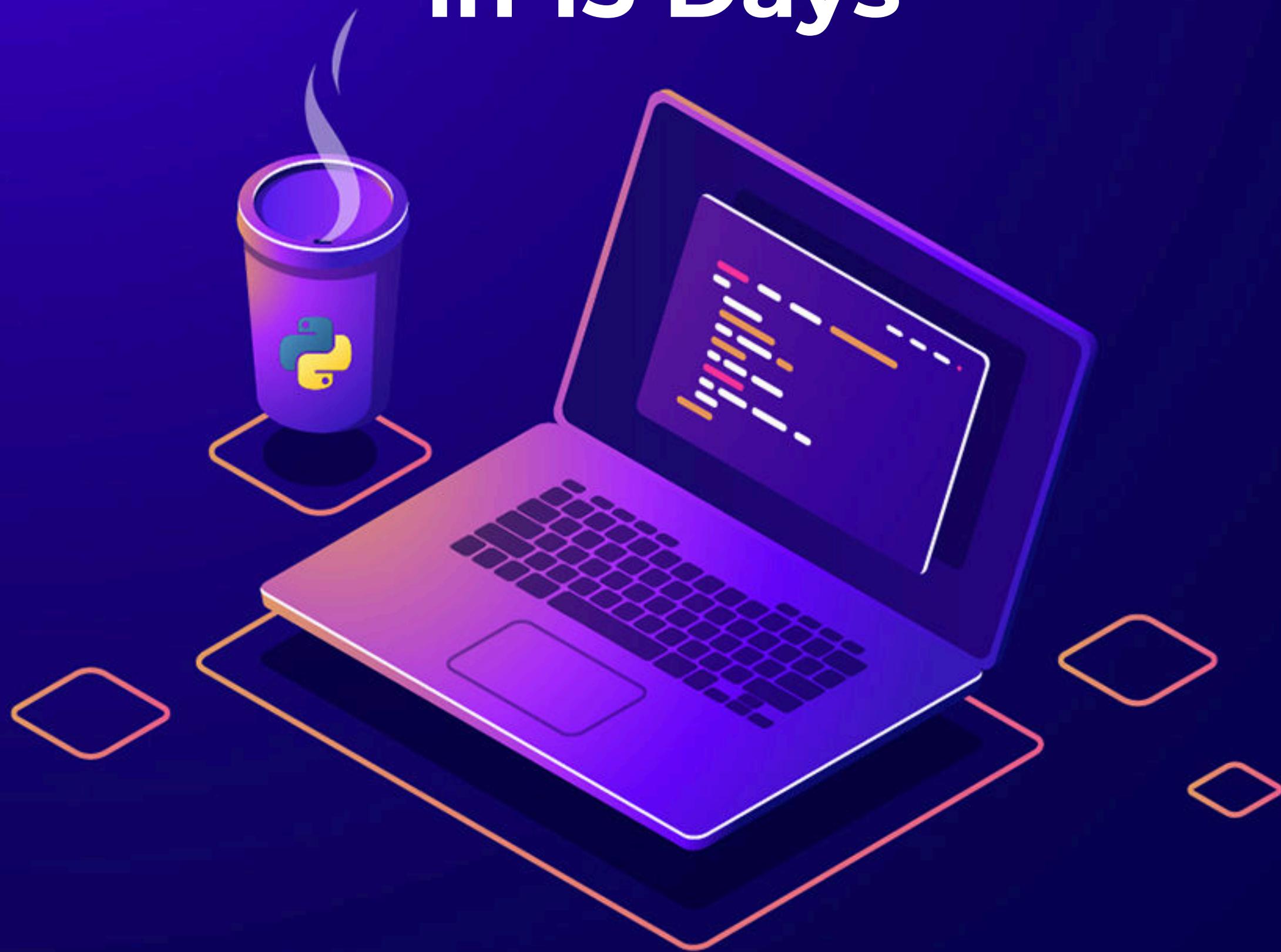


MASTER

PYTHON

in 15 Days



INTERMEDIATE → ADVANCE

Your Ultimate Guide to Data Science & ML

SUPERVISED LEARNING - CLASSIFICATION

Aim

Understand and implement supervised learning classification algorithms in Python.

Resources

- Scikit-Learn Classification Algorithms: https://scikit-learn.org/stable/supervised_learning.html#supervised-learning

Example Questions

Q 1: Train a Logistic Regression classifier on a dataset.

Q 2: Train a Support Vector Machine (SVM) classifier on a dataset.

Q 3: Train a Decision Tree classifier on a dataset.



Practice Questions

1. Given a CSV file with data about customers (features) and their churn status (target), train a Logistic Regression classifier.
2. Implement a program that uses Scikit-Learn to train a k-nearest neighbors (KNN) classifier on a dataset.
3. Write a Python program that uses Scikit-Learn to train a Random Forest classifier on a dataset.
4. Create a function that takes a Pandas DataFrame and trains a Gaussian Naive Bayes classifier on the data.
5. Given a CSV file with data about student scores (features) and their grades (target), train a Support Vector Machine (SVM) classifier.
6. Implement a program that uses Scikit-Learn to train a Gradient Boosting classifier on a dataset.
7. Write a Python program that uses Scikit-Learn to train a Multilayer Perceptron (MLP) classifier on a dataset.
8. Create a function that takes a Pandas DataFrame and trains a Quadratic Discriminant Analysis (QDA) classifier on the data.
9. Given a CSV file with data about housing prices (features) and their labels (target), train a Decision Tree classifier.
10. Implement a program that uses Scikit-Learn to train a Stochastic Gradient Descent (SGD) classifier on a dataset.

SUPERVISED LEARNING - REGRESSION

Aim

Understand and implement supervised learning regression algorithms in Python.

Resources

- Scikit-Learn Regression Algorithms: https://scikit-learn.org/stable/supervised_learning.html#supervised-learning

Example Questions

Q 1: Train a Linear Regression model on a dataset.

Q 2: Train a Decision Tree Regressor on a dataset.

Q 3: Train a Support Vector Regression (SVR) model on a dataset.



Practice Questions

1. Given a CSV file with data about house prices (features) and their sale prices (target), train a Linear Regression model.
2. Implement a program that uses Scikit-Learn to train a Ridge Regression model on a dataset.
3. Write a Python program that uses Scikit-Learn to train a Lasso Regression model on a dataset.
4. Create a function that takes a Pandas DataFrame and trains a Random Forest Regressor on the data.
5. Given a CSV file with data about car features (features) and their prices (target), train a Decision Tree Regressor.
6. Implement a program that uses Scikit-Learn to train a Gradient Boosting Regressor on a dataset.
7. Write a Python program that uses Scikit-Learn to train a K-Nearest Neighbors (KNN) Regressor on a dataset.
8. Create a function that takes a Pandas DataFrame and trains a Support Vector Regression (SVR) model on the data.
9. Given a CSV file with data about stock prices (features) and their future prices (target), train a Neural Network Regressor.
10. Implement a program that uses Scikit-Learn to train a Random Forest Regressor with hyperparameter tuning using Grid Search.

UNSUPERVISED LEARNING - CLUSTERING

Aim

Understand and implement unsupervised learning clustering algorithms in Python.

Resources

- Scikit-Learn Clustering Algorithms: https://scikit-learn.org/stable/unsupervised_learning.html#unsupervised-learning

Example Questions

Q 1: Perform K-Means clustering on a dataset.

Q 2: Perform Hierarchical clustering on a dataset.

Q 3: Perform DBSCAN clustering on a dataset.



Practice Questions

1. Given a CSV file with data about customer purchases (features), perform K-Means clustering on the data.
2. Implement a program that uses Scikit-Learn to perform Hierarchical clustering on a dataset.
3. Write a Python program that uses Scikit-Learn to perform DBSCAN clustering on a dataset.
4. Create a function that takes a Pandas DataFrame and performs K-Means clustering with different numbers of clusters.
5. Given a CSV file with data about student performance (features), perform Agglomerative clustering on the data.
6. Implement a program that uses Scikit-Learn to perform Spectral clustering on a dataset.
7. Write a Python program that uses Scikit-Learn to perform Affinity Propagation clustering on a dataset.
8. Create a function that takes a Pandas DataFrame and performs Mini-Batch K-Means clustering on the data.
9. Given a CSV file with data about credit card transactions (features), perform DBSCAN clustering with different hyperparameters.
10. Implement a program that uses Scikit-Learn to perform Mean Shift clustering on a dataset.

UNSUPERVISED LEARNING - DIMENSIONALITY REDUCTION

Aim

Understand and implement unsupervised learning dimensionality reduction techniques in Python.

Resources

- Scikit-Learn Dimensionality Reduction: https://scikit-learn.org/stable/unsupervised_learning.html#dimensionality-reduction

Example Questions

Q 1: Perform Principal Component Analysis (PCA) on a dataset.

Q 2: Perform t-distributed Stochastic Neighbor Embedding (t-SNE) on a dataset.

Q 3: Perform Singular Value Decomposition (SVD) on a dataset.



Practice Questions

1. Given a CSV file with data about customer transactions (features), perform PCA on the data.
2. Implement a program that uses Scikit-Learn to perform t-SNE on a dataset.
3. Write a Python program that uses Scikit-Learn to perform LLE (Locally Linear Embedding) on a dataset.
4. Create a function that takes a Pandas DataFrame and performs SVD on the data.
5. Given a CSV file with data about images (features), perform PCA with different numbers of components.
6. Implement a program that uses Scikit-Learn to perform Isomap on a dataset.
7. Write a Python program that uses Scikit-Learn to perform NMF (Non-Negative Matrix Factorization) on a dataset.
8. Create a function that takes a Pandas DataFrame and performs Kernel PCA on the data.
9. Given a CSV file with data about gene expression (features), perform t-SNE with different perplexity values.
10. Implement a program that uses Scikit-Learn to perform MDS (Multi-Dimensional Scaling) on a dataset.

MODEL EVALUATION AND SELECTION

Aim

Understand how to evaluate and select the best machine learning model in Python.

Resources

- Scikit-Learn Model Evaluation: https://scikit-learn.org/stable/model_evaluation.html

Example Questions

Q 1: Evaluate a classification model using accuracy score.

Q 2: Evaluate a regression model using mean squared error.

Q 3: Use cross-validation to evaluate a model's performance.



Practice Questions

1. Given a CSV file with data about customer churn (features) and their churn status (target), evaluate a classification model using accuracy.
2. Implement a program that uses Scikit-Learn to evaluate a regression model using mean squared error.
3. Write a Python program that uses cross-validation to evaluate a Support Vector Machine (SVM) classifier.
4. Create a function that takes a Pandas DataFrame and evaluates a Random Forest classifier using F1-score.
5. Given a CSV file with data about house prices (features) and their sale prices (target), evaluate a regression model using R-squared score.
6. Implement a program that uses Scikit-Learn to evaluate a K-Nearest Neighbors (KNN) classifier using precision and recall.
7. Write a Python program that uses cross-validation to evaluate a Decision Tree Regressor using mean absolute error.



Practice Questions

8. Create a function that takes a Pandas DataFrame and evaluates a Gradient Boosting classifier using ROC-AUC score.
9. Given a CSV file with data about student scores (features) and their grades (target), evaluate a regression model using root mean squared error.
10. Implement a program that uses Scikit-Learn to evaluate a Naive Bayes classifier using confusion matrix and classification report.

INTRODUCTION TO DEEP LEARNING

Aim

Get introduced to Deep Learning concepts and libraries in Python.

Resources

- Introduction to Deep Learning: <https://www.analyticsvidhya.com/blog/2020/02/introduction-to-deep-learning/>
- Keras Official Website: <https://keras.io/>
- TensorFlow Official Website: <https://www.tensorflow.org/>

Example Questions

Q 1: Create a simple Neural Network using Keras.

Q 2: Train a Neural Network on a dataset using Keras.



Practice Questions

1. Given a CSV file with data about customer behavior (features), create a Neural Network using Keras for binary classification.
2. Implement a program that uses Keras to train a Neural Network for multiclass classification on a dataset.
3. Write a Python program that uses Keras to create a Deep Neural Network for regression on a dataset.
4. Create a function that takes a Pandas DataFrame and trains a Neural Network with multiple hidden layers using Keras.
5. Given a CSV file with image data (features) and corresponding labels (target), create a Convolutional Neural Network (CNN) using Keras.
6. Implement a program that uses Keras to create a Recurrent Neural Network (RNN) for sequence data classification on a dataset.
7. Write a Python program that uses Keras to create a Long Short-Term Memory (LSTM) network for time series prediction.



Practice Questions

8. Create a function that takes a Pandas DataFrame and trains a GAN (Generative Adversarial Network) using Keras for image generation.
9. Given a CSV file with audio data (features) and corresponding labels (target), create a Deep Neural Network using Keras for audio classification.
10. Implement a program that uses Keras to create a Transfer Learning model using pre-trained weights for a specific task.

NATURAL LANGUAGE PROCESSING (NLP) BASICS

Aim

Introduction to Natural Language Processing (NLP) and NLP libraries in Python.

Resources

- Natural Language Processing with Python: <https://www.nltk.org/book/>
- NLTK Official Website: <https://www.nltk.org/>

Example Questions

Q 1: Tokenize text using NLTK.

Q 2: Perform text normalization (lowercasing) on a document.

Q 3: Remove stopwords from a document.



Practice Questions

1. Given a text file with multiple sentences, tokenize the sentences and words using NLTK.
2. Implement a program that uses NLTK to remove punctuation marks from a document.
3. Write a Python program that uses NLTK to perform stemming on a document.
4. Create a function that takes a Pandas DataFrame and removes special characters from a text column using NLTK.
5. Given a text file with multiple paragraphs, tokenize the paragraphs using NLTK.
6. Implement a program that uses NLTK to convert all words in a document to their base or root form (lemmatization).
7. Write a Python program that uses NLTK to count the frequency of each word in a document.
8. Create a function that takes a Pandas DataFrame and performs named entity recognition on a text column using NLTK.
9. Given a text file with multiple sentences, tokenize the sentences and remove numbers using NLTK.
10. Implement a program that uses NLTK to perform part-of-speech tagging on a document.

ADVANCED NLP TECHNIQUES

Aim

Learn advanced Natural Language Processing (NLP) techniques in Python.

Resources

- Text Classification with Python and Scikit-Learn: <https://realpython.com/python-keras-text-classification/>
- spaCy Official Website: <https://spacy.io/>

Example Questions

Q 1: Perform sentiment analysis using a pre-trained model.

Q 2: Perform text classification using Scikit-Learn.

Q 3: Perform named entity recognition using spaCy.



Practice Questions

1. Given a CSV file with text data and corresponding sentiment labels, perform sentiment analysis using NLTK.
2. Implement a program that uses Scikit-Learn to perform text classification using TF-IDF vectorization.
3. Write a Python program that uses spaCy to perform named entity recognition on a text document.
4. Create a function that takes a Pandas DataFrame and uses spaCy to extract noun phrases from a text column.
5. Given a text file with multiple paragraphs, extract the most important keywords using TF-IDF and Scikit-Learn.
6. Implement a program that uses spaCy to perform dependency parsing on a text document.
7. Write a Python program that uses Scikit-Learn to train a text classification model using word embeddings (e.g., Word2Vec).
8. Create a function that takes a Pandas DataFrame and performs text summarization using NLTK or spaCy.
9. Given a text file with multiple sentences, identify the language of each sentence using NLTK.
10. Implement a program that uses Scikit-Learn to perform topic modeling on a text document using Latent Dirichlet Allocation (LDA).

WEB SCRAPING WITH PYTHON

Aim

Learn how to scrape data from websites using Python libraries.

Resources

- Web Scraping with Python: A Comprehensive Guide: <https://realpython.com/beautiful-soup-web-scraping-python/>
- BeautifulSoup Official Documentation: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

Example Questions

Q 1: Scrape data from a website using BeautifulSoup.

Q 2: Scrape data from a paginated website.

Q 3: Scrape data from a website with dynamic content (using Selenium).



Practice Questions

1. Scrape data from a website and save it as a CSV file.
2. Implement a program that scrapes product details (name, price, description) from an e-commerce website.
3. Write a Python program that scrapes news headlines and summaries from a news website.
4. Create a function that scrapes job listings (title, company, location) from a job portal website.
5. Scrape data from a website and store it in a JSON file.
6. Implement a program that scrapes weather data (temperature, humidity, wind speed) from a weather website.
7. Write a Python program that scrapes stock market data (stock symbols, prices, volumes) from a financial website.
8. Create a function that scrapes movie details (title, genre, release date) from a movie database website.
9. Scrape data from a website and store it in a SQLite database.
10. Implement a program that scrapes recipe details (name, ingredients, instructions) from a cooking website.

WORKING WITH APIs

Aim

Learn how to work with APIs (Application Programming Interfaces) in Python.

Resources

- Introduction to APIs: <https://www.programmableweb.com/api-university/what-are-apis-and-how-do-they-work>
- Requests Official Documentation: <https://docs.python-requests.org/en/master/>

Example Questions

Q 1: Make GET requests to an API and extract data.

Q 2: Make POST requests to an API and send data.

Q 3: Work with authentication in API requests.



Practice Questions

1. Make a GET request to an API and extract specific data from the response.
2. Implement a program that makes multiple API requests and saves the data in a CSV file.
3. Write a Python program that uses authentication to make a POST request to an API.
4. Create a function that retrieves weather data (temperature, humidity, description) from a weather API.
5. Make a GET request to a financial API and extract stock market data (stock symbols, prices, volumes).
6. Implement a program that uses pagination in API requests to collect data from multiple pages.
7. Write a Python program that fetches movie details (title, genre, release date) from a movie database API.
8. Create a function that retrieves data (name, address, phone number) of restaurants from a restaurant API.
9. Make a GET request to a social media API and extract user data (username, followers, posts).
10. Implement a program that fetches book details (title, author, publication date) from a book database API.

DATA VISUALIZATION WITH MATPLOTLIB AND SEABORN

Aim

Learn how to create visualizations in Python using Matplotlib and Seaborn libraries.

Resources

- Data Visualization with Matplotlib: <https://matplotlib.org/stable/tutorials/introductory/pyplot.html>
- Data Visualization with Seaborn: <https://seaborn.pydata.org/tutorial.html>

Example Questions

Q 1: Create a simple line plot using Matplotlib.

Q 2: Create a bar plot using Seaborn.

Q 3: Create a scatter plot with multiple groups using Matplotlib.



Practice Questions

1. Create a line plot to visualize the trend of temperature over a week.
2. Implement a program that creates a bar plot to compare the sales of different products.
3. Write a Python program that uses Seaborn to create a scatter plot to show the relationship between two variables.
4. Create a function that generates a pie chart to represent the distribution of different categories.
5. Make a subplot with two line plots to compare the performance of two teams over time.
6. Implement a program that uses Seaborn to create a heatmap to visualize the correlation between features in a dataset.
7. Write a Python program that uses Matplotlib to create a histogram to show the distribution of exam scores.
8. Create a function that generates a box plot to compare the distribution of data across different groups.
9. Make a subplot with a line plot and a bar plot to visualize both the trend and the comparison of values.
10. Implement a program that uses Seaborn to create a pair plot to visualize the relationships between multiple variables in a dataset.

INTERACTIVE DATA VISUALIZATION WITH PLOTLY

Aim

Learn how to create interactive visualizations in Python using Plotly library.

Resources

- Plotly Python Library: <https://plotly.com/python/>
- Plotly Express: <https://plotly.com/python/plotly-express/>

Example Questions

Q 1: Create a simple line plot using Plotly.

Q 2: Create a bar plot using Plotly Express.

Q 3: Create a scatter plot with multiple groups using Plotly.



Practice Questions

1. Create an interactive line plot to visualize the trend of temperature over a week with Plotly.
2. Implement a program that creates an interactive bar plot to compare the sales of different products using Plotly Express.
3. Write a Python program that uses Plotly to create an interactive scatter plot to show the relationship between two variables.
4. Create a function that generates an interactive pie chart to represent the distribution of different categories using Plotly.
5. Make an interactive subplot with two line plots to compare the performance of two teams over time with Plotly.
6. Implement a program that uses Plotly to create an interactive heatmap to visualize the correlation between features in a dataset.
7. Write a Python program that uses Plotly to create an interactive histogram to show the distribution of exam scores.



Practice Questions

8. Create a function that generates an interactive box plot to compare the distribution of data across different groups using Plotly.
9. Make an interactive subplot with a line plot and a bar plot to visualize both the trend and the comparison of values using Plotly.
10. Implement a program that uses Plotly Express to create an interactive pair plot to visualize the relationships between multiple variables in a dataset.



WORKING WITH TIME SERIES DATA

Aim

Learn how to work with time series data in Python using Pandas and Matplotlib.

Resources

- Time Series Analysis with Python: <https://towardsdatascience.com/time-series-analysis-in-python-an-introduction-70d5a5b1d52a>
- Time Series Analysis in Python with Pandas: <https://www.datacamp.com/community/tutorials/time-series-analysis-tutorial>

Example Questions

Q 1: Read time series data from a CSV file using Pandas.

Q 2: Plot time series data using Matplotlib.

Q 3: Resample time series data to a lower frequency (e.g., from hourly to daily).



Practice Questions

1. Read time series data from a CSV file and plot it using Matplotlib.
2. Implement a program that calculates and plots the rolling mean of a time series using Pandas.
3. Write a Python program that uses Pandas to resample time series data to a lower frequency (e.g., from daily to weekly).
4. Create a function that generates a seasonal decomposition plot for a time series using Pandas and Matplotlib.
5. Calculate and plot the autocorrelation and partial autocorrelation of a time series using Pandas and Matplotlib.
6. Implement a program that uses Pandas to calculate and plot the trend, seasonality, and residuals of a time series.
7. Write a Python program that uses Pandas to calculate the moving average of a time series and plot it using Matplotlib.



Practice Questions

8. Create a function that generates a time series decomposition using seasonal decomposition of time series (STL) in Python.
9. Calculate and plot the autocorrelation and partial autocorrelation functions of a time series using Pandas and Matplotlib.
10. Implement a program that uses Pandas to detect and remove outliers in a time series and visualize the results using Matplotlib.

DATA WRANGLING AND CLEANING

Aim

Learn how to clean and wrangle data in Python using Pandas.

Resources

- Data Cleaning with Pandas: <https://towardsdatascience.com/data-cleaning-with-pandas-63b34858a3a6>
- Data Wrangling with Pandas: https://pandas.pydata.org/docs/user_guide/duplication.html

Example Questions

Q 1: Drop missing values from a DataFrame using Pandas.

Q 2: Remove duplicate rows from a DataFrame using Pandas.

Q 3: Merge two DataFrames based on a common key using Pandas.



Practice Questions

1. Drop missing values from a DataFrame and save the cleaned data to a new CSV file.
2. Implement a program that removes duplicate rows from a DataFrame and visualizes the cleaned data.
3. Write a Python program that uses Pandas to fill missing values in a DataFrame with the mean or median of the column.
4. Create a function that merges two DataFrames based on a common key and saves the merged data to a new CSV file.
5. Merge two DataFrames using an outer join and handle missing values appropriately.
6. Implement a program that renames columns in a DataFrame to make them more descriptive and meaningful.
7. Write a Python program that uses Pandas to convert categorical variables into numerical values using one-hot encoding.
8. Create a function that aggregates data in a DataFrame to calculate summary statistics for different groups.
9. Aggregate data in a DataFrame to calculate the mean, median, and standard deviation for each group.
10. Implement a program that performs data transformation in a DataFrame, such as scaling numerical features or applying log transformation.

PUTTING IT ALL TOGETHER - REAL-WORLD PROJECT

Aim

Work on a real-world project that involves multiple concepts and skills learned throughout the 30-day upskilling program.

Resources

- Real-World Datasets: Kaggle (<https://www.kaggle.com/datasets>) and UCI Machine Learning Repository (<https://archive.ics.uci.edu/ml/index.php>)



Project Guidelines

- Choose a real-world dataset of interest and download it.
- Clean and preprocess the data using Pandas to remove missing values, handle duplicates, and perform necessary data wrangling.
- Perform exploratory data analysis using various data visualization libraries (Matplotlib, Seaborn, Plotly) to gain insights into the dataset.
- Apply machine learning models (e.g., regression, classification, clustering) to solve a specific problem related to the dataset.
- Evaluate the performance of the models using appropriate metrics and techniques (e.g., cross-validation, hyperparameter tuning).
- Create visualizations and summary reports to present the findings of your analysis and model performance.
- Document the entire project, explaining the steps taken, insights gained, and conclusions made.



Project Steps

- Choose a dataset and import it into your Python environment.
- Perform data cleaning and preprocessing to get the data ready for analysis.
- Conduct exploratory data analysis to understand the distribution and relationships of the variables.
- Apply appropriate machine learning models to address a specific problem or task.
- Evaluate the performance of the models using suitable metrics and techniques.
- Visualize the results and present your findings in a clear and concise manner.
- Prepare a final report documenting the entire project and your learnings.



WHY BOSSCODER?

 **750+** Alumni placed at Top Product-based companies.

 More than **136% hike** for every **2 out of 3** working professional.

 Average package of **24LPA**.

The syllabus is most up-to-date and the list of problems provided covers all important topics.

Lavanya
 Meta



Course is very well structured and streamlined to crack any MAANG company

Rahul .
 Google



[EXPLORE MORE](#)