

50 Days of Data Analytics Journey

Project Introduction

As a graduate of the HyperionDev data science bootcamp, I come into this 50-day data analytics journey equipped with foundational skills in Python and data analysis. However, I recognise the importance of continually honing and expanding my knowledge to stay competitive in the rapidly evolving field of data science. This project serves as an opportunity for me to further cement my existing skills and knowledge while revisiting and reinforcing techniques previously learned. By completing 50 data analytic tasks over the course of this journey, I aim to deepen my understanding of Python programming, data manipulation, and statistical analysis techniques. Through meticulous organisation and documentation of each task, I intend to create a comprehensive timeline and development map that not only showcases my progress but also highlights areas for further growth. Additionally, I aspire to enhance my skills in data visualisation and dashboard creation, ultimately becoming a more proficient and confident data analyst capable of tackling complex real-world data challenges with ease.

Day 1: Data Loading and Basic Exploration

- **Task:** Load a dataset into a Pandas DataFrame and perform basic exploration.
- **Description:** Use Pandas' `read_csv()` function to load a dataset into a DataFrame. Display the first few rows of the dataset using the `head()` method to understand its structure and format. Check the data types of each column using the `info()` method to identify any inconsistencies or missing values. Calculate summary statistics for numerical columns using the `describe()` method to gain insights into the data distribution. Identify missing values in the dataset using methods like `isna()` and visualise them if necessary to understand

Day 2: Exploratory Data Analysis - Feature Correlation

- **Task:** Analyse feature correlations within the dataset.
- **Description:** Calculate correlation coefficients between numerical features and visualise the correlation matrix using a heatmap. Identify highly correlated features and consider their impact on the analysis.

Day 3: Data Preprocessing - Feature Encoding

- **Task:** Encode categorical variables into numerical format.
- **Description:** Use techniques like one-hot encoding or label encoding to convert categorical variables into a format suitable for machine learning models. Consider the impact of encoding schemes on model performance and interpretability.

Day 4: Data Visualisation - Univariate Analysis

- **Task:** Visualise the distribution of a single variable in the dataset.
- **Description:** Create histograms, box plots, or kernel density plots to understand the distribution of individual variables.

Day 5: Data Visualisation - Bivariate Analysis

- **Task:** Explore relationships between pairs of variables in the dataset.
- **Description:** Create scatter plots, pair plots, or violin plots to visualise relationships between two variables.

Day 6: Data Preprocessing - Scaling and Normalisation

- **Task:** Scale and normalise numerical features in the dataset.
- **Description:** Use techniques like Min-Max scaling or Z-score normalisation to scale numerical features to a similar range.

Day 7: Data Preprocessing - Encoding Categorical Variables

- **Task:** Encode categorical variables into numerical format.
- **Description:** Use techniques like one-hot encoding or label encoding to convert categorical variables into a format suitable for machine learning models.

Day 8: Feature Engineering - Creating New Features

- **Task:** Create new features from existing ones in the dataset.
- **Description:** Generate new features based on domain knowledge or by combining existing features to improve model performance.

Day 9: Feature Engineering - Handling Date and Time Data

- **Task:** Extract relevant information from date and time columns.
- **Description:** Use Pandas' date-time functions to extract features such as day of the week, month, or year from date-time columns.

Day 10: Correlation Analysis

- **Task:** Explore correlations between pairs of numerical variables.
- **Description:** Calculate correlation coefficients and visualise correlations using heat map or pair plots.

Day 11: Outlier Detection and Removal

- **Task:** Identify and remove outliers from the dataset.
- **Description:** Use statistical methods or visualisation techniques to detect outliers and remove them from the dataset.

Day 12: Data Sampling Techniques

- **Task:** Explore different data sampling techniques.
- **Description:** Learn about techniques like random sampling, stratified sampling, and under-sampling/oversampling for imbalanced datasets.

Day 13: Dimensionality Reduction - Principal Component Analysis (PCA)

- **Task:** Apply PCA to reduce the dimensionality of the dataset.
- **Description:** Use PCA to transform high-dimensional data into a lower-dimensional space while preserving important information.

Day 14: Dimensionality Reduction - t-Distributed Stochastic Neighbour Embedding (t-SNE)

- **Task:** Apply t-SNE for visualisation of high-dimensional data.
- **Description:** Use t-SNE to visualise high-dimensional data in a two- or three-dimensional space while preserving local structures.

Day 15: Introduction to Machine Learning Algorithms

- **Task:** Explore basic machine learning algorithms.
- **Description:** Learn about algorithms like linear regression, logistic regression, decision trees, and k-nearest neighbours.

Day 16: Supervised Learning - Linear Regression

- **Task:** Implement linear regression for predicting continuous outcomes.
- **Description:** Use Scikit-learn to build a linear regression model and evaluate its performance using metrics like mean squared error.

Day 17: Supervised Learning - Logistic Regression

- **Task:** Implement logistic regression for binary classification tasks.
- **Description:** Use logistic regression to predict binary outcomes and evaluate the model's performance using metrics like accuracy and confusion matrix.

Day 18: Supervised Learning - Decision Trees and Random Forests

- **Task:** Implement decision trees and random forests for classification and regression.
- **Description:** Use decision trees and random forests to build predictive models and understand the concept of ensemble learning.

Day 19: Supervised Learning - Support Vector Machines (SVM)

- **Task:** Implement SVM for classification and regression tasks.
- **Description:** Learn about SVM and its kernel methods for handling non-linear data and build SVM models for classification and regression.

Day 20: Supervised Learning - Naive Bayes Classifier

- **Task:** Implement Naive Bayes classifier for classification tasks.
- **Description:** Understand the Bayes theorem and assumptions of Naive Bayes classifier and build models for text classification.

Day 21: Model Evaluation Metrics

- **Task:** Learn about different model evaluation metrics.
- **Description:** Explore metrics like accuracy, precision, recall, F1-score, ROC-AUC, and learn when to use them based on the nature of the problem.

Day 22: Cross-Validation Techniques

- **Task:** Implement cross-validation for model evaluation.
- **Description:** Learn about k-fold cross-validation, stratified k-fold cross-validation, and leave-one-out cross-validation techniques for robust model evaluation.

Day 23: Hyperparameter Tuning - Grid Search and Random Search

- **Task:** Perform hyperparameter tuning using grid search and random search.
- **Description:** Optimize model performance by tuning hyperparameters using exhaustive grid search and randomized search techniques.

Day 24: Ensemble Learning - Bagging and Boosting

- **Task:** Implement bagging and boosting ensemble methods.
- **Description:** Learn about bagging (e.g., Random Forest) and boosting (e.g., AdaBoost, Gradient Boosting) techniques to improve model performance.

Day 25: Medium Complexity Capstone Project - Customer Segmentation

- **Project Description:** Perform customer segmentation analysis on a retail dataset to identify different customer segments based on their purchasing behavior.
- **Tasks:**
 1. Load and preprocess the retail dataset, including handling missing values and scaling numerical features.
 2. Apply K-Means clustering algorithm to segment customers into distinct groups.
 3. Analyze and interpret the characteristics of each customer segment.
 4. Visualize the results using interactive dashboards using tools like Plotly or Dash.
- **Report/Learning Material:** Provide a detailed report discussing the insights gained from customer segmentation analysis, including the char-

acteristics of each segment and actionable recommendations for marketing strategies.

Day 26: Midway Reflection

- **Reflection:**

- Reflect on the progress made so far in the data analytics project.
- Evaluate the skills and techniques learned during the first half of the project.
- Identify challenges faced and lessons learned.
- Set goals and priorities for the remaining days of the project.
- Consider areas for improvement and additional learning opportunities.

Day 27: Association Rule Learning - Apriori Algorithm

- **Project Description:** Implement the Apriori algorithm for association rule learning.
- **Tasks:**
 1. Understand the principles of association rule learning and the Apriori algorithm.
 2. Preprocess transactional data and convert it into a suitable format for the Apriori algorithm.

3. Implement the Apriori algorithm to discover frequent itemsets and association rules.
 4. Evaluate and interpret the discovered rules to extract meaningful insights.
- **Report/Learning Material:** Provide a report detailing the implementation of the Apriori algorithm, including the discovered association rules and their significance in the context of the dataset.

Day 28: Introduction to Natural Language Processing (NLP)

- **Project Description:** Learn the basics of NLP and text preprocessing techniques.
- **Tasks:**
 1. Understand the fundamental concepts of natural language processing (NLP).
 2. Explore common text preprocessing techniques such as tokenization, stemming, and lemmatisation.
 3. Apply text preprocessing techniques to prepare text data for further analysis.
- **Report/Learning Material:** Provide a tutorial or guide on text preprocessing techniques in NLP, along with code examples and resources for further learning.

Day 29: Text Classification - Sentiment Analysis

- **Project Description:** Implement sentiment analysis for text classification.
- **Tasks:**
 1. Understand the concept of sentiment analysis and its applications.
 2. Preprocess text data and prepare it for sentiment analysis.
 3. Implement a sentiment analysis model using machine learning or deep learning techniques.
 4. Evaluate the performance of the sentiment analysis model and interpret the results.
- **Report/Learning Material:** Present a report discussing the implementation of sentiment analysis, including model performance metrics and insights gained from the analysis.

Day 30: Text Classification - Named Entity Recognition (NER)

- **Project Description:** Implement NER for extracting entities from text data.
- **Tasks:**
 1. Understand the concept of Named Entity Recognition (NER) and its applications.

2. Preprocess text data and prepare it for NER.
 3. Implement an NER model using machine learning or deep learning techniques.
 4. Evaluate the performance of the NER model and analyze the extracted entities.
- **Report/Learning Material:** Provide a report detailing the implementation of NER, including the performance of the model and examples of extracted entities.

Day 31: Text Classification - Document Classification

- **Project Description:** Implement document classification for categorizing text documents into predefined classes.
- **Tasks:**
 1. Understand the concept of document classification and its applications.
 2. Preprocess text data and convert it into a suitable format for document classification.
 3. Implement a document classification model using machine learning or deep learning algorithms.
 4. Evaluate the performance of the document classification model using appropriate metrics.

- **Report/Learning Material:** Present a report discussing the implementation of document classification, including model performance evaluation and insights gained from the classification process.

Day 32: Time Series Analysis - Decomposition and Forecasting

- **Project Description:** Perform time series analysis to decompose time series data into trend, seasonality, and residual components, and make forecasts.
- **Tasks:**
 1. Understand the components of time series data and their characteristics.
 2. Decompose time series data using methods like additive or multiplicative decomposition.
 3. Apply forecasting techniques such as ARIMA (AutoRegressive Integrated Moving Average) or Exponential Smoothing to make predictions.
 4. Evaluate the accuracy of the forecasts and analyse the results.
- **Report/Learning Material:** Provide a report detailing the time series analysis process, including decomposition, forecasting methods used, and insights gained from the forecasted values.

Day 33: Time Series Analysis - Seasonal Decomposition with LOESS (STL)

- **Project Description:** Perform seasonal decomposition using LOESS (Locally Estimated Scatterplot Smoothing) to decompose time series data into trend, seasonality, and residual components.
- **Tasks:**
 1. Understand the STL decomposition method and its advantages in handling non-linear trends and seasonality.
 2. Decompose time series data using the STL decomposition algorithm.
 3. Analyze the decomposed components and interpret the results.
- **Report/Learning Material:** Provide a report discussing the implementation of STL decomposition, including insights gained from the decomposition process and comparisons with other decomposition methods.

Day 34: Introduction to Geospatial Analysis

- **Project Description:** Learn the basics of geospatial analysis and visualization.
- **Tasks:**
 1. Understand the concepts of geospatial data and its applications.

2. Explore common geospatial datasets and formats such as shapefiles, GeoJSON, and raster data.
 3. Visualize geospatial data using libraries like GeoPandas, Matplotlib, and Folium.
- **Report/Learning Material:** Provide a tutorial or guide on geospatial analysis and visualization techniques, including code examples and resources for further learning.

Day 35: Geospatial Data Visualisation - Choropleth Maps

- **Project Description:** Create choropleth maps to visualise spatial distributions of data.
- **Tasks:**
 1. Understand the concept of choropleth maps and their use cases.
 2. Preprocess geospatial data and join it with attribute data.
 3. Create choropleth maps using libraries like GeoPandas, Matplotlib, or Plotly.
 4. Customize the maps with color schemes, legends, and annotations.
- **Report/Learning Material:** Provide a report detailing the process of creating choropleth maps, including insights gained from visualizing spatial distributions of data.

Day 36: Geospatial Data Visualization - Interactive Maps with Folium

- **Project Description:** Create interactive maps using the Folium library.
- **Tasks:**
 1. Understand the capabilities of the Folium library for creating interactive maps.
 2. Load geospatial data and overlay it on the map.
 3. Customize map elements such as markers, popups, and tooltips.
 4. Add interactive features such as zooming, panning, and layer control.
- **Report/Learning Material:** Provide a tutorial or guide on creating interactive maps with Folium, including code examples and resources for further exploration.

Day 37: Spatial Analysis - Spatial Joins and Buffers

- **Project Description:** Perform spatial analysis operations such as spatial joins and buffers.
- **Tasks:**
 1. Understand the concepts of spatial joins and buffers in geospatial analysis.

2. Perform spatial joins to combine geospatial datasets based on spatial relationships.
 3. Create buffers around spatial features to analyze spatial proximity or containment.
 4. Visualize the results of spatial analysis operations.
- **Report/Learning Material:** Provide a report discussing the implementation of spatial joins and buffers, including examples of real-world applications and insights gained from the analysis.

Day 38: Spatial Analysis - Point Pattern Analysis

- **Project Description:** Analyze the spatial distribution of points using point pattern analysis techniques.
- **Tasks:**
 1. Understand the concepts of point pattern analysis and its applications.
 2. Calculate spatial statistics such as point density, nearest neighbor distances, and spatial autocorrelation.
 3. Visualize point patterns and spatial statistics to identify spatial patterns and clusters.
 4. Interpret the results of point pattern analysis and draw conclusions.

- **Report/Learning Material:** Provide a report detailing the process of point pattern analysis, including the analysis of spatial statistics and visualization of point patterns.

Day 39: Spatial Analysis - Network Analysis

- **Project Description:** Perform network analysis on spatial data to analyze connectivity and accessibility.
- **Tasks:**
 1. Understand the concepts of network analysis and its applications in transportation, urban planning, and logistics.
 2. Build network datasets from road networks, public transportation networks, or other spatial networks.
 3. Perform network analysis operations such as routing, shortest path analysis, and network centrality measures.
 4. Visualize network analysis results and interpret the findings.
- **Report/Learning Material:** Provide a report discussing the implementation of network analysis, including examples of real-world applications and insights gained from the analysis.

Day 40: Introduction to Dashboards

- **Project Description:** Learn the basics of creating interactive dashboards for data visualization.
- **Tasks:**
 1. Understand the importance of dashboards in data analysis and communication.
 2. Explore different dashboarding tools and libraries such as Plotly Dash, Tableau, or Power BI.
 3. Design a simple dashboard layout with interactive components such as charts, filters, and sliders.
- **Report/Learning Material:** Provide a tutorial or guide on creating interactive dashboards, including examples of dashboard layouts and components.

Day 41: Dashboard Design Principles

- **Project Description:** Learn about best practices and design principles for creating effective dashboards.
- **Tasks:**
 1. Understand the principles of effective data visualization and dashboard design.
 2. Explore concepts such as data-ink ratio, chart selection, color theory, and layout design.
 3. Apply design principles to improve the clarity, readability, and usability of dashboards.

- **Report/Learning Material:** Provide a guide on dashboard design principles, including examples of good and bad dashboard designs.

Day 42: Dashboard Implementation with Plotly Dash

- **Project Description:** Implement an interactive dashboard using the Plotly Dash library.
- **Tasks:**
 1. Set up a basic web application with Plotly Dash.
 2. Design and implement interactive components such as drop-downs, sliders, and buttons.
 3. Integrate Plotly charts and visualizations into the dashboard layout.
- **Report/Learning Material:** Provide a tutorial or guide on building dashboards with Plotly Dash, including code examples and resources for further learning.

Day 43: Dashboard Implementation with Streamlit

- **Project Description:** Implement an interactive dashboard using the Streamlit library.
- **Tasks:**

1. Set up a Streamlit application and create a basic dashboard layout.
 2. Add interactive widgets and components such as sliders, text inputs, and select boxes.
 3. Integrate data visualizations and plots into the dashboard using Matplotlib, Plotly, or other libraries.
- **Report/Learning Material:** Provide a tutorial or guide on building dashboards with Streamlit, including code examples and resources for further exploration.

Day 44: Dashboard Implementation with Power BI

- **Project Description:** Implement an interactive dashboard using Microsoft Power BI.
- **Tasks:**
 1. Connect Power BI to a dataset and import data for analysis.
 2. Design and customize dashboard visuals such as charts, tables, and maps.
 3. Create interactive elements such as slicers, filters, and drill-through actions.
- **Report/Learning Material:** Provide a tutorial or guide on building dashboards with Power BI, including examples of dashboard creation and customization.

Day 45: Advanced Dashboard Techniques

- **Project Description:** Explore advanced techniques for building interactive dashboards.
- **Tasks:**
 1. Implement dynamic data filtering and sorting in dashboards.
 2. Add interactivity with hover effects, tooltips, and click events.
 3. Embed external content such as videos, images, and web pages into dashboards.
- **Report/Learning Material:** Provide a tutorial or guide on advanced dashboard techniques, including code examples and resources for further learning.

Day 46: Dashboard Deployment and Hosting

- **Project Description:** Deploy and host interactive dashboards for sharing and collaboration.
- **Tasks:**
 1. Choose a hosting platform for deploying dashboards such as Heroku, AWS, or Microsoft Azure.
 2. Deploy the dashboard application to the chosen hosting platform.

3. Set up security and access controls for sharing the dashboard with others.
- **Report/Learning Material:** Provide a tutorial or guide on dashboard deployment and hosting, including step-by-step instructions and best practices.

Day 47: Dashboard Optimization and Performance Tuning

- **Project Description:** Optimize dashboard performance for speed and efficiency.
- **Tasks:**
 1. Identify potential bottlenecks and performance issues in dashboards.
 2. Implement optimization techniques such as data caching, lazy loading, and asynchronous updates.
 3. Test and measure the performance improvements achieved.
- **Report/Learning Material:** Provide a guide on dashboard optimization techniques, including strategies for improving performance and reducing load times.

Day 48: Dashboard Accessibility and Usability

- **Project Description:** Ensure accessibility and usability of dashboards for all users.
- **Tasks:**
 1. Implement accessibility features such as keyboard navigation and screen reader compatibility.
 2. Optimize dashboard layout and design for usability on different devices and screen sizes.
 3. Conduct usability testing with a diverse group of users to gather feedback and identify areas for improvement.
- **Report/Learning Material:** Provide a guide on dashboard accessibility and usability best practices, including tips for designing inclusive dashboards.

Day 49: Dashboard Maintenance and Updates

- **Project Description:** Establish a maintenance plan for keeping dashboards up-to-date and relevant.
- **Tasks:**
 1. Monitor dashboard usage and performance regularly.
 2. Address bugs, issues, and user feedback promptly.

3. Update dashboards with new data, features, or visualizations as needed.
- **Report/Learning Material:** Provide a guide on dashboard maintenance and update procedures, including strategies for ensuring the long-term success of dashboard projects.

Day 50: Difficult Capstone Project - Predictive Maintenance for Industrial Equipment

- **Project Description:** Build a predictive maintenance model to anticipate equipment failures and schedule maintenance proactively, based on sensor data collected from industrial machinery.
- **Tasks:**
 1. Preprocess the sensor data, including handling time series data, feature engineering, and scaling.
 2. Implement advanced machine learning algorithms such as LSTM (Long Short-Term Memory) or CNN (Convolutional Neural Network) for predictive modeling.
 3. Evaluate the performance of the predictive maintenance model using appropriate metrics such as precision, recall, and F1-score.
 4. Develop a web-based dashboard to visualize real-time equipment health status and maintenance schedules.
- **Report/Learning Material:** Present a comprehensive report detailing the predictive maintenance model's development, evaluation results,

and the potential impact on minimizing downtime and maximizing operational efficiency in industrial settings. Provide learning materials such as code snippets, tutorials, and resources used for implementing advanced machine learning techniques and developing the dashboard.