CONTENTS

CHAPTER 1

Internship Organisation Profile

1. Introduction

Anjanaas Foundation is a registered non-profit organisation based in Avanigadda, Krishna District, Andhra Pradesh. The foundation is dedicated to empowering individuals through education, skill development, health, agriculture, and community development. Built upon a decade-long legacy of educational service, Anjanaas Foundation has expanded its mission to address social protection and financial independence for the most vulnerable sections of society. The foundation plays a significant role in fostering experiential learning through its well-structured internship programmes.

2. Vision

To create a self-sustaining society by providing quality education, skill development, and opportunities for holistic growth, ensuring that every individual can contribute meaningfully to the community.

3. Mission

- To bridge the gap between education and employment by equipping individuals with industry-relevant skills.
- To provide inclusive learning environments that cater to diverse communities, including underprivileged and differently-abled individuals.
- To promote innovation, entrepreneurship, and sustainability through practical training programmes.

4. Objectives

- To offer structured internship programmes that provide hands-on experience in various disciplines.
- To collaborate with educational institutions to integrate internships into academic curricula.
- To enhance employability by offering training in high-demand skills such as IT, business management, agriculture, and life sciences.
- To promote research and development through skill-based learning initiatives.

5. Internship Programme Overview

a. Internship Scope and Reach

Anjanaas Foundation has provided more than 10,000 internship opportunities across various universities in Andhra Pradesh. The foundation collaborates with institutions such as Krishna University to formalise internship programmes through Memoranda of Understanding, ensuring academic recognition for participating students.

b. Key Training Areas

Information Technology and Digital Skills

Python, Data Science, Cloud Computing, Full Stack Development, Digital Marketing, DevOps, ServiceNow, MySQL, Tableau, Web Technologies, Generative AI, Machine Learning, Agentic AI.

Business and Management

Office Administration, Tally, SAP, Competitive Exam Preparation, Tourism.

Science and Agriculture

Life Sciences, Agriculture, Forestry, Pharmacy, Food Technology.

c. Industry Collaboration and Career Development

Anjanaas Foundation works with leading companies across India, ensuring that trained interns receive priority consideration in recruitment drives. Practical learning experiences prepare interns for real-world challenges, enhancing their employability and professional readiness.

CHAPTER 2

Executive Summary

This internship programme focused on the study and practical application of Business Analytics using Python. The training was conducted under the guidance of Anjanaas Foundation, where students were introduced to Python programming from the fundamentals through to data analytics libraries and forecasting techniques, culminating in a data-driven mini project.

The programme commenced with Python programming fundamentals, covering conditional statements, loops, functions, and modules. Object-Oriented Programming concepts including classes, objects, encapsulation, inheritance, abstraction, and polymorphism were then introduced, providing students with a structured programming foundation essential for building analytical applications.

The core analytics training progressed through two major Python libraries. NumPy was introduced for numerical computation, array operations, and matrix manipulation. The Pandas library was then studied for structured data management, covering DataFrames, data loading, filtering, grouping, merging, and data transformation operations essential for real-world data analysis.

The programme extended to Statistics, covering descriptive statistics, probability distributions, correlation, and statistical inference applied to business problems. Real-world applications of Business Analytics were explored through case studies in sales analysis, customer segmentation, and performance measurement.

The final phase introduced Exploratory Data Analysis, feature engineering techniques including lag features, rolling statistics, and cyclical encoding, followed by forecasting model implementation. Students implemented Linear Regression, Moving Average, and Weighted Moving Average models from scratch using NumPy, evaluated them using industry-standard metrics, and generated future forecasts.

The programme culminated in the development of a final mini project titled "Data-Driven Sales Forecasting Using Python," in which students designed and implemented a complete analytics pipeline covering data generation, EDA, feature engineering, model training, evaluation, and business insight extraction.

Sector of Development

This internship falls under the Business Analytics and Data Science sector, with a focus on Python programming, numerical computation, data manipulation, statistical analysis, and predictive modelling. The training covered multiple domains including Python fundamentals, object-oriented programming, data processing with NumPy and Pandas, statistical methods, and machine learning-based forecasting techniques.

Learning Objectives

- To understand Python programming fundamentals including control flow, functions, and modules.
- To learn object-oriented programming concepts and their application in analytical programs.
- To gain proficiency in numerical computation using the NumPy library.
- To develop data manipulation and analysis skills using the Pandas library.
- To understand statistical concepts and their application to business data problems.
- To explore real-world business analytics applications across industry sectors.
- To perform exploratory data analysis and engineer meaningful predictive features.
- To implement and evaluate forecasting models from scratch using NumPy.
- To apply all learned skills in the development of a complete data analytics mini project.

Outcomes Achieved

- Developed strong Python programming skills covering all core language constructs.
- Gained practical understanding of object-oriented programming principles.
- Learned to perform efficient numerical operations using NumPy arrays and matrices.
- Acquired proficiency in data manipulation, cleaning, and analysis using Pandas.
- Understood descriptive and inferential statistics as applied to business datasets.
- Explored real-world analytics use cases across sales, operations, and customer data.
- Implemented exploratory data analysis workflows and feature engineering pipelines.
- Successfully built and evaluated three forecasting models using NumPy from scratch.
- Designed and developed a complete data-driven sales forecasting project.
- Improved analytical thinking, structured programming habits, and technical documentation skills.

CHAPTER 3

Internship Part

Description of Activities and Responsibilities

The internship focused on building a thorough understanding of Python programming and its application to business data analysis and forecasting. Students progressed from Python language fundamentals through data processing libraries, statistical methods, and predictive modelling, concluding with the development of a complete data analytics project.

Activities carried out during the internship included writing Python programs for data manipulation and algorithmic problem solving, implementing numerical computations using NumPy, performing data analysis and transformation using Pandas, applying statistical techniques to business datasets, and developing forecasting models and a final analytical mini project.

Work Environment and Tools Used

The internship was conducted in an online learning environment organised by Anjanaas Foundation, where students participated in structured training sessions delivered through live Zoom classes and YouTube video lectures. Sessions combined theoretical explanation with live coding demonstrations.

Tool / Technology	Purpose
Python 3	Core programming language for all analytical tasks
Google Colab	Cloud-based Python runtime environment for all coding sessions
NumPy	Numerical computation, array operations, and matrix mathematics
Pandas	DataFrame-based data loading, cleaning, manipulation, and analysis
statistics (stdlib)	Descriptive statistics and distribution analysis
Zoom	Live interactive class sessions with faculty
YouTube	Recorded lecture playlist for revision and self-paced learning

Weekly Work Schedule Overview

The internship followed an eight-week structured training schedule designed to provide progressive learning from Python programming fundamentals to advanced data analytics and forecasting.

Weeks 1 – 2 — Foundations

Python programming fundamentals including conditional statements, loops, functions, modules, and object-oriented programming concepts covering classes, encapsulation, inheritance, abstraction, and polymorphism.

Weeks 3 – 6 — Core Analytics Training

Introduction to Business Analytics; NumPy for numerical computation; Pandas for data manipulation and analysis; Statistics with business applications; real-world analytics use cases; exploratory data analysis and feature engineering.

Weeks 7 – 8 — Forecasting and Project

Implementation of Linear Regression, Moving Average, and Weighted Moving Average forecasting models; model evaluation using R2, MAE, RMSE, and MAPE; development of the Data-Driven Sales Forecasting mini project; documentation and submission.

Tasks Performed

- Writing Python programs using conditional statements, loops, and iterative logic.
- Defining and calling functions, importing standard library modules, and organising code into reusable modules.
- Implementing object-oriented programming concepts including classes, objects, constructors, and inheritance.
- Performing numerical computations and array operations using NumPy.
- Loading, cleaning, filtering, grouping, and transforming structured datasets using Pandas DataFrames.
- Applying descriptive and inferential statistical methods to business data.
- Exploring real-world business analytics examples across multiple industry domains.
- Performing exploratory data analysis on synthetic sales datasets.
- Engineering lag features, rolling statistics, and cyclical time encodings for forecasting.
- Implementing OLS Linear Regression, Moving Average, and Weighted Moving Average models from scratch.
- Evaluating model performance using R2, MAE, RMSE, and MAPE metrics.
- Designing and developing the Data-Driven Sales Forecasting mini project.
- Preparing project documentation and internship logbook for final submission.

Skills Acquired

Python Programming

Proficiency in Python covering all core language constructs — conditionals, loops, functions, modules, and object-oriented design.

Numerical Computing

Ability to perform efficient array-based numerical operations and matrix computations using NumPy.

Data Analysis

Skill in loading, cleaning, transforming, and analysing structured business data using Pandas.

Statistical Analysis

Understanding of descriptive statistics, correlation, and statistical reasoning applied to business datasets.

Predictive Modelling

Experience in implementing regression and time-series forecasting models from first principles.

Business Analytics

Ability to extract actionable business insights from data and communicate findings clearly.

Reflection

The internship provided a comprehensive and structured learning experience in Python-based business analytics. The progression from Python language fundamentals through numerical computing and data manipulation to statistical analysis and predictive modelling created a coherent pathway that built both

programming proficiency and analytical thinking.

The hands-on coding approach, with every concept reinforced through live coding on Google Colab, ensured that students developed practical programming competency rather than theoretical understanding alone. Working through real data scenarios using Pandas and NumPy made abstract concepts tangible and directly applicable to professional data analysis tasks.

The development of the final Sales Forecasting project integrated all the skills built across the programme into a single end-to-end analytical pipeline. Overall, the internship significantly strengthened programming skills, data reasoning ability, and the professional discipline required for systematic data analysis work.

CHAPTER 4

Outcomes Description

Work Environment Experience

The internship was conducted in a structured online learning environment organised through the training programmes of Anjanaas Foundation. The environment emphasised disciplined learning, systematic training, and collaborative knowledge sharing among participants.

Students interacted regularly with faculty through live sessions conducted on Zoom and supplementary recorded lectures published on YouTube. This combination of live instruction and recorded content allowed students to attend interactive sessions while also revisiting complex programming demonstrations at their own pace.

All coding was carried out on Google Colab, a cloud-based Python environment that requires no local installation. This ensured that all students, regardless of their hardware capabilities, had equal access to a fully functional Python runtime with all required libraries pre-installed.

The structured weekly schedule maintained a professional learning culture that encouraged punctuality, consistent participation, and progressive skill development. Regular guidance from faculty ensured that students received timely support throughout the programme.

Real-Time Technical Skills Acquired

The internship helped students acquire several technical skills directly relevant to professional data analysis and business intelligence roles.

Students developed strong Python programming proficiency, including the ability to write structured programs using conditionals, loops, and functions. Object-oriented programming skills were developed by designing classes with encapsulation, implementing inheritance hierarchies, and applying abstraction and polymorphism in program design.

Practical experience in NumPy enabled students to perform efficient array-based computations, implement mathematical operations on datasets, and understand the vectorised computation model that underpins data science workflows.

The Pandas library training gave students the ability to work with real-world tabular data, perform complex group-by operations, merge datasets, apply transformation functions, and prepare data for downstream analysis and modelling.

The forecasting model implementation phase provided students with a clear understanding of how predictive models are built, trained, and evaluated from first principles — without relying on black-box library APIs. This reinforced conceptual understanding alongside practical competency.

Managerial Skills Acquired

The internship also developed several managerial and professional competencies. Students learned to plan their weekly tasks according to the structured programme schedule, improving time management and prioritisation skills.

The final project development required goal-oriented planning, systematic problem decomposition, and disciplined execution across data generation, EDA, feature engineering, modelling, evaluation, and documentation phases. This developed students' ability to manage a technical analytical project from start to

finish.

Students also developed analytical thinking and decision-making skills while choosing appropriate feature engineering strategies, selecting evaluation metrics, and interpreting model outputs to generate business recommendations.

Communication Skills Development

The internship contributed to the improvement of communication skills in both written and verbal forms. Students participated in live session discussions, asking questions about programming logic, data analysis approaches, and model interpretation. These interactions improved confidence in articulating technical reasoning.

Written communication skills were developed through the preparation of the project documentation and the internship logbook. Students learned to present technical concepts, code explanations, analytical findings, and business recommendations in a clear, structured, and professionally formatted manner.

Participation in Team Discussions and Collaborative Learning

Collaborative learning played an important role throughout the programme. Students engaged in group discussions during live sessions about programming approaches, data analysis strategies, and model evaluation interpretations. These discussions helped participants gain multiple perspectives on solving analytical problems and developed the ability to engage constructively with peers in a technical learning context.

Technological Developments Observed During the Internship

During the internship, students observed several important developments in the field of data analytics and Python-based business intelligence.

The increasing adoption of Python as the primary language for business analytics, data science, and machine learning was observed across industry and academic contexts. Python's open-source ecosystem — particularly NumPy, Pandas, and Matplotlib — has made data analysis accessible to a broad range of professionals beyond traditional software engineers.

Students also observed the growing importance of cloud-based development environments such as Google Colab, which have democratised access to computational resources and enabled collaborative coding without hardware constraints.

The increasing demand for data-driven decision-making across sales, operations, finance, and marketing functions was noted as a key driver of the business analytics skill set developed during the programme. Organisations of all sizes are increasingly relying on Python-based analytics workflows to replace manual reporting and spreadsheet-based analysis.

Conclusion

The Business Analytics Using Python internship provided valuable exposure to both the technical and applied dimensions of data analytics. Students progressed from foundational Python programming through data manipulation and statistical analysis to predictive modelling and business insight generation.

The hands-on coding approach using Google Colab, with every concept demonstrated through live examples, ensured that students developed genuine practical competency in Python-based analytics. The final project demonstrated how a complete analytical pipeline can be built from raw data generation through modelling to actionable business recommendations.

Overall, the internship strengthened technical programming skills, analytical reasoning, and professional work habits while providing practical insight into how data analytics is applied in real business environments. The skills developed during this programme provide a strong foundation for further study or career development in data science, business intelligence, and analytics-driven roles.

CHAPTER 5

Photos and Video Links

Photographs

Search for images and screenshots related to the topics covered in this internship programme — Python code outputs, data visualisations, Pandas DataFrame examples, statistical charts, NumPy array operations, and project results. Print them in colour, cut, and paste them in this section.

Video Links

All class sessions for this internship programme were recorded and uploaded to the Anjanaas Foundation YouTube channel as a playlist. The YouTube Course Playlist link is given below. Students are encouraged to use these recordings for revision.

YouTube Course Playlist :
https://youtube.com/playlist?list=PLx3CGw_4F7Gx9m8RJuvQwoOmwK-j7tY-b&si;=xUQZDayEJaac_8Ed

Zoom Meeting Link

Live training sessions for this programme were conducted through Zoom. The meeting link is provided below.

Zoom Meeting Link :
https://us06web.zoom.us/j/83744165103?pwd=BYPqdUynuq2BTyHTUM4ycUaNj5nMPN.1

CHAPTER 6

Activity Log

WEEK 1 — Python Fundamentals and Conditional Statements

Day	Brief Description of Activity	Learning Outcome	Person In-Charge
Day 1	Introduction to Business Analytics and the role of Python in data analysis	Understanding the relevance of Python for business analytics careers	Faculty
Day 2	Python syntax, variables, data types, and operators	Learning the basic building blocks of Python programming	Faculty
Day 3	Conditional statements: if, elif, and else for decision-making in programs	Writing programs that execute different logic based on conditions	Faculty
Day 4	Nested conditions and boolean expressions in analytical logic	Building complex decision structures for data filtering	Faculty
Day 5	Practical coding: writing conditional programs for data classification tasks	Applying conditional logic to real analytical scenarios	Faculty
Day 6	Program lab: debugging conditional programs and testing edge cases	Developing problem-solving and debugging skills	Faculty

Objective: To introduce Python programming fundamentals and conditional statements as the foundation for analytical program development.

Detailed Report

The first week of the internship commenced with an introduction to Business Analytics and the central role that Python plays in modern data analysis workflows. Students were introduced to the breadth of applications where Python is used in business contexts, including sales analysis, customer segmentation, demand forecasting, and operational performance monitoring. This established the applied purpose behind the technical skills that would be developed throughout the programme.

Python syntax and programming fundamentals were then introduced. Students learned about variables, primitive data types including integers, floats, strings, and booleans, and the arithmetic and comparison operators used to build computational logic. The interactive nature of Google Colab was demonstrated, allowing students to write and execute Python code directly in their web browsers without any local installation requirements.

Conditional statements — if, elif, and else — were studied as the primary mechanism for implementing decision-making logic in programs. Students learned how conditions are evaluated and how different code paths are executed based on the truthfulness of logical expressions. Nested conditions and compound boolean expressions were then covered to handle more complex decision scenarios.

Practical coding exercises applied conditional logic to analytical tasks such as classifying sales performance levels, determining discount eligibility, and categorising data records based on threshold values. The week concluded with a debugging session in which students identified and corrected logical errors in conditional programs, developing systematic problem-solving habits.

WEEK 2 — Loops, Functions, Modules and Object-Oriented Programming

Day	Brief Description of Activity	Learning Outcome	Person In-Charge
Day 1	For loops and while loops: iterating over data collections and sequences	Understanding repetition structures for processing multiple data records	Faculty
Day 2	Defining and calling functions: parameters, return values, and scope	Learning to build reusable code blocks for analytical tasks	Faculty
Day 3	Python modules: importing standard library modules and organising code	Understanding code organisation and the Python module system	Faculty
Day 4	Object-Oriented Programming: classes, objects, constructors, and encapsulation	Learning to model analytical entities as objects with data and behaviour	Faculty
Day 5	Inheritance, abstraction, and polymorphism in Python OOP	Understanding code reuse and flexible class design	Faculty
Day 6	Practical lab: building a class-based data processing program	Applying OOP principles to structure an analytical program	Faculty

Objective: To develop proficiency in Python loops, functions, modules, and object-oriented programming as tools for structured analytical program design.

Detailed Report

The second week extended Python programming skills into the constructs used to build larger, structured analytical programs. Loops were introduced as the mechanism for processing collections of data records iteratively. Both for loops, used to iterate over sequences such as lists and ranges, and while loops, used for condition-based repetition, were studied. Students practiced writing loop-based programs to calculate running totals, filter records meeting specified criteria, and accumulate results across multiple data points.

Functions were then introduced as the primary tool for organising code into reusable, named units. Students learned how to define functions with parameters, implement their logic, and return computed values. The concept of variable scope — the distinction between local and global variables — was explained in the context of analytical programs where different processing steps share data. Students wrote functions for common analytical tasks such as computing averages, applying business rules, and formatting data outputs.

Python modules were introduced as the means by which related functions and classes are grouped into importable units. Students explored the Python standard library, importing modules such as statistics, math, and datetime for use in analytical programs. The concept of creating custom modules was also discussed.

Object-Oriented Programming was then studied in depth. Students learned how classes define the structure and behaviour of objects, how constructors initialise object state, and how encapsulation protects internal data from unintended modification. Inheritance was demonstrated by building specialised analytical classes from base classes, and polymorphism was illustrated through methods that behave differently across different class types. The week concluded with a practical session building a class-based program structure.

WEEK 3 — Introduction to Business Analytics and NumPy

Day	Brief Description of Activity	Learning Outcome	Person In-Charge
Day 1	Introduction to Business Analytics: definition, scope, and industry applications	Understanding where and how business analytics creates value	Faculty
Day 2	Types of analytics: descriptive, diagnostic, predictive, and prescriptive	Learning the analytics maturity model and its business implications	Faculty
Day 3	Introduction to NumPy: arrays, array creation, and basic operations	Understanding NumPy as the numerical foundation of Python analytics	Faculty
Day 4	NumPy array indexing, slicing, and mathematical operations	Learning to extract and transform data within numerical arrays	Faculty
Day 5	Matrix operations and broadcasting in NumPy	Understanding vectorised computation and element-wise operations	Faculty
Day 6	Practical: implementing numerical data processing programs using NumPy	Applying NumPy to analytical computation tasks	Faculty

Objective: To introduce the scope and applications of Business Analytics and develop practical NumPy proficiency for numerical data processing.

Detailed Report

The third week opened with a conceptual introduction to Business Analytics, establishing the framework within which the technical skills developed throughout the programme are applied. Students studied the definition and scope of business analytics, exploring how organisations use data to understand past performance, diagnose the causes of outcomes, predict future results, and prescribe optimal courses of action. Real-world examples from sales, operations, marketing, and finance illustrated the breadth of analytics applications.

The four types of analytics — descriptive, diagnostic, predictive, and prescriptive — were studied as a maturity model. Students understood that most organisations begin with descriptive analytics (reporting what happened) and progress through diagnostic and predictive analytics as their data capabilities develop. This framework provided context for the forecasting techniques that would be introduced in later weeks.

The NumPy library was then introduced as the numerical computing foundation of the Python analytics ecosystem. Students learned how NumPy arrays differ from Python lists in terms of performance, memory efficiency, and supported operations. Array creation methods including np.array, np.zeros, np.ones, np.arange, and np.linspace were studied and practised.

Array indexing, slicing, and mathematical operations were then covered. Students learned to extract subsets of data from arrays, apply element-wise mathematical operations, and use NumPy's built-in aggregation functions such as sum, mean, std, min, and max. Matrix operations and the broadcasting mechanism that allows operations between arrays of different shapes were also introduced. Practical exercises applied these skills to numerical data processing scenarios.

WEEK 4 — Pandas Library and Data Manipulation

Day	Brief Description of Activity	Learning Outcome	Person In-Charge
Day 1	Introduction to Pandas: Series and DataFrame structures, creation, and inspection	Understanding Pandas as the primary tool for tabular data analysis	Faculty
Day 2	Data loading: reading CSV files, inspecting shape, columns, and data types	Learning to import and assess real-world structured datasets	Faculty
Day 3	Data selection, filtering, and conditional subsetting using Pandas	Extracting relevant subsets of data based on analytical criteria	Faculty
Day 4	GroupBy operations: aggregating and summarising data by category	Computing group-level statistics for business reporting	Faculty
Day 5	Data transformation: applying functions, handling missing values, and type conversion	Preparing and cleaning datasets for analysis	Faculty
Day 6	Practical with Pandas examples: analysis of sales and business datasets	Applying Pandas skills to end-to-end data analysis tasks	Faculty

Objective: To develop practical proficiency in the Pandas library for loading, cleaning, transforming, and analysing structured business data.

Detailed Report

The fourth week introduced the Pandas library, the primary tool in Python for working with structured tabular data. Students learned about the two fundamental Pandas data structures — Series, a one-dimensional labelled array, and DataFrame, a two-dimensional table with labelled rows and columns. The creation of DataFrames from dictionaries, lists, and existing NumPy arrays was demonstrated, and students practised inspecting DataFrames using head, tail, info, and describe methods.

Data loading was then covered, with students learning how to read structured data from CSV files using pd.read_csv and inspect the resulting DataFrame's shape, column names, data types, and missing value counts. The importance of understanding the structure and quality of a dataset before beginning analysis was emphasised as a foundational data science practice.

Data selection and filtering techniques were studied in depth. Students learned to select individual columns and subsets of columns, filter rows using boolean conditions, and combine multiple conditions with logical operators. These operations form the backbone of data exploration and preparation tasks in professional analytics workflows.

GroupBy operations were introduced as the primary mechanism for computing group-level summaries. Students learned to group DataFrames by categorical columns and apply aggregation functions such as sum, mean, count, and min/max to each group. These operations directly enable the kind of sales-by-category and performance-by-region analysis used in the final project. The week concluded with practical data analysis exercises on business datasets.

WEEK 5 — Statistics and Real-World Business Analytics Applications

Day	Brief Description of Activity	Learning Outcome	Person In-Charge
Day 1	Descriptive statistics: mean, median, mode, variance, and standard deviation	Understanding the statistical summary of business data distributions	Faculty
Day 2	Correlation analysis: Pearson correlation and its business interpretation	Learning to identify relationships between business variables	Faculty
Day 3	Probability distributions and their role in business decision-making	Understanding statistical foundations of predictive reasoning	Faculty
Day 4	Real-world applications of Business Analytics: sales forecasting, customer analysis	Exploring how analytics is applied across business functions	Faculty
Day 5	Case study: applying statistical analysis to a business dataset using Pandas and NumPy	Practising end-to-end statistical analysis on real scenarios	Faculty
Day 6	Practical lab: computing statistics and interpreting results in business context	Developing skill in statistical interpretation and insight communication	Faculty

Objective: To develop statistical reasoning skills and explore real-world business analytics applications using Python.

Detailed Report

The fifth week introduced statistical methods applied to business data. Students studied the core descriptive statistics — mean, median, mode, variance, and standard deviation — and learned how each measure characterises a different aspect of a dataset's distribution. The distinction between mean and median as measures of central tendency was discussed in the context of skewed business data such as sales figures and customer transaction values.

Correlation analysis was introduced as the primary statistical tool for identifying relationships between business variables. Students learned to compute Pearson correlation coefficients using Pandas and NumPy and to interpret the strength and direction of correlations in business terms. The distinction between correlation and causation was emphasised as a critical concept for responsible analytical reasoning.

Probability distributions were introduced to provide the statistical foundation for predictive modelling. The normal distribution, its properties, and its prevalence in business data were discussed, along with an introduction to how probability underpins statistical forecasting and risk assessment.

Real-world business analytics applications were then explored through examples from sales forecasting, customer lifetime value analysis, inventory optimisation, and employee performance analytics. These case studies demonstrated how the Python skills developed during the programme translate directly into professional analytical deliverables. The week concluded with a practical session in which students performed a complete statistical analysis on a sample business dataset, computing and interpreting all descriptive statistics and correlation measures.

WEEK 6 — Exploratory Data Analysis and Feature Engineering

Day	Brief Description of Activity	Learning Outcome	Person In-Charge
Day 1	Exploratory Data Analysis: data overview, missing value detection, and distribution analysis	Learning the systematic process of understanding a new dataset	Faculty
Day 2	Categorical and temporal data analysis: grouping by category and time period	Performing multi-dimensional data breakdowns for business insight	Faculty
Day 3	Lag features: creating time-shifted variables for forecasting	Understanding how past values are encoded as predictive features	Faculty
Day 4	Rolling statistics: computing windowed means and standard deviations without data leakage	Learning to summarise recent trends as model-ready features	Faculty
Day 5	Cyclical time encoding: sine and cosine transformation of month numbers	Encoding calendar seasonality correctly for linear models	Faculty
Day 6	Practical: building a complete feature engineering pipeline in Python	Assembling all feature engineering steps into a structured class	Faculty

Objective: To perform systematic exploratory data analysis and engineer predictive features including lag variables, rolling statistics, and cyclical time encodings.

Detailed Report

The sixth week introduced Exploratory Data Analysis as the systematic process of understanding a dataset before modelling. Students learned to compute and interpret dataset overviews including total records, data types, date ranges, unique category values, and missing value counts. Summary statistics were computed for the sales dataset and ASCII bar charts were used to visualise monthly revenue trends without requiring external plotting libraries.

Categorical and temporal data breakdowns were then performed using Pandas GroupBy operations. Students computed total sales, average sales, and market share by product category, and regional sales rankings by geographic region. These analyses replicated the type of business reporting performed by analytics teams in retail and consumer goods organisations.

Feature engineering was introduced as the process of creating new input variables from raw data to improve the predictive power of models. Lag features were the first engineering technique studied: by shifting the sales column by one, two, and three time steps within each category-region group, students created features that encode what sales were in previous months — the strongest predictors in many time-series forecasting problems.

Rolling statistics were then introduced, computing windowed means and standard deviations over windows of three and six months. Students learned the critical importance of applying shift(1) before the rolling window to ensure that the current period's target value is never included in the feature — preventing data leakage. Cyclical time encoding using sine and cosine transformations of the month number was studied as the correct way to represent calendar seasonality in linear models, ensuring that December and January are represented as adjacent rather than maximally distant.

WEEK 7 — Forecasting Models and Model Evaluation

Day	Brief Description of Activity	Learning Outcome	Person In-Charge
Day 1	OLS Linear Regression from scratch: normal equation, NumPy implementation, feature matrix	Understanding and implementing regression without external ML libraries	Faculty
Day 2	Moving Average model: causal windowed prediction and iterative future forecasting	Learning simple baseline time-series forecasting	Faculty
Day 3	Weighted Moving Average (EWMA): exponential smoothing and correct forecast propagation	Understanding exponential weighting for recent-history-sensitive forecasting	Faculty
Day 4	Train-test split: chronological splitting for time-series integrity	Learning why shuffled splits are inappropriate for time-series data	Faculty
Day 5	Model evaluation metrics: R2, MAE, RMSE, and MAPE — computation and interpretation	Measuring and comparing model performance with standard analytics metrics	Faculty
Day 6	Practical: training all three models, comparing metrics, and generating 3-month forecasts	Executing a complete model training, evaluation, and forecasting workflow	Faculty

Objective: To implement three forecasting models from scratch using NumPy, evaluate them with standard metrics, and generate future sales forecasts.

Detailed Report

The seventh week introduced predictive modelling for business analytics, focusing on implementing three forecasting approaches from first principles using NumPy — without relying on external machine learning libraries. This design reinforced conceptual understanding of how models work rather than treating them as black-box tools.

Ordinary Least Squares Linear Regression was the first model implemented. Students derived the normal equation solution and implemented it using NumPy's np.linalg.lstsq function, which solves the system numerically. The feature matrix constructed from lag features, rolling statistics, and cyclical time encodings was used to train the model on the chronological training split. Students learned why Units_Sold and Avg_Price must be excluded from the feature set — as they are algebraically derived from the target variable and would allow the model to reconstruct sales trivially.

The Moving Average and Exponentially Weighted Moving Average models were then implemented on the aggregate monthly sales series using a chronological, month-boundary train-test split. The EWMA forecast propagation was implemented correctly, updating the anchor value at each step to ensure forecasts propagate forward accurately rather than anchoring to the last known observation.

Model evaluation used four metrics — R-squared, MAE, RMSE, and MAPE — computed on the held-out test set. Linear regression achieved an R2 of 0.29, reflecting genuine predictive skill from the engineered features. The three-model comparison gave students direct practice in reading evaluation outputs as the basis for model selection decisions.

WEEK 8 — Final Project, Documentation and Submission

Day	Brief Description of Activity	Learning Outcome	Person In-Charge
Day 1	Final project overview: Sales Forecasting system architecture and module planning	Understanding the complete project scope and how all skills combine	Faculty
Day 2	Data generation module: SalesDataGenerator class with seasonality, trend, and regional variation	Implementing realistic synthetic multi-dimensional sales data	Faculty
Day 3	EDA and Feature Engineering modules: SalesAnalyzer and FeatureEngineer class implementation	Building the analytical and feature preparation pipeline	Faculty
Day 4	Model and Evaluation modules: implementing and comparing all three forecasting models	Completing the modelling and evaluation pipeline	Faculty
Day 5	Forecasting and Business Insights modules: 3-month forecast and insight extraction	Generating actionable outputs from the analytical pipeline	Faculty
Day 6	Project documentation, logbook completion, final review, and submission	Finalising all internship deliverables and completing the programme	Faculty

Objective: To design and implement the Data-Driven Sales Forecasting mini project integrating all skills from the programme, and to complete all documentation for final submission.

Detailed Report

The eighth and final week was dedicated to the development of the mini project titled Data-Driven Sales Forecasting Using Python. The project architecture was planned first, mapping each of the programme's skill areas — data generation, EDA, feature engineering, modelling, and insight extraction — to a corresponding class in the project codebase. This planning phase reinforced object-oriented design principles by translating analytical requirements into class structures.

The SalesDataGenerator class was implemented to produce two years of synthetic monthly sales data across three product categories and four geographic regions. The data incorporated seasonal variation specific to each category, regional sales multipliers, a 0.5% monthly growth trend, and Gaussian random noise. Students verified that the generated dataset exhibited the expected patterns through EDA before proceeding to feature engineering.

The SalesAnalyzer and FeatureEngineer classes were then implemented. The analyser produced dataset overviews, summary statistics, category and region performance tables, monthly trend bar charts, and a correlation matrix. The feature engineer added lag features, leakage-free rolling statistics using shift(1) before rolling windows, cyclical month encodings, and label-encoded categorical variables.

The ModelEvaluator class trained all three models, applied the correct evaluation methodology, and printed the comparison table. The SalesForecaster generated three-month forward predictions using the trained linear regression model with correct lag propagation across forecast steps. The BusinessInsights class extracted the top category, region, peak month, year-on-year growth, and highest average price, and generated four strategic recommendations.

The week concluded with the preparation of the project report and the completion of the internship logbook. Students reviewed all programme content, finalised their documentation, and submitted all internship deliverables. The programme concluded with a review session reflecting on the full learning journey from Python fundamentals through analytics to the development of a complete data-driven forecasting application.