Data Analysis with python — Importing Datasets.

Lecture 1 : The problem

Why Data Analysis ?

→ Data is everywhere

→ Data analysis / data science helps us answer questions from data.

→ Data analysis plays an important role in :

- Discoving useful information
- Answering questions
- Predicting future or the unknown

Lec 2 : Understanding the Data

dataset = open dataset by Jeffrey C. Schlemmer (1985)

CSV → comma seperated values

Easy to import dataset

1st row → header } which contains a column name for each of the 26 columns

Attributes → i) symboling = insurance risk level of a car

ii) Normalized losses :- It is the relative average loss payment per insured vehicle year

continous from 65 to 256

Lect 3 : Python Packages for Data Science

A python library is a collection of functions and methods that allow you to perform lots of actions without writing any code.

The libraries usually contain built in modules providing different functionalities which you can use directly. And there are extensive libraries offering a broad range of facilities.

Python Data Analysis Libraries

1. Scientific computing Libraries

A) Pandas (Data structures tools)

for effective data manipulation and analysis

The primary instrunt of pandas are is the 2-dimensional table consisting of column and row labels, which are called a data frame. It is designed to provided easy indexing functionality

2. Visualization Libraries

B) NumPy (Arrays & matrices)

The Numpy library uses arrays for its i/p and o/ps. It can be extended to objects for matrices and with minor coding changes, developers perform fast array processing.

3. Algorithmic libraries

C) SciPy (Integrals, solving differential eqns, optimization)

→ some advanced math problems as listed on this slide, as well as data visualization

2. Visualization libraries.

↓ It is the best way to communicate with others by showing them meaningful results of analysis. These libraries enable you to create graphs, charts and maps.

① Matplotlib :- (plots & graphs, most popular)
It is great for making graphs and plots

② Seaborn :- (heat maps, time series, violin plots)

3. Algorithmic libraries :-

It tackles the machine learning tasks from basic to complex. Here we introduce 2 packages.

① Scikit-learn :- (machine learning : regression, classification, ...)
↓
contains tools statistical modeling.
This library is built on Numpy, Scipy and Matplotlib

② Statmodels :- (Explore data, estimate statistical models, and perform statistical tests)
↓
It allows users to explore data, estimate statistical models, and perform statistical models.

Importing and Exporting data in Python

Data acquisition is a process of loading and reading data into notebook from various sources.

(1) Importing Data :- Process of loading and reading data into Python from various resources.

* Two important properties :

i) Format :- It is the way data is encoded. Different can know encoding schemes by looking at the ending of the file name.

- Various formats :- .csv, .json, .xlsx, .hdf

ii) File path of dataset :- The path tells us where the data is stored.

- computer : /desktop/mydata.csv
- internet : https://archive.ics.uci.edu/autos/imports-85.data

→ In pandas, the read-csv method can read in files with columns separated by commas into a pandas data frame.

Reading data in pandas can be done in 3 lines

i) Import pandas

ii) Then define a variable with a file path

iii) Then use the read_csv method to import the data.

>> import pandas as pd

>> url = " "

>> df = pd.read_csv(url, header = None)

→ read_csv assumes the data contains a header. Our data on used cars has no column headers

* Printing the dataframe in Python

→ (1) df prints the entire dataframe (not recommended for large datasets)

→ (2) df.head(n) to show the first n rows of dataframe

→ (3) df.tail(n) shows the bottom n rows of dataframe

For adding headers

→ Replace default header (by df.columns = headers)

df.head(5) ⟶ ~~first 4~~ first 5 rows

* Exporting a Pandas dataframe to CSV

→ Preserve progress anytime by saving modified dataset using

path = " "

df.to_csv(path)

Exporting to different formats in Python.

Data Format	Read	Save
csv	pd.read_csv()	df.to_csv()
json	pd.read_json()	df.to_json()
Excel	pd.read_excel()	df.to_excel()
sql	pd.read_sql()	df.to_sql()

Lect : Getting started analyzing data in Python

* Basic insights from the data

• Understand your data before you begin any analysis

→ Pandas has several built-in methods that can be used to understand

• Should check : • Data Types

• Data distribution

• Locate potential issues with the data

* Basic Insights of Dataset - Data Types

Pandas Type	Native Python Type	Description
object	string	numbers and strings
int64	int	Numeric characters
float64	float	Numeric characters with d
datetime64, timedelta[ns]	N/A (but see the datetime module in Python's standard library)	time data

Why check data types ?

→ Potential info and type mismatch

eg.

^ The car price column which we should expect to contain continuous numeric numbers is assigned the data type of object. It would be more natural to, it t have the float type

→ Compatibility with python methods (which python functions can be applied to a specific column)

- In pandas, we use dataframe.dtypes to check data types.
  → the data type of each column is returned in a series

df.dtypes

- Returns a statistical summary
  → Tells whether mathematical issues exists or not

df.describe()

It returns the no. of terms in the column as count, average column value as mean, column standard deviation as std,

* dataFrame.describe(include = "all")

→ Provide full summary statistics

For object type column different set of stats is evaluated like unique, top, frequency

- unique: No. of distinct objects in the column
- top: Top is most frequently occuring object
- frequency: It is the no. of times the top object appears in the colums.

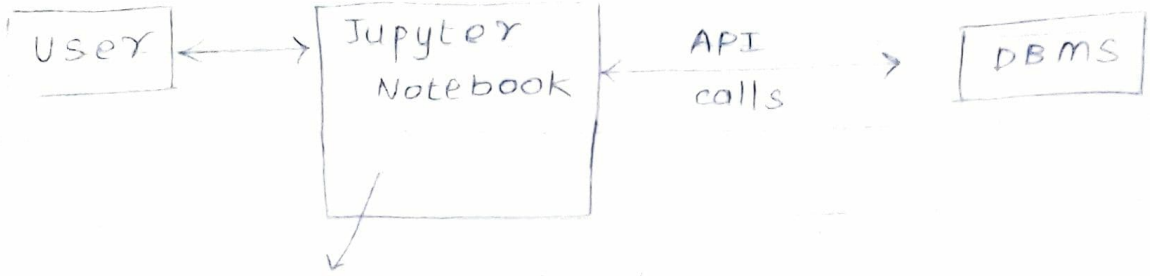
some value :- NaN (Not a number)

This is because that particular statistical metric cannot be calculated for that specific column data type.

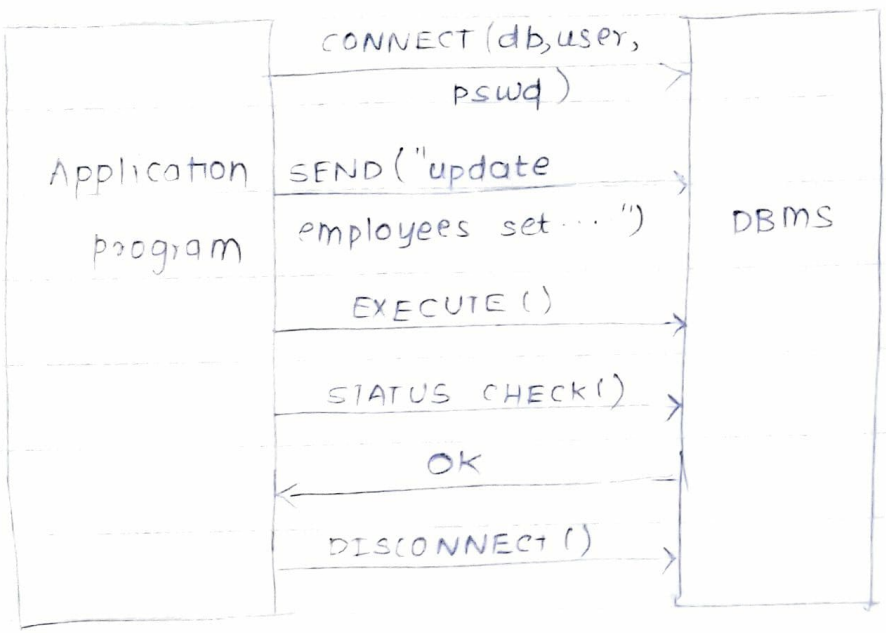
* dataframe.info() { provides a concise summary of your DataFrame

df.info() → This functions shows the top 30 rows and bottom 30 rows of the data frame

Lect : Accessing Databases with Python



What is a SQL API?

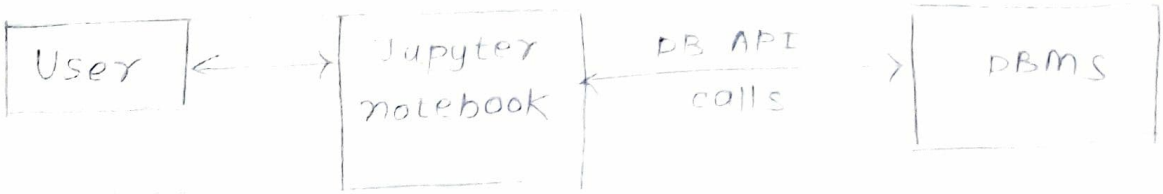


An application programming interface is a set of functions that you can call to get access to some type of servers

The SQL API consists of library function calls as an application programming interface, API for DBMS

* What is a DB-API?

DB-API is Python's standard API for accessing relational databases.



It is a standard that allow you to write a single program that works with multiple kinds of relational databases instead of writing a separate program for each one.

Concepts of the Python DB API

* Connection Objects
  → Database connections
  → Manage transactions
* Cursor Objects
  → Database Queries

→ The cursor works similar to a cursor in a text processing system where your result set and get your data into the applications

What are Connection methods ?

- cursor() ⟶ The cursor method returns a new cursor object using the connection
- commit() ⟶ The commit() method is used to commit any pending transaction to the database.
- rollback() ⟶ This method causes the database to roll back to the start of any pending transactions
- close() ⟶ Used to close database connection