SVKM's NMIMS
Data Warehousing and Data Mining
BTech CS- Cyber
Sem IV ,2nd Year
Academic Year :2024-25

Name: Arjun Mehta
K036
B. Tech CSE Cybersecurity

Semester 4
28-01-2025

Lab Experiment

Aim: To analyze and implement Exploratory Data Analysis

Analyze the following Python libraries

1. Numpy
2. Pandas
3. NLTK
4. SCIKIT-LEARN

Analyze the below for EACH Python LIBRARY GIVEN above :

1. Significance
2. Functions /Features
3. EXAMPLE code (in Python)

1. Numpy

1. Significance:

Numpy is one of the foundational libraries for scientific computing in Python. It provides support for arrays, matrices, and large multi-dimensional data structures, as well as a wide

range of mathematical functions to operate on these arrays. It is heavily used for numerical computing and is a key library in data science, machine learning, and artificial intelligence applications.

2. Functions / Features:

- **Array Creation**: You can create Numpy arrays (ndarray) with ease, including multi-dimensional arrays.
- **Mathematical Functions**: Functions for arithmetic, trigonometric, statistical, and linear algebra operations.
- **Array Manipulation**: Reshaping, stacking, and splitting arrays.
- **Broadcasting**: Allowing operations on arrays of different shapes.
- **Random Module**: Generate random numbers, samples, and distributions.
- **Efficiency**: Optimized for performance, better than Python's native list structures for numerical tasks.

3. Example Code:

```
import numpy as np

# Creating a 2D Numpy array
array = np.array([[1, 2, 3], [4, 5, 6]])

# Array operations
sum_array = np.sum(array)  # Summing all elements in the array
mean_array = np.mean(array)  # Calculating the mean of array elements

print("Array:\n", array)
print("Sum of all elements:", sum_array)
print("Mean of array elements:", mean_array)
```

```
Array:
 [[1 2 3]
 [4 5 6]]
Sum of all elements: 21
Mean of array elements: 3.5
```

```
1
```

Observation:

Numpy enhances Python's capabilities for handling numerical data efficiently with minimal code. Its performance advantages over native Python structures are evident when working with large datasets.

Conclusion:

Numpy is essential for anyone working with large datasets, performing numerical calculations, or doing data analysis in Python.

---

2. Pandas

1. Significance:

Pandas is the go-to library for data manipulation and analysis in Python. It is used to handle structured data (e.g., CSV, Excel, SQL databases) and provides fast, flexible, and expressive data structures (Series and DataFrame). It allows for easy data wrangling, transformation, and cleaning.

2. Functions / Features:

- **DataFrame**: A 2D structure for storing data in rows and columns.
- **Series**: A one-dimensional array-like object.
- **Data Import/Export**: Read/write data from/to various formats like CSV, Excel, SQL, JSON, etc.
- **Data Cleaning**: Handle missing values, duplicates, and apply transformations.
- **Grouping**: Aggregate data based on categories.
- **Time Series**: Handle date and time-based data effectively.

3. Example Code:

```
import pandas as pd

# Creating a DataFrame
data = {'Name': ['Alice', 'Bob', 'Charlie'],
        'Age': [25, 30, 35],
        'City': ['New York', 'Los Angeles', 'Chicago']}
df = pd.DataFrame(data)

# Basic operations
mean_age = df['Age'].mean()  # Calculating mean age
city_count = df['City'].value_counts()  # Count occurrences of cities

print("DataFrame:\n", df)
print("Mean Age:", mean_age)
print("City count:\n", city_count)
```

```
DataFrame:
      Name  Age         City
0    Alice   25     New York
1      Bob   30  Los Angeles
2  Charlie   35      Chicago
Mean Age: 30.0
City count:
 New York       1
Los Angeles    1
Chicago        1
Name: City, dtype: int64
```

Observation:

Pandas makes it simple to work with structured data, allowing for quick manipulation, summarization, and analysis of large datasets.

Conclusion:

Pandas is indispensable for data scientists and analysts who need to clean, manipulate, and analyze data quickly and efficiently.

---

3. NLTK (Natural Language Toolkit)

1. Significance:

NLTK is one of the most popular libraries for text processing and natural language processing (NLP) in Python. It provides easy-to-use interfaces for over 50 corpora and lexical resources, such as WordNet, and offers tools for classification, tokenization, stemming, tagging, parsing, and semantic reasoning.

2. Functions / Features:

- **Text Tokenization**: Breaking text into words or sentences.
- **Text Classification**: Classify text into categories.
- **Stemming & Lemmatization**: Reduce words to their root forms.
- **POS (Part of Speech) Tagging**: Identify word types in sentences.
- **Corpora**: Access to various datasets and text corpora.
- **NLP Operations**: Named Entity Recognition (NER), sentence parsing, and more.

3. Example Code:

```
import nltk
from nltk.tokenize import word_tokenize

# Example text
text = "Hello, how are you doing today?"

# Tokenize text into words
words = word_tokenize(text)

print("Tokenized Words:", words)
```

```
words = word_tokenize(text)

print("Tokenized Words:", words)
```

```
Tokenized Words: ['Hello', ',', 'how', 'are', 'you', 'doing', 'today', '?']
```

```
1
```

Observation:

NLTK offers comprehensive tools for anyone working on natural language processing, from beginners to experts.

Conclusion:

NLTK is essential for anyone involved in NLP and text mining. Its vast collection of tools and corpora makes it a powerful library for language analysis and processing tasks.

---

4. Scikit-learn

1. Significance:

Scikit-learn is a widely-used machine learning library in Python. It provides simple and efficient tools for data mining and data analysis. It is built on top of Numpy, SciPy, and matplotlib and focuses on machine learning algorithms for classification, regression, clustering, and dimensionality reduction.

2. Functions / Features:

- **Supervised Learning**: Algorithms like Linear Regression, Decision Trees, SVM, etc.
- **Unsupervised Learning**: Algorithms like K-Means, DBSCAN, PCA, etc.
- **Model Evaluation**: Tools for cross-validation, scoring metrics (accuracy, F1 score, etc.).
- **Feature Selection**: Select important features to improve model performance.
- **Pipeline**: Build workflows that streamline training and prediction tasks.
- **Preprocessing**: Scaling, encoding, and imputation of data.

3. Example Code:

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score

# Load dataset
iris = load_iris()
X = iris.data
y = iris.target

# Split data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)

# Create a SVM classifier
clf = SVC()

# Train the model
clf.fit(X_train, y_train)

# Make predictions
y_pred = clf.predict(X_test)

# Evaluate model performance
accuracy = accuracy_score(y_test, y_pred)
print("Model Accuracy:", accuracy)
```

```
23 # Evaluate model performance
24 accuracy = accuracy_score(y_test, y_pr
25 print("Model Accuracy:", accuracy)
```

```
Model Accuracy: 1.0
```

```
]:  1
```

Observation:

Scikit-learn offers a straightforward approach to implementing machine learning algorithms, making it a perfect tool for beginners and experienced practitioners alike.

Conclusion:

Scikit-learn is an essential library for building and evaluating machine learning models in Python. It is versatile, efficient, and easy to use.

Final Observations and Conclusion:

- **Numpy** is essential for numerical computing, enabling efficient handling of large data sets and mathematical computations.
- **Pandas** excels in data manipulation and is indispensable for data wrangling and exploration.
- **NLTK** provides the necessary tools for text processing and natural language processing, making it an essential library for those working in NLP.
- **Scikit-learn** offers a comprehensive suite of machine learning tools that can be used to implement and evaluate algorithms easily.