Python Advance Assignment 3

1. What is the process for loading a dataset from an external source?

1. Import CSV files: pd.read_csv("file.scv",header = None, names = ['col1', 'col2', 'col3']) or np.genfromtxt('file.csv', delimiter=',')

2. Import File from URL: pd.read_csv("http://winterolympicsmedals.com/medals.csv")

3. Read Text File: pd.read_table("example2.txt") or pd.read_csv("example2.txt", sep ="\t") or numpy.loadtxt("example2.txt")

4. Read Excel File:
pd.read_excel("https://www.eia.gov/dnav/pet/hist_xls/RBRTEd.xls",sheetname="Data 1", skiprows=2)

5. Read delimited file: (file separated with white spaces)
pd.read_table("http://www.ssc.wisc.edu/~bhansen/econometrics/invest.dat", sep="\s+", header = None)

6. Read SAS File: pd.read_sas('cars.sas7bdat')

7. Read Stata File: pd.read_stata('cars.dta')

8. Import R Data File: import pyreadr; result = pyreadr.read_r('sampledata.RData'); df1 = result["df1"] # extract the pandas data frame for object df1

9. Read SQL Table: We can extract table from SQL database (SQL Server / Teradata). See the program below -

SQL Server: We can read data from tables stored in SQL Server by building a connection. We need to have server, User ID (UID), database details to establish connection.

```
import pandas, pyodbc

conn = pyodbc.connect("Driver={SQL
Server};Server=serverName;UID=UserName;PWD=Password;Database=RCO_DW;")

df = pandas.read_sql_query('select * from dbo.Table WHERE ID > 10', conn)
```

10. Import Data from SPSS File:

```
import pyreadstat

df, meta = pyreadstat.read_sav("file.sav", apply_value_formats=True)
```

11. Reading Pickle file

```
with open('test.pkl','wb') as f:
```

```
pickle.dump(pdDf, f)
```

2. How can we use pandas to read JSON files?

We can use the pandas library to read JSON files using the read_json() function, which reads a JSON file and creates a pandas DataFrame object. The function supports various parameters to customize the reading of the JSON file, such as specifying the orientation of the JSON data, selecting specific columns, and handling missing values.

```
import pandas as pd

df = pd.read_json('data.json')

print(df)
```

3. Describe the significance of DASK.

Dask is a parallel computing framework for Python that is designed to handle large datasets that cannot fit into memory on a single machine. It allows users to process data in a distributed and parallel manner across multiple machines or processors, making it easier to scale computations and handle larger datasets. Dask provides high-level APIs that are compatible with the popular scientific computing libraries in Python, including NumPy, Pandas, and Scikit-learn, making it easy to integrate with existing code. Additionally, Dask's flexible and modular architecture allows it to be used in a variety of computing environments, from laptops to clusters, and enables users to customize it to their specific needs.

4. Describe the functions of DASK.

Parallelization: Dask allows users to parallelize computations on large datasets that are too large to fit into memory on a single machine.

Distributed computing: Dask can distribute computations across multiple machines in a cluster, enabling even larger datasets to be processed.

Integration with other libraries: Dask can integrate with other libraries in the scientific Python ecosystem, such as NumPy, Pandas, and Scikit-Learn, to provide parallel and distributed computing capabilities.

Lazy evaluation: Dask uses lazy evaluation to delay computation until it is necessary, reducing memory usage and improving performance.

Task scheduling: Dask provides a task scheduler that manages the distribution of computations across a cluster of machines, optimizing resource utilization and minimizing communication overhead.

5. Describe Cassandra's features.

a) Distributed: Each node in the cluster has same role. There's no question of failure & the data set is distributed across the cluster but one issue is there that is the master isn't present in each node to support request for service.

b) Supports replication & Multi data center replication: Replication factor comes with best configurations in cassandra. Cassandra is designed to have a distributed system, for the deployment of large number of nodes for across multiple data centers and other key features too.

c) Scalability: It is designed to r/w throughput, Increase gradually as new machines are added without interrupting other applications.

d) Fault-tolerance: Data is automatically stored & replicated for fault-tolerance. If a node Fails, then it is replaced within no time.

e) MapReduce Support: It supports Hadoop integration with MapReduce support.Apache Hive & Apache Pig is also supported.

f) Query Language: Cassandra has introduced the CQL (Cassandra Query Language). Its a simple interface for accessing the Cassandra.