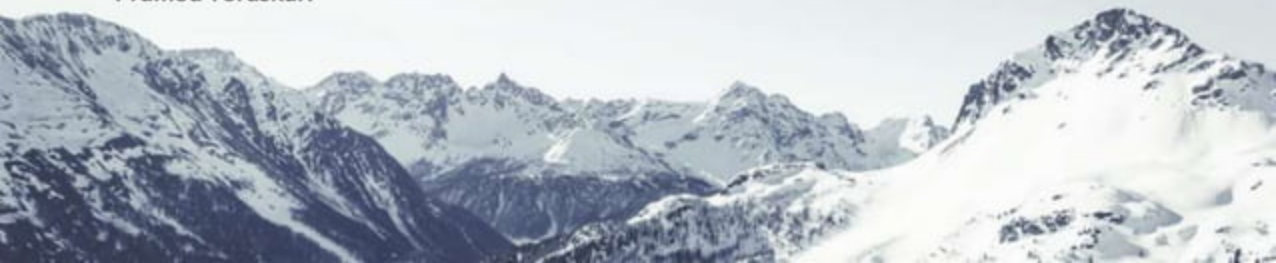
Basic of Python for Data Analysis

Pramod Toraskar.



Why learn Python for data analysis?

Here are some reasons which go in favour of learning Python:

- Open Source – free to install
- Awesome online community
- Very easy to learn
- Can become a common language for data science and production of web based analytics products.

Choosing a development environment

1. Terminal / Shell based

2. IDLE (default environment)

3. iPython notebook – similar to markdown in R

   iPython environment - jupyter

   http://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/install.html

Recall Python libraries and Data Structures

Lists, Strings, Tuples, Dictionary..

Following are a list of libraries, you will need for any scientific computations and data analysis:

- **NumPy** (Numerical Python). The most powerful feature of NumPy is n-dimensional array. This library also contains basic linear algebra functions, Fourier transforms, advanced random number capabilities and tools for integration with other low level languages like Fortran, C and C++

- **SciPy** (Scientific Python). SciPy is built on NumPy. It is one of the most useful library for variety of high level science and engineering modules like discrete Fourier transform, Linear Algebra, Optimization and Sparse matrices.

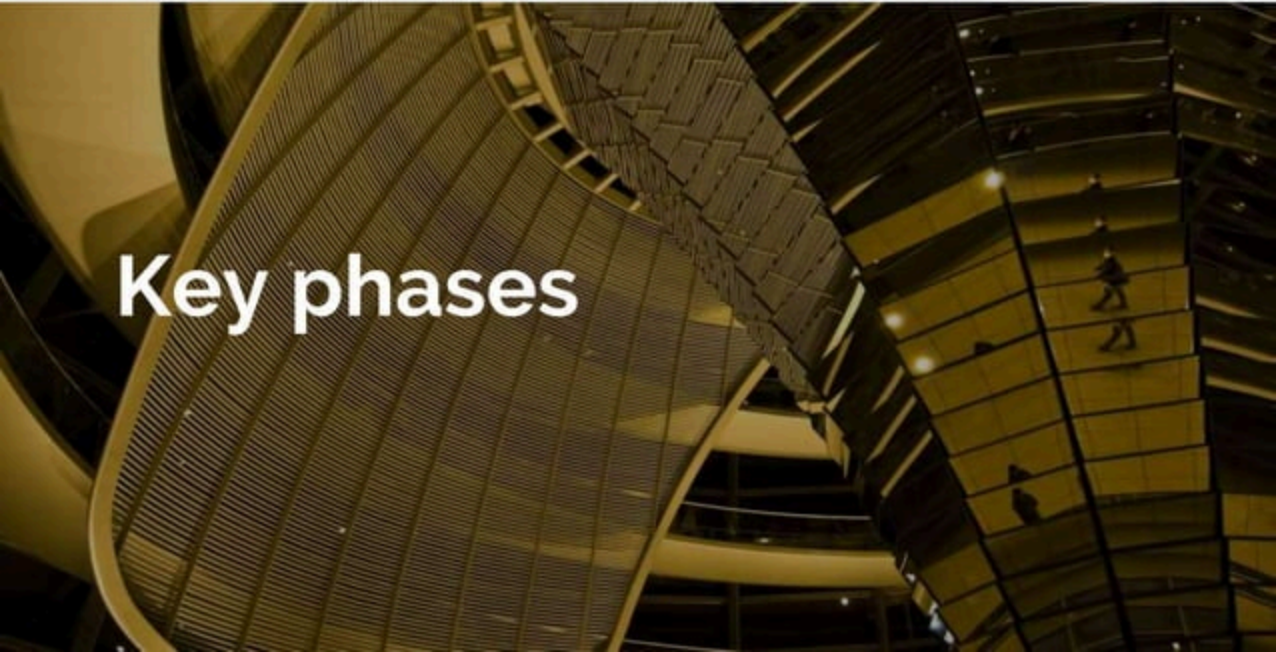
- **Matplotlib** for plotting vast variety of graphs, starting from histograms to line plots to heat plots.. You can use Pylab feature in ipython notebook (ipython notebook –pylab = inline) to use these plotting features inline. If you ignore the inline option, then pylab converts ipython environment to an environment, very similar to Matlab. You can also use Latex commands to add math to your plot.

- **Pandas** for structured data operations and manipulations. It is extensively used for data munging and preparation. Pandas were added relatively recently to Python and have been instrumental in boosting Python's usage in data scientist community.

- **Scikit Learn** for machine learning. Built on NumPy, SciPy and matplotlib, this library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.

- **Statsmodels** (statistical modeling), **Seaborn** (statistical data visualization), **Bokeh** (creating interactive plots, dashboards and data applications on modern web-browsers. It empowers the user to generate elegant and concise graphics in the style of D3.js.)

Key phases



The 3 key phases

01

Data Exploration:
Finding out more about the data we have

- numpy
- matplotlib
- Pandas

```
import pandas as pd
import numpy as np
import matplotlib as plt
```

```
df = pd.read_csv("/home/ptoraska/Downloads/Loan_Prediction/train.csv")
#Reading the dataset in a dataframe using Pandas
```

Data Exploration

Once you have read the dataset, you can have a look at few top rows by using the function head()

df.head(10)

```
In [3]: df.head(10)  #Printing first 10 rows of dataset
```

Out[3]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Cr
0	LP001002	Male	No	0	Graduate	No	5849	0	NaN	360	1
1	LP001003	Male	Yes	1	Graduate	No	4583	1508	128	360	1
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0	66	360	1
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358	120	360	1
4	LP001008	Male	No	0	Graduate	No	6000	0	141	360	1
5	LP001011	Male	Yes	2	Graduate	Yes	5417	4196	267	360	1
6	LP001013	Male	Yes	0	Not Graduate	No	2333	1516	95	360	1
7	LP001014	Male	Yes	3+	Graduate	No	3036	2504	158	360	0
8	LP001018	Male	Yes	2	Graduate	No	4006	1526	168	360	1
9	LP001020	Male	Yes	1	Graduate	No	12841	10968	349	360	1

The 3 key phases

02

Data Munging:

Cleaning the data and playing with it to make it better suit statistical modeling.

1. There are missing values in some variables. We should estimate those values wisely depending on the amount of missing values and the expected importance of variables.

1. While looking at the distributions, we saw that Applicant Income and Loan Amount seemed to contain extreme values at either end. Though they might make intuitive sense, but should be treated appropriately.

Check missing values in the dataset

Let us look at missing values in all the variables because most of the models don't work with missing data and even if they do, imputing them helps more often than not. So, let us check the number of nulls / NaNs in the dataset

df.apply(lambda x: sum(x.isnull()),axis=0)

```
In [14]: df.apply(lambda x: sum(x.isnull()),axis=0)

Out[14]: Loan_ID              0
         Gender              13
         Married              3
         Dependents          15
         Education            0
         Self_Employed       32
         ApplicantIncome      0
         CoapplicantIncome    0
         LoanAmount          22
         Loan_Amount_Term    14
         Credit_History      50
         Property_Area        0
         Loan_Status          0
         dtype: int64
```

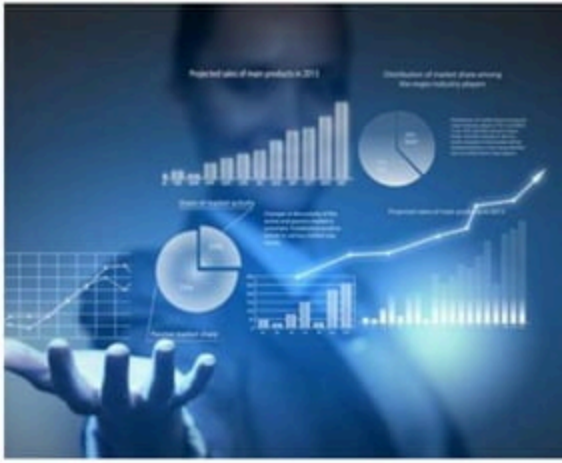
The 3 key phases

03

Predictive Modeling:

Running the actual algorithms and having fun

After, we have made the data useful for modeling, The Skicit-Learn (sklearn) is the most commonly used library in Python for this purpose



Building a Predictive Model in Python

sklearn requires all inputs to be numeric, we should convert all our categorical variables into numeric by encoding the categories.

This can be done using the following code:

```
from sklearn.preprocessing import LabelEncoder

var_mod =
['Gender','Married','Dependents','Education','Self_Employed','Property_Are
a','Loan_Status']

le = LabelEncoder()


for i in var_mod:
  df[i] = le.fit_transform(df[i])


df.dtypes
```

Model's

Logistic Regression

Is a classification algorithm

Decision Tree

is a type of supervised learning algorithm (having a pre-defined target variable) that is mostly used in classification problems.

Random Forest

Is a versatile machine learning method capable of performing both regression and classification tasks.

Thank you.