1Q.Define Data Science and explain the applications of data science in various sectors.
Ans. Data science :
Data science is an interdisciplinary field that involves using scientific methods, algorithms, processes, an d systems to extract insights and knowledge from structured and unstructured data. It combines various t echniques from mathematics, statistics, computer science, and domain-specific knowledge to analyze, int erpret, and derive actionable insights from data.
The primary goal of data science is to discover patterns, trends, correlations, and hidden information withi n large and complex datasets. This information is used to make informed decisions, predictions, optimizati ons, and to solve complex problems across different industries and sectors.

Applications of data science:
.Healthcare: Predictive analytics for disease diagnosis and patient care improvement.
.Finance: Risk assessment, fraud detection, algorithmic trading.
.E-commerce: Personalized recommendations, customer segmentation.
.Marketing: Targeted advertising, sentiment analysis, customer behavior prediction.
.Transportation: Route optimization, demand forecasting, self-driving technology.
.Entertainment: Content recommendation, user engagement analysis, trend forecasting.

2Q.Define Data Wrangling. Explain its importance and various steps involved in it.

Ans.Data Wrangling :
The process of cleaning, structuring, and enriching raw data into a usable format for analysis.

Importance of Data Wrangling:

.Data Quality Assurance: Data wrangling helps in identifying and correcting issues such as missing value s, outliers, duplicates, and inconsistencies, ensuring high data quality.
.Improved Analysis: Clean and structured data enables accurate analysis, allowing for better-informed de cision- making.
.Compatibility and Consistency: Standardizing data formats and structures ensures compatibility across di fferent datasets and consistency in analysis.
.Efficient Processing: Well-prepared data reduces errors, enhances efficiency in processing, and minimize s the chances of errors in subsequent analysis.
.Enhanced Interpretation: Organized and cleaned data simplifies the interpretation of results, making it ea sier to draw actionable insights.

Steps Involved in Data Wrangling:

.Data Collection: Gather raw data from various sources like databases, APIs, spreadsheets, sensors, or lo gs.
.Data Cleaning: Address missing values, handle outliers, correct inaccuracies, and remove duplicates to e nsure data quality.
.Data Transformation: Restructure, normalize, or convert data types to ensure consistency and compatibili ty across datasets.
.Data Enrichment: Merge datasets, extract relevant features, or create new variables to enhance the infor mation contained in the data.
.Data Integration: Combine and integrate multiple datasets if required for a comprehensive analysis.
.Validation and Quality Checks: Perform sanity checks, validate against defined rules, and ensure the dat a adheres to quality standards.
.Documentation: Maintain records of changes made during the wrangling process, creating a transparent t rail for future reference.
.Data Formatting: Standardize the format and structure of data for easier analysis and interpretation.

3Q.Write about the essential python libraries.

Ans: Essential Python Libraries:
.NumPy
.Pandas
.Matplotlib and Seaborn
.Scikit-learn
.TensorFlow and PyTorch

NumPy:
Purpose: NumPy is a powerful Python library used to perform numerical calculations on arrays .
Importance: Provides high-performance mathematical functions and operations, enabling efficient numeri cal calculations
Key Features: Array manipulation, mathematical functions, linear algebra operations, random number gen eration.

Pandas:
Purpose: Pandas is a popular Python library that offers easy-to-use data structures and tools for data ma nipulation and analysis, particularly designed to handle structured data like tables and time series, allowin g users to work with data efficiently through functionalities for cleaning, exploring, and transforming datas ets.
Importance: Facilitates data cleaning, exploration, transformation, and analysis in tabular form.
Key Features: Data manipulation, handling missing data, time series analysis, powerful data aggregation, and merging capabilities.

Matplotlib:
Purpose: It help us to create static 2D plotting , interactive, and publication-quality visualizations.
Importance: Essential for data visualization, enabling users to create various types of plots, histograms, s catterplots, etc.
Key Features: Line plots, bar charts, pie charts, scatter plots, customization options for visuals.

Seaborn:
Purpose: Statistical data visualization library built on top of Matplotlib, providing higher-level, attractive vis ualizations.
Importance: Simplifies complex visualizations, enhances Matplotlib’s default styles, and offers specialized plots for statistical analysis.
Key Features: Heatmaps, violin plots, joint plots, specialized statistical plots.

Scikit-learn:
Purpose: Machine learning library providing simple and efficient tools for data mining and data analysis.
Importance: Offers a wide range of algorithms for classification, regression, clustering, dimensionality red uction, and model evaluation.
Key Features: Regression, classification, clustering, model selection, pre-processing techniques, evaluati on metrics.

TensorFlow / PyTorch:
Purpose: Deep learning frameworks for building and training neural networks.
Importance: Enables the creation of complex neural network architectures, facilitating deep learning resea rch and applications.
Key Features: Neural network creation, training, deployment, and support for GPU acceleration.

SciPy:
Purpose: Scientific computing library that builds on NumPy, providing additional functionality for optimizati on, integration, interpolation, etc.
Importance: Offers tools for mathematical and scientific computation beyond what NumPy provides.
Key Features: Integration, optimization, interpolation, signal processing, linear algebra.

NLTK (Natural Language Toolkit):
Purpose: Natural language processing library for working with human language data.
Importance: Facilitates tasks like tokenization, stemming, tagging, parsing, and semantic reasoning.
Key Features: Text processing, linguistic data analysis, and modeling.

4Q.Write about the built in data structures used in python.
Ans:
 Lists
 Tuples
 Dictionaries
 Sets

.Lists:
  Lists are ordered and mutable collections in Python, capable of holding various data types within square brackets [].
example:
```
# Creating a list
my_list = [1, 2, 3, 'hello', 'world', True]
print(my_list)  # Output: [1, 2, 3, 'hello', 'world', True]

# Modifying a list element
my_list[0] = 10
print(my_list)  # Output: [10, 2, 3, 'hello', 'world', True]
```

.Tuples:
   Tuples are ordered and immutable collections represented within parentheses (). They are similar to list s but cannot be modified after creation.
example:
```
# Creating a tuple
my_tuple = (1, 2, 'apple', 'banana', True)
print(my_tuple)  # Output: (1, 2, 'apple', 'banana', True)

# Accessing tuple elements
print(my_tuple[2])  # Output: 'apple'
```

.Dictionaries:
  Dictionaries are unordered collections of key-value pairs enclosed within curly braces {}. They are used f or storing data in a structured manner where values are accessed via unique keys.

```
# Creating a dictionary
my_dict = {'name': 'Alice', 'age': 30, 'city': 'New York'}
print(my_dict)  # Output: {'name': 'Alice', 'age': 30, 'city': 'New York'}

# Accessing dictionary values using keys
print(my_dict['age'])  # Output: 30

# Modifying dictionary values
my_dict['age'] = 35
print(my_dict)  # Output: {'name': 'Alice', 'age': 35, 'city': 'New York'}
```

.Sets:
     Sets are unordered collections of unique elements enclosed within curly braces {}. They eliminate dupli cate values automatically.

```
# Creating a set
my_set = {3, 6, 2, 3, ’apple’, ’banana’, ’apple’}
print(my_set)  # Output: {2, 3, 6, ’apple’, ’banana’}

# Performing set operations
set1 = {1, 2, 3, 4}
set2 = {3, 4, 5, 6}

# Union of sets
union_set = set1 | set2
print(union_set)  # Output: {1, 2, 3, 4, 5, 6}

# Intersection of sets
intersection_set = set1 & set2
print(intersection_set)  # Output: {3, 4}
```

5Q.Explain about List comprehension with example?
Ans:
List comprehension in Python is a concise and efficient way to create lists by applying an expression to e ach element from an existing iterable and optionally applying filtering conditions. It allows for the creation of new lists in a single line of code, making code more readable and compact compared to traditional for l oops.

Syntax:
```
new_list = [expression for item in iterable if condition]
```

.expression is the operation or transformation to be applied to each element.
.item is a variable representing each element in the iterable.
.iterable is the existing collection or sequence used to generate the new list.
.condition (optional) is a filtering condition that selects only certain elements based on a specified criterion .

example:
```
#program
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]
newlist = [x for x in fruits if "a" in x]
print(newlist)
#output:
 [’apple’, ’banana’, ’mango’]
```

6Q.Explain about Dictionary comprehension with example?
Ans:Dictionary comprehension is a concise and efficient way to create dictionaries in Python by applying an expression to each element of an iterable and forming key-value pairs.

syntax:
```
new_dict = {key_expression: value_expression for item in iterable if condition}
```

.key_expression: Expression defining the key of the dictionary.
.value_expression: Expression defining the value of the dictionary.
.item: Variable representing each element in the iterable.
.iterable: Existing collection or sequence used to generate the dictionary.
.condition (optional): Filtering condition to select certain elements based on a specified criterion.

```
#program:

lis =["apple","apple","banana","pineapple"]
diccom ={num:name for num,name in enumerate(lis)}
print(diccom)

#output:
{0: 'apple', 1: 'apple', 2: 'banana', 3: 'pineapple'}
```

7Q.What are Functions? Explain about Local and Global Variable?
Ans:
def for function:-
   In Python, a function is a reusable block of code that performs a specific task or set of tasks. It allows y ou to encapsulate a sequence of statements into a single unit, making the code modular, more organized, and easier to understand. Functions can accept input arguments, perform operations based on those inp uts, and optionally return a result.
example for function:

```
#function to greet someone.
   def greet(name):
   print(f"Hello, {name}!")
#calling function.
greet("Alice")  # Output: Hello, Data science!
```

"Local and global variable":

1.Local Variables: Variables defined inside a function have a local scope and are accessible only within th at function. They are created when the function is called and destroyed when the function execution ends.

example:

```
a=1 #this is global varible
def myfunction():
   b =2 # this local varible
   print("a=",a)# display global varible
   print("b=",b)# display local varible
myfunction()
#out put :
a= 1 #global varible can be accessed from any where inside or out side of the function also
b= 2 #here b is printed because it have been called inside the function..
```

2.Global Variables: Variables defined outside of any function or declared using the global keyword within a function have a global scope and can be accessed throughout the program.
example:

```
a=1 #this is global varible
def myfunction():
   b =2 # this local varible
   print("a=",a)# display global varible
   print("b=",b)# display local varible
myfunction()
print(a)# available because it is global right so.. #out put 1
print(b)# error , not available because  it is local right so..
```

8Q.Explain about Anonymous (lambda) Functions?
Ans:

A lambda function is a small anonymous function.
A lambda function can take any number of arguments, but can only have one expression.

Syntax
lambda arguments : expression

example :

```
x = lambda a : a + 10
print(x(5)) #output : 15
```

```
# Using lambda with map() to calculate squares of numbers in a list
numbers = [1, 2, 3, 4, 5]
squared_numbers = list(map(lambda x: x**2, numbers))
print(squared_numbers)  # Output: [1, 4, 9, 16, 25]
```

9Q.Explain about importance of data, types of data and sources of data?

Importance of Data:
Data is a fundamental asset in modern society, playing a crucial role in decision-making, analysis, innovat ion, and problem-solving across various domains. Its importance lies in several aspects.
.Basis for Decision-making: Data serves as the foundation for informed decision-making, providing insight s and evidence to support strategic choices.
.Business Intelligence: It enables organizations to understand customer behavior, market trends, and ope rational efficiency, aiding in optimizing business strategies.
.Innovation and Research: Data fuels research, innovation, and technological advancements by facilitatin g experimentation, analysis, and discovery.
.Personalization and User Experience: Data allows businesses to personalize products, services, and exp eriences by understanding individual preferences and behaviors.
.Predictive Analysis: It enables predictive modeling and forecasting, helping in anticipating trends, risks, a nd outcomes.

Types of Data:
.Structured Data: Organized data with a defined format, often stored in databases or spreadsheets. Exam ples include numerical values, dates, and categories in tabular formats.
.Unstructured Data: Data lacking a predefined structure or format, such as text, images, videos, social me dia posts, and emails.
.Semi-structured Data: Data that is partially organized, often having some structure but not conforming to a strict schema. Examples include JSON, XML, and CSV files.

Sources of Data:
.Internal Sources: Data generated and collected within an organization, including databases, sales record s, customer information, and operational data.
.External Sources: Data obtained from sources outside the organization, such as:
.Public Databases: Government data repositories, census data, weather reports, etc.
.Third-party Providers: Data obtained from vendors, partners, or external agencies.
.Web and Social Media: Data collected from websites, social media platforms, blogs, and online forums.
.Sensors and IoT Devices: Data generated by sensors, IoT devices, wearables, and smart technology.
.Surveys and Feedback: Data collected through surveys, feedback forms, market research, and user resp onses.

10Q.What are the different types of data quality issues in process of Data wrangling?
During the process of data wrangling, which involves cleaning, transforming, and preparing raw data for a nalysis, various data quality issues can arise. Addressing these issues is essential to ensure the accuracy

, consistency, and reliability of the data being used for analysis. Here are different types of data quality iss ues commonly encountered in data wrangling:

Missing Values:
Occurs when data entries are incomplete or absent for certain records or attributes.
Handling missing values involves strategies such as imputation (replacing missing values with estimated or calculated values), deletion, or flagging missing values for later analysis.

Inconsistent Formatting:
Inconsistencies in data formats, such as variations in date formats, different representations of categorical data, or inconsistent units of measurement.
Standardizing formats ensures uniformity, making data consistent and easier to analyze.

Duplicate Entries:
Repetition of identical or very similar records in the dataset.
Identifying and removing duplicates helps maintain data integrity and prevents skewing of analysis results .

Outliers:
Data points that significantly differ from the rest of the dataset, potentially influencing statistical analysis or machine learning models.
Techniques like truncation, transformation, or removing outliers after careful examination can be employe d to handle outlier data.

Inaccurate Data:
Errors or inaccuracies in the data due to human error, measurement discrepancies, or outdated informatio n.
Verification, cross-referencing with external sources, and manual inspection can help identify and rectify i naccurate data.

Data Inconsistency:
Contradictory information or inconsistencies across different datasets or fields within the same dataset.
Resolving inconsistencies involves data validation, reconciliation, and ensuring that data from multiple so urces align correctly.

Incompatible Data Types:
Mismatched data types within the same column or attribute, causing issues during analysis or calculations .
Converting data types to a consistent format across the dataset resolves compatibility issues.

Data Entry Errors:
Typos, misspellings, or incorrect entries made during data collection or data entry processes.
Data cleaning techniques such as fuzzy matching, pattern recognition, and validation rules help identify a nd correct entry errors.

Biased Data Sampling:
Biases in the data collection process leading to skewed representations of certain groups or populations.
Addressing biased sampling involves ensuring representative sampling methods and minimizing biases d uring data collection.

################## The end ## ##