

**1) State Regression Algorithms in Python.****Ans. 1.a)**

Regression algorithms are used to make predictions about continuous variables, such as housing prices, student scores, or medical outcomes. Python, being one of the most widely used programming languages in data science and machine learning, has a variety of powerful libraries for implementing regression algorithms.

**1. Linear Regression**

Linear Regression is an ML algorithm used for supervised learning. Linear regression performs the task to predict a dependent variable(target) based on the given independent variable(s). So, this regression technique finds out a linear relationship between a dependent variable and the other given independent variables. Hence, the name of this algorithm is Linear Regression.

**2. Logistic Regression**

Logistic Regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.). In other words, the logistic regression model predicts  $P(Y=1)$  as a function of  $X$ .

**3. Lasso Regression**

- LASSO stands for Least Absolute Selection Shrinkage Operator. Shrinkage is basically defined as a constraint on attributes or parameters.
- The algorithm operates by finding and applying a constraint on the model attributes that cause regression coefficients for some variables to shrink toward a zero.
- Variables with a regression coefficient of zero are excluded from the model.
- So, lasso regression analysis is basically a shrinkage and variable selection method, and it helps to determine which of the predictors are most important.

**4. Decision Trees**

The decision tree models can be applied to all those data which contains numerical features and categorical features. Decision trees are good at capturing non-linear interaction between the features and the target variable. Decision trees somewhat match human-level thinking so it's very intuitive to understand the data.

**5. Support Vector Regression (SVR)**

Support Vector Machine. SVR also uses the same idea of SVM but here it tries to predict the real values. This algorithm uses hyperplanes to segregate the data. In case this separation is not possible then it uses kernel trick where the dimension is increased and then the data points become separable by a hyperplane.

## 2) Describe the Following term in Python

i) Numpy

ii) Pandas

iii) Scikit-Learn

**Ans. 2)**

### **i) Numpy**

- NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices.
- NumPy was created in 2005 by Travis Oliphant.
- It is an open-source project, and you can use it freely.
- NumPy stands for Numerical Python.
- NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.
- The array object in NumPy is called ndarray
- It is optimized to work with latest CPU architectures.

### **ii) Pandas**

- Pandas is a Python library used for working with data sets.
- It has functions for analysing, cleaning, exploring, and manipulating data.
- The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008.
- Pandas allows us to analyse big data and make conclusions based on statistical theories.
- Pandas can clean messy data sets and make them readable and relevant.
- Relevant data is very important in data science.

### **iii) Scikit-Learn**

- Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python.
- It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python.
- This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

### 3) Justify the Data Visualization with Matplotlib and Seaborn in Python

#### Ans. 3)

Data visualization with Matplotlib and Seaborn in Python is justified for several reasons:

1. **Ease of Use:** Matplotlib is a widely used and easy-to-learn plotting library in Python. It provides a simple and consistent interface for creating a wide variety of plots, from basic line charts to complex 3D plots. Seaborn, built on top of Matplotlib, further enhances the default visualizations and provides a high-level interface for statistical graphics.
2. **Flexibility:** Matplotlib and Seaborn offer a high degree of customization. You can control almost every aspect of a plot, from the axis labels to the colors and styles. This flexibility is crucial when you need to tailor your visualizations to specific requirements or when presenting data in a way that is most meaningful to your audience.
3. **Publication Quality Graphics:** Both libraries are capable of producing publication-quality graphics suitable for presentations, reports, and academic publications. This is important when communicating insights and findings derived from your data to a wider audience.
4. **Integration with Jupyter Notebooks:** Matplotlib and Seaborn work seamlessly with Jupyter Notebooks, making them the go-to choices for data exploration and analysis in a notebook environment. The ability to visualize data interactively within a notebook enhances the iterative and exploratory nature of data analysis.
5. **Rich Set of Plot Types:** Matplotlib supports a wide range of plot types, including line plots, scatter plots, bar plots, histograms, pie charts, and more. Seaborn, while providing a higher-level interface, specializes in statistical visualization and offers aesthetically pleasing defaults for many common plot types.
6. **Statistical Visualizations with Seaborn:** Seaborn is particularly useful for statistical data visualization. It simplifies the process of creating informative and attractive statistical graphics, such as violin plots, box plots, and pair plots. These visualizations can reveal patterns, trends, and relationships within the data.
7. **Community and Documentation:** Matplotlib and Seaborn have large and active communities, which means ample resources are available for learning and problem-solving. The documentation for both libraries is extensive, making it easy to find information and examples for creating specific types of visualizations.
8. **Integration with Pandas:** Matplotlib and Seaborn integrate seamlessly with Pandas DataFrames, which are commonly used for data manipulation and analysis in Python. This makes it easy to visualize data directly from Pandas, creating a smooth workflow for data exploration and visualization.

The combination of Matplotlib and Seaborn in Python is justified due to their ease of use, flexibility, ability to create publication-quality graphics, support for various plot types, statistical visualization capabilities, community support, and integration with popular data analysis tools like Jupyter Notebooks and Pandas.

#### 4) Distinguish between Supervised and Unsupervised learning

**Ans. 4)**

Supervised vs. Unsupervised Learning:

Parameters	Supervised machine learning	Unsupervised machine learning
Input Data	Algorithms are trained using labelled data.	Algorithms are used against data that is not labelled
Computational Complexity	Simpler method	Computationally complex
Accuracy	Highly accurate	Less accurate
No. of classes	No. of classes is known	No. of classes is not known
Data Analysis	Uses offline analysis	Uses real-time analysis of data
Algorithms used	Linear and Logistics regression, Random Forest, Support Vector Machine, Neural Network, etc.	K-Means clustering, Hierarchical clustering, Apriori algorithm, etc.
Output	Desired output is given.	Desired output is not given.
Training data	Use training data to infer model.	No training data is used.
Complex model	It is not possible to learn larger and more complex models than with supervised learning.	It is possible to learn larger and more complex models with unsupervised learning.
Model	We can test our model.	We cannot test our model.
Called as	Supervised learning is also called classification.	Unsupervised learning is also called clustering.
Example	Example: Optical character recognition.	Example: Find a face in an image.

## 5) Interpret Flask Extension for database integration Concept in Python.

### Ans. 5)

In Flask, extensions are additional packages that add functionality to your application. When it comes to database integration, Flask provides extensions that simplify the process of working with databases. One popular extension for this purpose is Flask-SQLAlchemy.

1. **Flask:** Flask is a lightweight web framework for Python. It provides the tools and libraries needed to build a web application, but it doesn't dictate the choice of tools for other concerns like database management.
2. **Database Integration:** When you're building a web application, you often need to store and retrieve data. Databases are the go-to solution for this. However, interacting with databases can be complex, involving SQL queries, connection management, and more. Flask extensions for database integration streamline this process.
3. **Flask-SQLAlchemy:** This is a Flask extension that adds support for SQLAlchemy to Flask applications. SQLAlchemy is an Object-Relational Mapping (ORM) library for Python, providing a set of high-level API for interacting with relational databases.
  - **Object-Relational Mapping (ORM)**
  - **SQLAlchemy**
  -
4. **How it Works:**
  - **Models:** You define your database models as Python classes, where each class represents a table in your database, and each instance of the class represents a row in that table.
  - **Queries:** You can perform database operations using SQLAlchemy queries, which are much more Pythonic and abstracted than raw SQL queries.
  - **Migration:** Flask-SQLAlchemy also provides tools for managing database migrations, which are changes to your database schema over time.