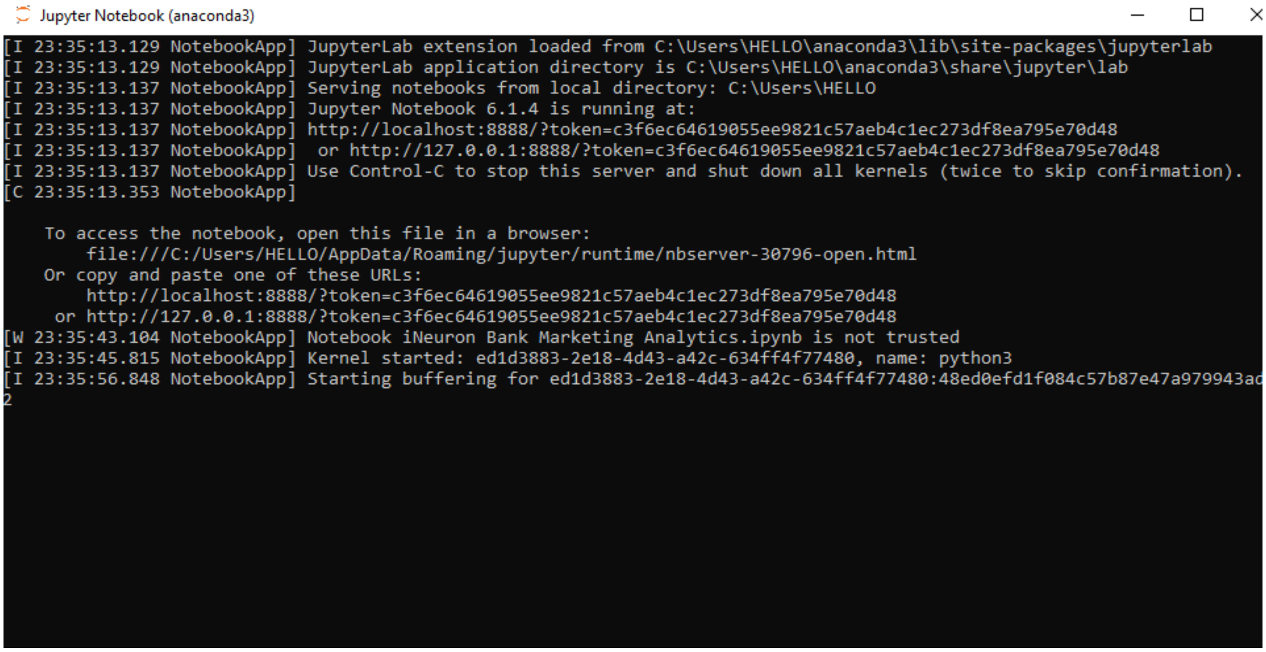
JupyterLab
2.2.6
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Notebook
6.1.4
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the

Powershell Prompt
0.0.1
Run a Powershell terminal with your current environment from Navigator activated

Qt Console
4.7.7
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Spyder
4.1.5
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing,



```
[I 23:35:13.129 NotebookApp] JupyterLab extension loaded from C:\Users\HELLO\anaconda3\lib\site-packages\jupyterlab
[I 23:35:13.129 NotebookApp] JupyterLab application directory is C:\Users\HELLO\anaconda3\share\jupyter\lab
[I 23:35:13.137 NotebookApp] Serving notebooks from local directory: C:\Users\HELLO
[I 23:35:13.137 NotebookApp] Jupyter Notebook 6.1.4 is running at:
[I 23:35:13.137 NotebookApp] http://localhost:8888/?token=c3f6ec64619055ee9821c57aeb4c1ec273df8ea795e70d48
[I 23:35:13.137 NotebookApp]  or http://127.0.0.1:8888/?token=c3f6ec64619055ee9821c57aeb4c1ec273df8ea795e70d48
[I 23:35:13.137 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 23:35:13.353 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///C:/Users/HELLO/AppData/Roaming/jupyter/runtime/nbserver-30796-open.html
    Or copy and paste one of these URLs:
        http://localhost:8888/?token=c3f6ec64619055ee9821c57aeb4c1ec273df8ea795e70d48
     or http://127.0.0.1:8888/?token=c3f6ec64619055ee9821c57aeb4c1ec273df8ea795e70d48
[W 23:35:43.104 NotebookApp] Notebook iNeuron Bank Marketing Analytics.ipynb is not trusted
[I 23:35:45.815 NotebookApp] Kernel started: ed1d3883-2e18-4d43-a42c-634ff4f77480, name: python3
[I 23:35:56.848 NotebookApp] Starting buffering for ed1d3883-2e18-4d43-a42c-634ff4f77480:48ed0efd1f084c57b87e47a979943ad
2
```

3.1 Data Sourcing:

The dataset is in csv (comma separated values) format. MS Excel is used to load the data.

Citation Request:

This dataset is publicly available for research. The details are described in [Moro et al., 2014].
Please include this citation if you plan to use this database:

[Moro et al., 2014] S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, In press, http://dx.doi.org/10.1016/j.dss.2014.03.001

Available at: [pdf] http://dx.doi.org/10.1016/j.dss.2014.03.001
[bib] http://www3.dsi.uminho.pt/pcortez/bib/2014-dss.txt

1. Title: Bank Marketing (with social/economic context)
2. Sources:
   Created by: Sérgio Moro (ISCTE-IUL), Paulo Cortez (Univ. Minho) and Paulo

Rita (ISCTE-IUL) @ 2014

3. Past Usage:

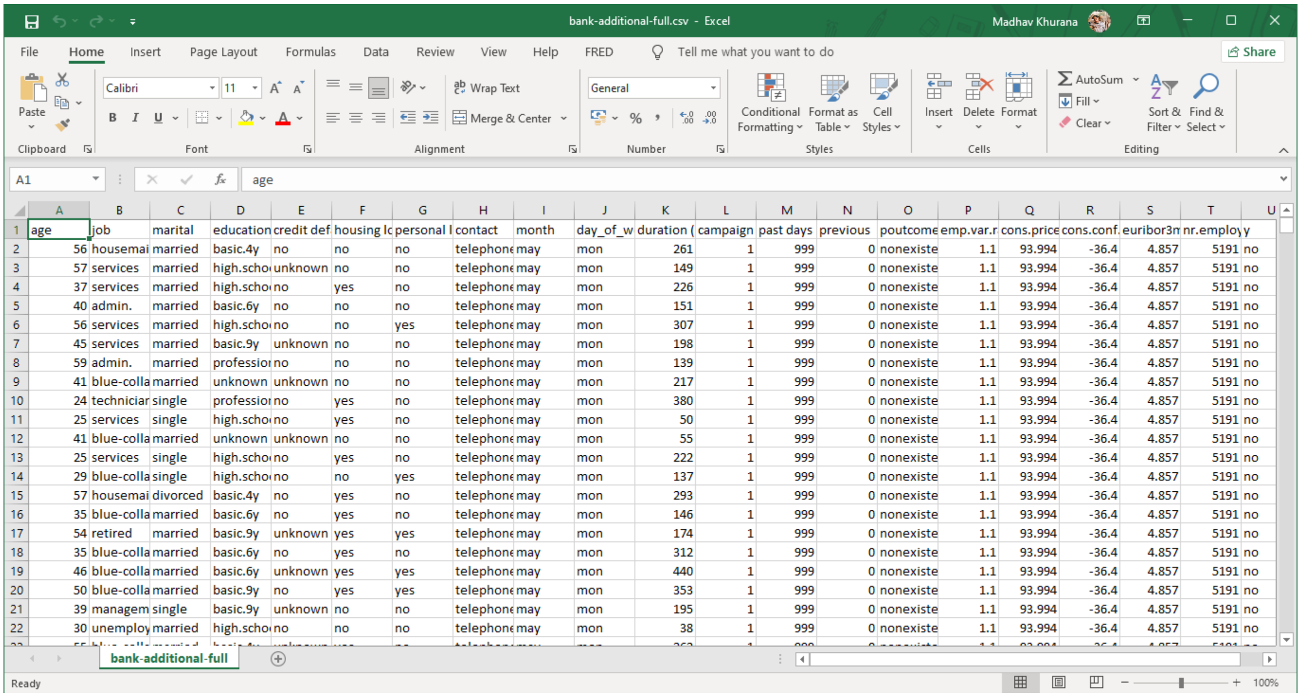
The full dataset (bank-additional-full.csv) was described and analyzed in:
S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems (2014), doi:10.1016/j.dss.2014.03.001.

3.2 Data Overview:

- This dataset is based on "Bank Marketing" UCI dataset (please check the description at: http://archive.ics.uci.edu/ml/datasets/Bank+Marketing).
- The data is enriched by the addition of five new social and economic features/attributes (national wide indicators from a ~10M population country), published by the Banco de Portugal and publicly available at: https://www.bportugal.pt/estatisticasweb.
- This dataset is almost identical to the one used in [Moro et al., 2014] (it does not include all attributes due to privacy concerns).
- Using the rminer package and R tool (http://cran.r-project.org/web/packages/rminer/), we found that the addition of the five new social and economic attributes (made available here) lead to substantial improvement in the prediction of a success, even when the duration of the call is not included. Note: the file can be read in R using: d=read.table("bank-additional-full.csv",header=TRUE,sep=";")

The zip file includes two datasets:

1) bank-additional-full.csv with all examples, ordered by date (from May 2008 to November 2010).
2) bank-additional.csv with 10% of the examples (4119), randomly selected from bank-additional-full.csv.
3) The smallest dataset is provided to test more computationally demanding machine learning algorithms (e.g., SVM).
4) The binary classification goal is to predict if the client will subscribe a bank term deposit (variable y).
5) Number of Instances: 41188 for bank-additional-full.csv
6) Number of Attributes: 20 + output attribute.



A1 | age

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	age	job	marital	education	credit def	housing lc	personal l	contact	month	day_of_w	duration (	campaign	past days	previous	poutcome	emp.var.r	cons.price	cons.conf.	euribor3m	nr.employ	y
2	56	housemai	married	basic.4y	no	no	no	telephone	may	mon	261	1	999	0	nonexiste	1.1	93.994	-36.4	4.857	5191	no
3	57	services	married	high.scho	unknown	no	no	telephone	may	mon	149	1	999	0	nonexiste	1.1	93.994	-36.4	4.857	5191	no
4	37	services	married	high.scho	no	yes	no	telephone	may	mon	226	1	999	0	nonexiste	1.1	93.994	-36.4	4.857	5191	no
5	40	admin.	married	basic.6y	no	no	no	telephone	may	mon	151	1	999	0	nonexiste	1.1	93.994	-36.4	4.857	5191	no
6	56	services	married	high.scho	no	no	yes	telephone	may	mon	307	1	999	0	nonexiste	1.1	93.994	-36.4	4.857	5191	no
7	45	services	married	basic.9y	unknown	no	no	telephone	may	mon	198	1	999	0	nonexiste	1.1	93.994	-36.4	4.857	5191	no
8	59	admin.	married	professio	no	no	no	telephone	may	mon	139	1	999	0	nonexiste	1.1	93.994	-36.4	4.857	5191	no
9	41	blue-colla	married	unknown	unknown	no	no	telephone	may	mon	217	1	999	0	nonexiste	1.1	93.994	-36.4	4.857	5191	no
10	24	techniciar	single	professio	no	yes	no	telephone	may	mon	380	1	999	0	nonexiste	1.1	93.994	-36.4	4.857	5191	no
11	25	services	single	high.scho	no	yes	no	telephone	may	mon	50	1	999	0	nonexiste	1.1	93.994	-36.4	4.857	5191	no
12	41	blue-colla	married	unknown	unknown	no	no	telephone	may	mon	55	1	999	0	nonexiste	1.1	93.994	-36.4	4.857	5191	no
13	25	services	single	high.scho	no	yes	no	telephone	may	mon	222	1	999	0	nonexiste	1.1	93.994	-36.4	4.857	5191	no
14	29	blue-colla	single	high.scho	no	no	yes	telephone	may	mon	137	1	999	0	nonexiste	1.1	93.994	-36.4	4.857	5191	no
15	57	housemai	divorced	basic.4y	no	yes	no	telephone	may	mon	293	1	999	0	nonexiste	1.1	93.994	-36.4	4.857	5191	no
16	35	blue-colla	married	basic.6y	no	yes	no	telephone	may	mon	146	1	999	0	nonexiste	1.1	93.994	-36.4	4.857	5191	no
17	54	retired	married	basic.9y	unknown	yes	yes	telephone	may	mon	174	1	999	0	nonexiste	1.1	93.994	-36.4	4.857	5191	no
18	35	blue-colla	married	basic.6y	no	yes	no	telephone	may	mon	312	1	999	0	nonexiste	1.1	93.994	-36.4	4.857	5191	no
19	46	blue-colla	married	basic.6y	unknown	yes	yes	telephone	may	mon	440	1	999	0	nonexiste	1.1	93.994	-36.4	4.857	5191	no
20	50	blue-colla	married	basic.9y	no	yes	yes	telephone	may	mon	353	1	999	0	nonexiste	1.1	93.994	-36.4	4.857	5191	no
21	39	managem	single	basic.9y	unknown	no	no	telephone	may	mon	195	1	999	0	nonexiste	1.1	93.994	-36.4	4.857	5191	no
22	30	unemploy	married	high.scho	no	no	no	telephone	may	mon	38	1	999	0	nonexiste	1.1	93.994	-36.4	4.857	5191	no

3.3 Data Description

Input variables:

# Bank client data:

1 - age (numeric)
2 - job : type of job (categorical: "admin.","blue-collar","entrepreneur","housemaid","management","retired","self-employed","services","student","technician","unemployed","unknown")
3 - marital : marital status (categorical: "divorced","married","single","unknown"; note: "divorced" means divorced or widowed)
4 - education (categorical: "basic.4y","basic.6y","basic.9y","high.school","illiterate","professional.course","university.degree","unknown")
5 - default: has credit in default? (categorical: "no","yes","unknown")
6 - housing: has housing loan? (categorical: "no","yes","unknown")
7 - loan: has personal loan? (categorical: "no","yes","unknown")

# related with the last contact of the current campaign:

8 - contact: contact communication type (categorical: "cellular","telephone")
9 - month: last contact month of year (categorical: "jan", "feb", "mar", ..., "nov", "dec")
10 - day_of_week: last contact day of the week (categorical: "mon","tue","wed","thu","fri")
11 - duration: last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y="no"). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.

# other attributes:

12 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)
13 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
14 - previous: number of contacts performed before this campaign and for this client (numeric)
15 - poutcome: outcome of the previous marketing campaign (categorical: "failure","nonexistent","success")

# social and economic context attributes

16 - emp.var.rate: employment variation rate - quarterly indicator (numeric)
17 - cons.price.idx: consumer price index - monthly indicator (numeric)
18 - cons.conf.idx: consumer confidence index - monthly indicator (numeric)
19 - euribor3m: euribor 3 month rate - daily indicator (numeric)
20 - nr.employed: number of employees - quarterly indicator (numeric)

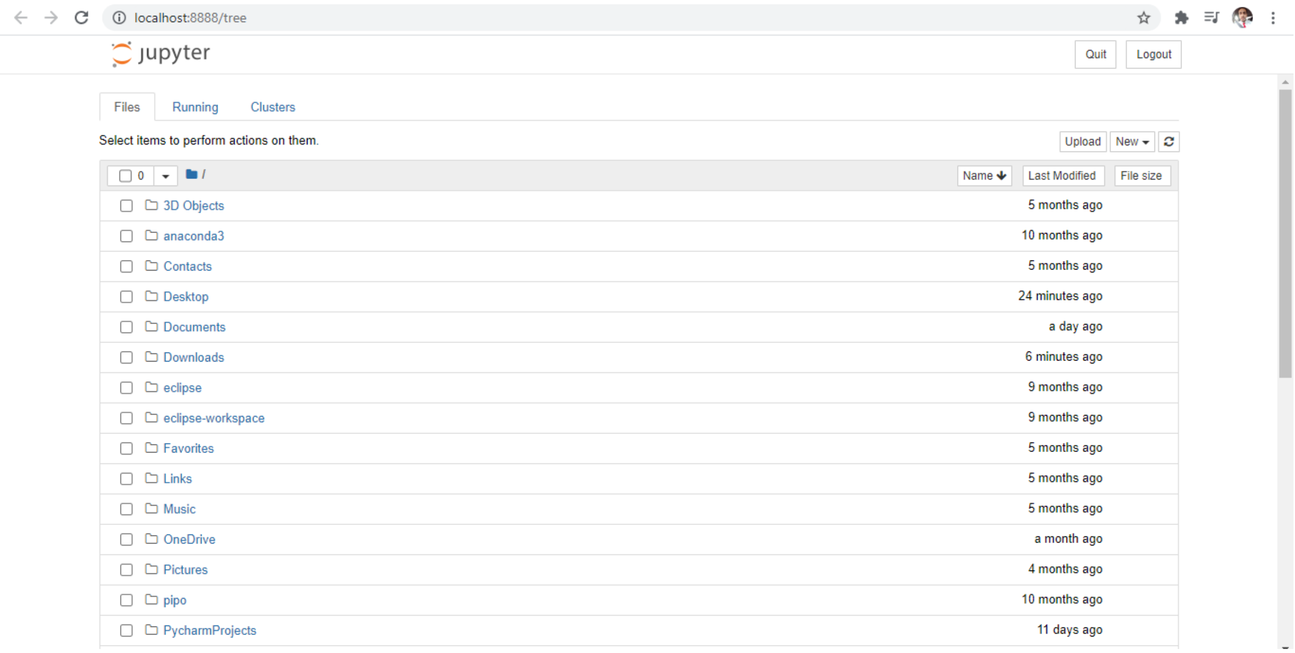
Output variable (desired target):

21 - y - has the client subscribed a term deposit? (binary: "yes","no")

3.4 Data loading in Python pandas Dataframe

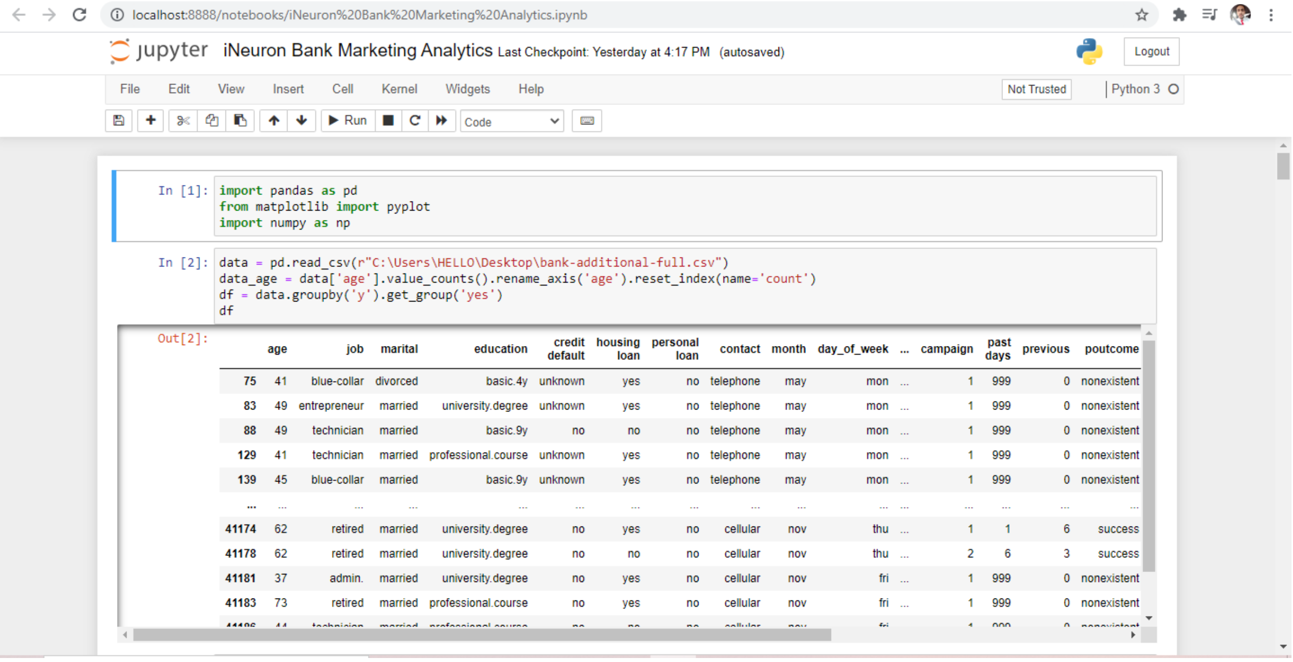
The DataFrame as a “two-dimensional, size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns)”. In plain terms, think of a DataFrame as a table of data, i.e. a single set of formatted two-dimensional data, with the following characteristics:

- There can be multiple rows and columns in the data.
- Each row represents a sample of data,
- Each column contains a different variable that describes the samples (rows).
- The data in every column is usually the same type of data – e.g. numbers, strings, dates.
- Usually, unlike an excel data set, DataFrames avoid having missing values, and there are no gaps and empty values between rows or columns.



Select items to perform actions on them.

Name	Last Modified	File size
□ 3D Objects	5 months ago	
□ anaconda3	10 months ago	
□ Contacts	5 months ago	
□ Desktop	24 minutes ago	
□ Documents	a day ago	
□ Downloads	6 minutes ago	
□ eclipse	9 months ago	
□ eclipse-workspace	9 months ago	
□ Favorites	5 months ago	
□ Links	5 months ago	
□ Music	5 months ago	
□ OneDrive	a month ago	
□ Pictures	4 months ago	
□ pipo	10 months ago	
□ PycharmProjects	11 days ago	


localhost:8888/notebooks/iNeuron%20Bank%20Marketing%20Analytics.ipynb
jupyter
iNeuron Bank Marketing Analytics Last Checkpoint: Yesterday at 4:17 PM (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3
Run
Code

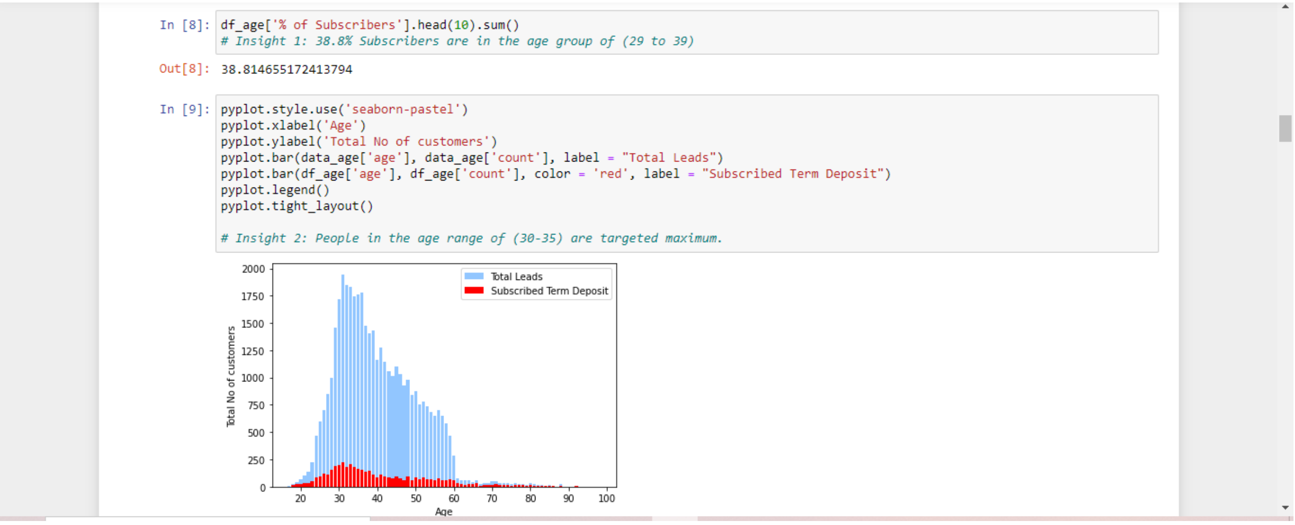
```
In [1]: import pandas as pd
        from matplotlib import pyplot
        import numpy as np
```

```
In [2]: data = pd.read_csv(r"C:\Users\HELLO\Desktop\bank-additional-full.csv")
        data_age = data['age'].value_counts().rename_axis('age').reset_index(name='count')
        df = data.groupby('y').get_group('yes')
        df
```

Out[2]:

	age	job	marital	education	credit default	housing loan	personal loan	contact	month	day_of_week	...	campaign	past days	previous	poutcome
75	41	blue-collar	divorced	basic.4y	unknown	yes	no	telephone	may	mon	...	1	999	0	nonexistent
83	49	entrepreneur	married	university.degree	unknown	yes	no	telephone	may	mon	...	1	999	0	nonexistent
88	49	technician	married	basic.9y	no	no	no	telephone	may	mon	...	1	999	0	nonexistent
129	41	technician	married	professional.course	unknown	yes	no	telephone	may	mon	...	1	999	0	nonexistent
139	45	blue-collar	married	basic.9y	unknown	yes	no	telephone	may	mon	...	1	999	0	nonexistent
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
41174	62	retired	married	university.degree	no	yes	no	cellular	nov	thu	...	1	1	6	success
41178	62	retired	married	university.degree	no	no	no	cellular	nov	thu	...	2	6	3	success
41181	37	admin.	married	university.degree	no	yes	no	cellular	nov	fri	...	1	999	0	nonexistent
41183	73	retired	married	professional.course	no	yes	no	cellular	nov	fri	...	1	999	0	nonexistent
41186	44	technician	married	professional.course	no	no	no	cellular	nov	fri	...	1	999	0	nonexistent

3.5 Data to Insights through Visualization

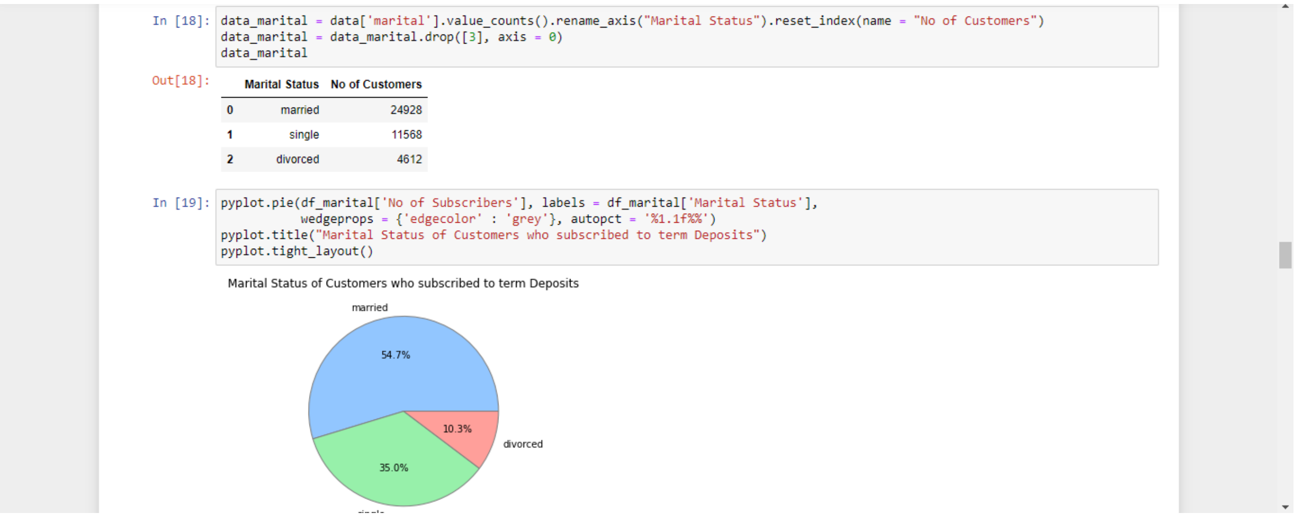


```
In [8]: df_age['% of Subscribers'].head(10).sum()
        # Insight 1: 38.8% Subscribers are in the age group of (29 to 39)

Out[8]: 38.814655172413794
```

```
In [9]: pyplot.style.use('seaborn-pastel')
        pyplot.xlabel('Age')
        pyplot.ylabel('Total No of customers')
        pyplot.bar(data_age['age'], data_age['count'], label = "Total Leads")
        pyplot.bar(df_age['age'], df_age['count'], color = 'red', label = "Subscribed Term Deposit")
        pyplot.legend()
        pyplot.tight_layout()

        # Insight 2: People in the age range of (30-35) are targeted maximum.
```



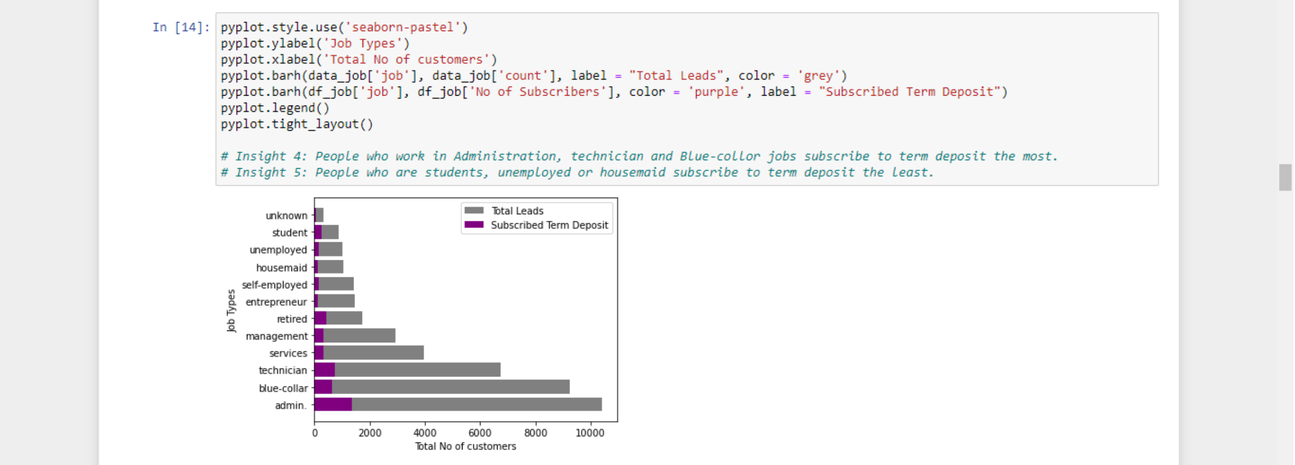
```
In [18]: data_marital = data['marital'].value_counts().rename_axis("Marital Status").reset_index(name = "No of Customers")
         data_marital = data_marital.drop([3], axis = 0)
         data_marital
```

Out[18]:

	Marital Status	No of Customers
0	married	24928
1	single	11568
2	divorced	4612

```
In [19]: pyplot.pie(df_marital['No of Subscribers'], labels = df_marital['Marital Status'],
                    wedgeprops = {'edgecolor' : 'grey'}, autopct = '%1.1f%%')
         pyplot.title("Marital Status of Customers who subscribed to term Deposits")
         pyplot.tight_layout()
```

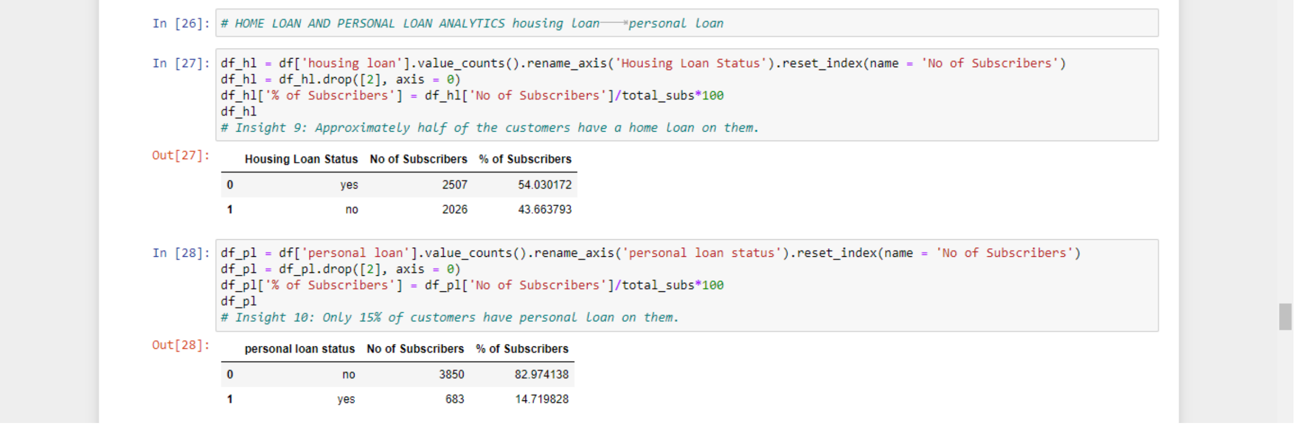
Marital Status of Customers who subscribed to term Deposits



```
In [14]: pyplot.style.use('seaborn-pastel')
         pyplot.ylabel('Job Types')
         pyplot.xlabel('Total No of customers')
         pyplot.barh(data_job['job'], data_job['count'], label = "Total Leads", color = 'grey')
         pyplot.barh(df_job['job'], df_job['No of Subscribers'], color = 'purple', label = "Subscribed Term Deposit")
         pyplot.legend()
         pyplot.tight_layout()

         # Insight 4: People who work in Administration, technician and Blue-collor jobs subscribe to term deposit the most.
         # Insight 5: People who are students, unemployed or housemaid subscribe to term deposit the least.
```

3.6 Data to Insights through of Data frames



```
In [26]: # HOME LOAN AND PERSONAL LOAN ANALYTICS housing loan——→personal loan

In [27]: df_hl = df['housing loan'].value_counts().rename_axis('Housing Loan Status').reset_index(name = 'No of Subscribers')
         df_hl = df_hl.drop([2], axis = 0)
         df_hl['% of Subscribers'] = df_hl['No of Subscribers']/total_subs*100
         df_hl
         # Insight 9: Approximately half of the customers have a home loan on them.
```

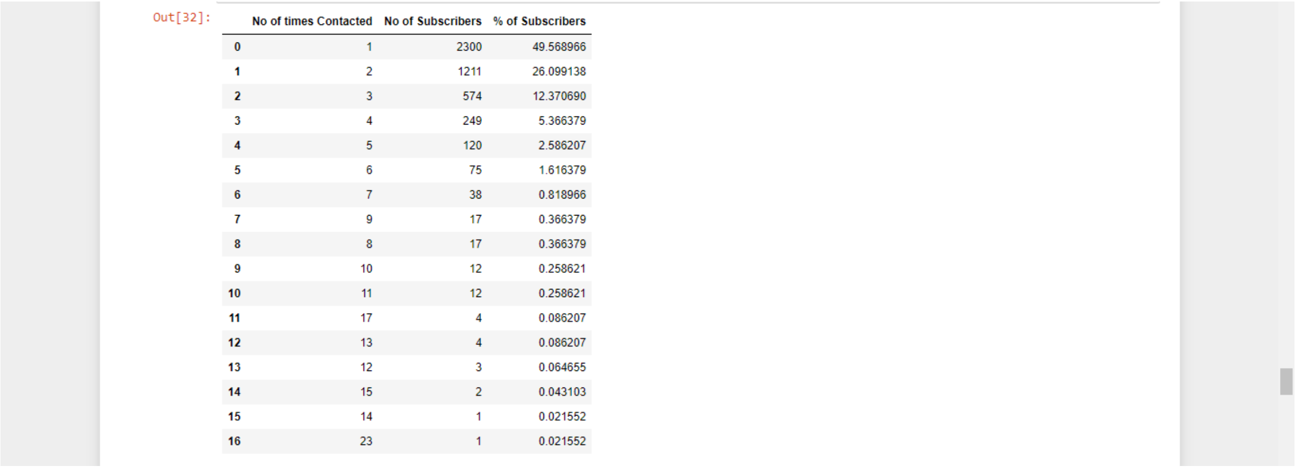
Out[27]:

	Housing Loan Status	No of Subscribers	% of Subscribers
0	yes	2507	54.030172
1	no	2026	43.663793

```
In [28]: df_pl = df['personal loan'].value_counts().rename_axis('personal loan status').reset_index(name = 'No of Subscribers')
         df_pl = df_pl.drop([2], axis = 0)
         df_pl['% of Subscribers'] = df_pl['No of Subscribers']/total_subs*100
         df_pl
         # Insight 10: Only 15% of customers have personal loan on them.
```

Out[28]:

	personal loan status	No of Subscribers	% of Subscribers
0	no	3850	82.974138
1	yes	683	14.719828



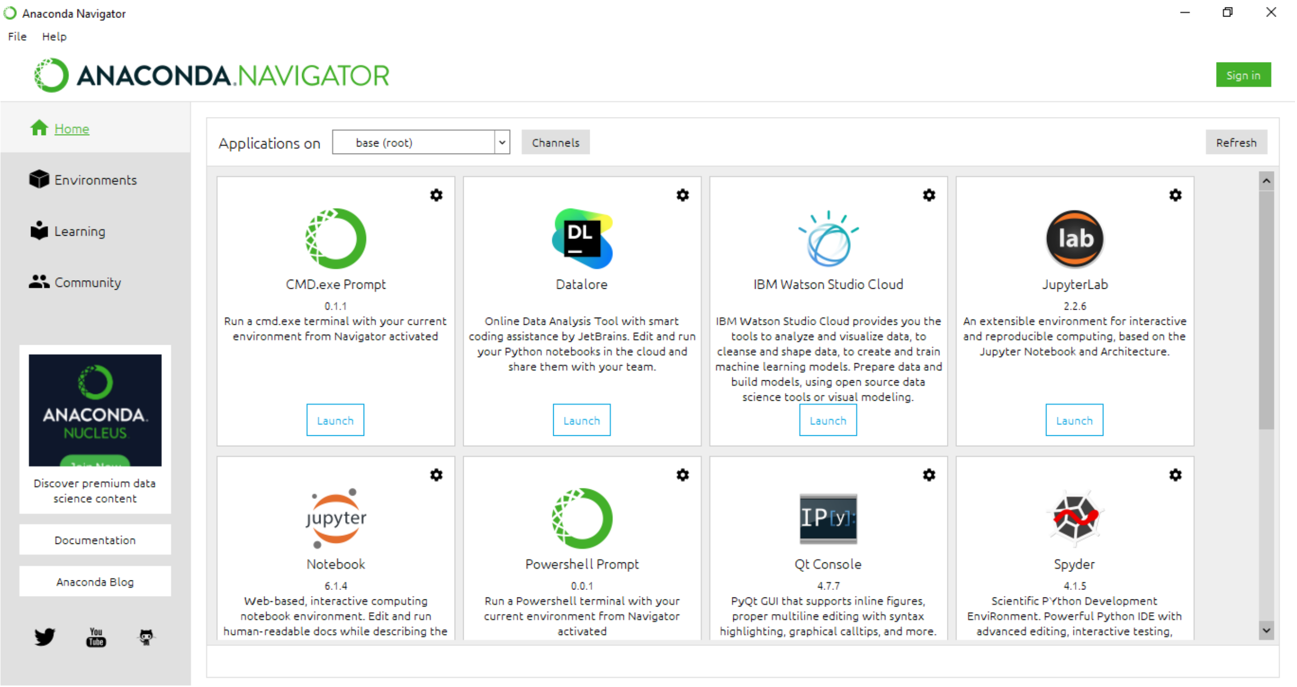
Out[32]:

	No of times Contacted	No of Subscribers	% of Subscribers
0	1	2300	49.568966
1	2	1211	26.099138
2	3	574	12.370690
3	4	249	5.366379
4	5	120	2.586207
5	6	75	1.616379
6	7	38	0.818966
7	9	17	0.366379
8	8	17	0.366379
9	10	12	0.258621
10	11	12	0.258621
11	17	4	0.086207
12	13	4	0.086207
13	12	3	0.064655
14	15	2	0.043103
15	14	1	0.021552
16	23	1	0.021552

3.7 Dataframes Generated:

- **df** – It contains all information of people who subscribed term deposit.
- **Data** - It contains all information of people who were contacted during the marketing campaign.
- **df_age** - It contains age information of people who subscribed term deposit.
- **data_age** - It contains age information of people who were contacted during the marketing campaign.
- **df_job** - It contains job information of people who subscribed term deposit.
- **data_job** - - It contains job information of people who were contacted during the marketing campaign.
- **df_marital** - It contains information related to their marital status of people who subscribed term deposit.
- **df_ed** - It contains all information related to education they have people who subscribed term deposit.
- **df_cd** - It contains all information of people who subscribed term deposit whether they defaulted on credit or not.
- **df_hl** - It contains all information of people who subscribed term deposit whether they have home loan or not.
- **df_pl** - It contains all information of people who subscribed term deposit whether they have personal loan or not.
- **df_dur** - It contains all information of call duration of people who subscribed term deposit.
- **df_cam** - It contains all information related to no of contacts performed to people who subscribed term deposit.

4. Deployment



CMD.exe Prompt
0.1.1
Run a cmd.exe terminal with your current environment from Navigator activated

Datalore
Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.

IBM Watson Studio Cloud
IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

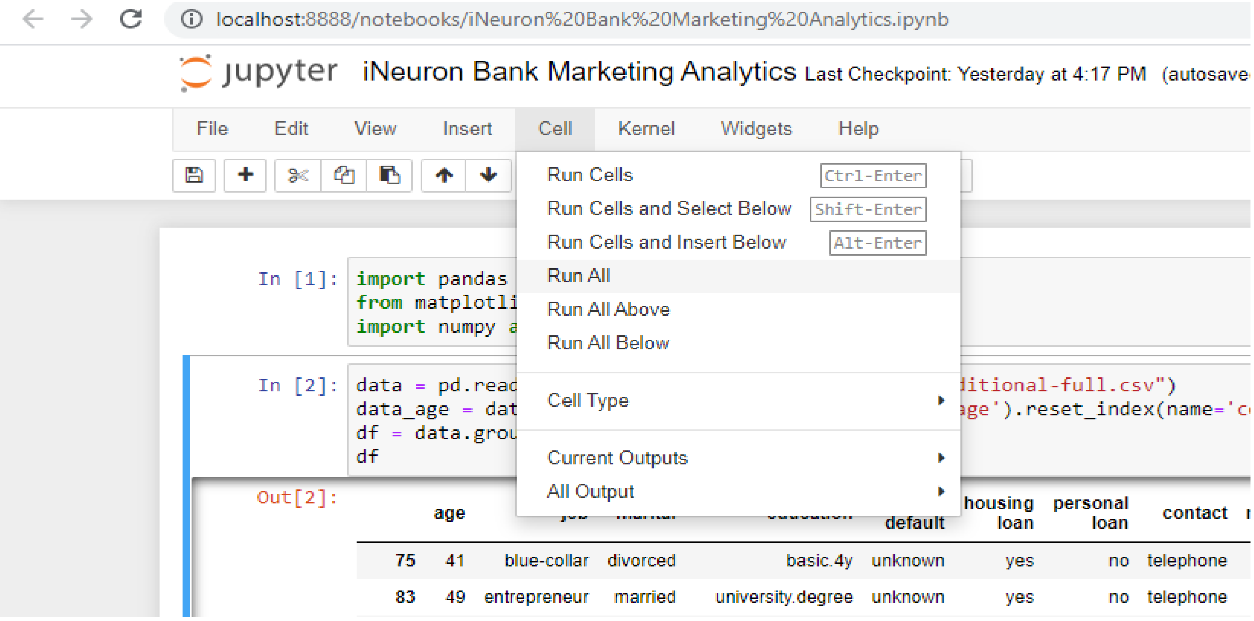
JupyterLab
2.2.6
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Notebook
6.1.4
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the

Powershell Prompt
0.0.1
Run a Powershell terminal with your current environment from Navigator activated

Qt Console
4.7.7
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Spyder
4.1.5
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing,


localhost:8888/notebooks/iNeuron%20Bank%20Marketing%20Analytics.ipynb
jupyter
iNeuron Bank Marketing Analytics Last Checkpoint: Yesterday at 4:17 PM (autosave
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Run Cells Ctrl-Enter
Run Cells and Select Below Shift-Enter
Run Cells and Insert Below Alt-Enter
Run All
Run All Above
Run All Below
Cell Type
Current Outputs
All Output

```
In [1]: import pandas
        from matplotli
        import numpy a

In [2]: data = pd.read                                  ditional-full.csv")
        data_age = dat                                  age').reset_index(name='c
        df = data.grou
        df
```

Out[2]:

	age	job	marital	education	default	housing loan	personal loan	contact
75	41	blue-collar	divorced	basic.4y	unknown	yes	no	telephone
83	49	entrepreneur	married	university.degree	unknown	yes	no	telephone

To execute the cells in the jupyter notebook, you have to

1. Open file named “iNeuron Bank marketing Analytics.ipynb” in the jupyter notebook using anaconda navigator
2. To run cells, you have 2 options either to run every cell individually by using SHIFT + RETURN
3. Or we can run all the cells by clicking on Cell button on the home ribbon, then click “Run All”
4. Scroll down gradually to see visualizations and Insights.