A

Practical Training II Report

on

Sales Forecasting using Machine Learning and Data Analytics

at

Navikaran Infotech Private Limited

Duration: 20th June 2023 – 30th July 2023

Submitted By:	**Submitted To:**
Aashi Goyal	Dr. Ritu Boora
200151520051	Assistant Professor
B.Tech. ECE (7th semester)	Deptt. of EEE, GJUS&T, Hisar

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

GURU JAMBHESHWAR UNIVERSITY OF SCIENCE AND TECHNOLOGY, HISAR (125001)

CERTIFICATE

NAVIKARAN
INFOTECH PVT. LTD.

9th August 2023

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Ms. Aashi Goyal** pursuing B.Tech, from Guru Jambheshwar University of Science and Technology, was a trainee with **NAVIKARAN INFOTECH PRIVATE LIMITED**, New Delhi. During her training period, she was associated with the Data Analytics team and working on **Forecasting of Retail Sales using Machine Learning and Data Analytics** from 20th June 2023 – 30th July 2023.

During the training period, her performance was found **to be Good** and we wish her success in all her future endeavors.

For **NAVIKARAN INFOTECH PRIVATE LIMITED**

Ankit Yadav
Manager – HR & Admin

NaviKaran Infotech Pvt. Ltd. |2nd Floor, Plot No 4, South Ganesh Nagar, New Delhi -110092
Contact: +91-9560779056
Email: contact@navikaran.com
Website: www.navikaran.com

ACKNOWLEDGMENT

I am deeply grateful to Mr. Ankit Yadav, the HR Head of NAVIKARAN INFOTECH PRIVATE LIMITED, for offering me the invaluable opportunity to intern within the organization. This opportunity has been an enriching experience that has contributed significantly to my personal and professional growth.

I thank the hardworking people who assisted me at NAVIKARAN INFOTECH PRIVATE LIMITED. Their consistent support, encouragement, and openness helped to create a highly supportive and happy workplace. I owe them all a debt of gratitude.

I greatly appreciate Miss Aanchal Aggarwal for her constant support and for giving me the tools I needed to complete my internship successfully. Their dedication to providing a learning environment and developing potential is genuinely admirable.

The support and guidance I received from the College internship coordinator, Department of ECE, played a pivotal role in securing and completing my internship at NAVIKARAN INFOTECH PRIVATE LIMITED. I extend my gratitude to them for their valuable advice and unwavering assistance.

I sincerely appreciate the collective efforts of my department's staff members and the encouragement I received from my friends. Their support was instrumental in ensuring the successful culmination of my internship journey.

In conclusion, I am humbled and profoundly thankful to everyone who has been a part of this journey. Your guidance, support, and encouragement have been instrumental in shaping my professional path, and I look forward to carrying forward the knowledge and experience gained during this internship into my future endeavors.

Thank you all for your invaluable contributions to my growth and development.

Aashi Goyal

TRAINING OBJECTIVES

Hands-on Experience: Gain practical hands-on experience in data analytics and machine learning, applying theoretical knowledge to real-world scenarios.

Data Handling Proficiency: Develop skills in data collection, preprocessing, and cleaning, ensuring the ability to work with raw data to derive meaningful insights.

Model Development: Learn how to build predictive models for regression and classification tasks, focusing on understanding model selection and evaluation.

Feature Engineering Mastery: Acquire expertise in feature engineering techniques to extract relevant information from datasets, enhancing model performance.

Data Visualization Skills: Develop proficiency in data visualization tools and techniques to communicate insights and findings from data analysis effectively.

Interdisciplinary Collaboration: Collaborate with professionals from diverse backgrounds within NAVIKARAN INFOTECH PRIVATE LIMITED, gaining exposure to cross-functional teamwork.

Problem-Solving: Cultivate problem-solving skills by tackling real-world challenges and addressing issues that arise during data analysis and modeling.

Professional Networking: Build a network of contacts within the organization, learning from experienced professionals and gaining insights into industry best practices.

Project Management: Gain experience in managing projects, including setting goals, timelines, and priorities, to ensure the successful completion of assigned tasks.

Documentation and Reporting: Hone the ability to document work processes, findings, and solutions in a clear and organized manner, culminating in a comprehensive internship report.

These objectives collectively aim to provide you with a well-rounded and practical experience during your internship at NAVIKARAN INFOTECH PRIVATE LIMITED, equipping you with valuable skills and insights for your future career in data analytics.

TABLE OF CONTENTS

COMPANY PROFILE

Navikaran Infotech Pvt. Ltd. is a software development company based in Delhi, India. They were founded in 2012 and have been providing IT solutions to businesses in the education and retail industries ever since.

Their main business offerings include:

- Internet-based applications
- IT products
- Online reporting tools
- Mobile applications
- Learning management systems

They develop end-to-end business solutions that meet their clients' needs and goals. They have a team of experienced developers with expertise in various technologies. Their target customers are businesses in the education and retail industries. They also offer their services to businesses in other industries, such as healthcare, manufacturing, and financial services.

Their competitive advantages include:

- Their focus on end-to-end solutions
- Their team of experienced developers
- Their expertise in a variety of technologies
- Their commitment to quality

Their plans include:

- Expanding their business to other countries
- Developing new products and services
- Continuing to provide their clients with high-quality solutions

Here are some of their notable projects:

- They developed a mobile application for a school that allows students to access their coursework, assignments, and grades on the go.
- They developed an online reporting tool for a retail company that allows them to track sales data and inventory levels.
- They developed a learning management system for a university that allows students to take online courses and track their progress.

If you are looking for a software development company to help you with your business, Navikaran Infotech Pvt. Ltd. is a good option. They have a proven track record of success and can provide the solutions you need to achieve your goals.

CHAPTER 1

INTRODUCTION

1.1 Introduction to Machine Learning

Machine learning, a subset of artificial intelligence, is a transformative field that has redefined the way computers learn and make decisions. It empowers machines to recognize patterns, derive insights from data, and improve their performance over time without explicit programming. In recent years, machine learning has revolutionized industries ranging from healthcare and finance to marketing and autonomous vehicles, making it a pivotal technology to understand and harness.

1.2 The Evolution of Machine Learning

Traditional software systems operate based on explicitly programmed instructions, following predefined rules and algorithms. While these systems are reliable for well-defined tasks, they struggle to adapt to the complexity and variability of real-world data and problems.

Machine learning, on the other hand, takes a different approach. It enables computers to learn from data by identifying patterns and relationships, allowing them to generalize and make predictions or decisions in new, unseen situations. This adaptability is what sets machine learning apart and makes it an invaluable tool in solving complex, data-driven problems.

1.3 Terminologies in Machine Learning

1. Data: Data is the lifeblood of machine learning. Algorithms learn from historical and real-time data, using it to recognize patterns and make predictions. The quality and quantity of data play a crucial role in the performance of machine learning models.

2. Algorithms: Machine learning algorithms are the mathematical engines that learn from data. They come in various forms, including supervised learning (where models are trained on labeled data), unsupervised learning (where models find patterns in unlabeled data), and reinforcement learning (where models learn through trial and error).

3. Training: The process of training a machine learning model involves exposing it to data, allowing it to learn patterns and relationships. The model is fine-tuned iteratively until it reaches a desired level of accuracy.

4. Validation and Testing: After training, models are validated and tested on new, unseen data to ensure they can generalize well and make accurate predictions or decisions.

5. Features: Features are the variables or attributes used to describe data. Feature selection and engineering are critical in building effective machine learning models.

1.4 Applications of Machine Learning

1. Healthcare: Predicting diseases, identifying medical conditions, and drug discovery.
Finance: Fraud detection, algorithmic trading, and credit risk assessment.
2. Marketing: Customer segmentation, recommendation systems, and personalized advertising.
3. Autonomous Systems: Self-driving cars, robotics, and drones.
4. Natural Language Processing: Language translation, chatbots, and sentiment analysis.
5. Image and Speech Recognition: Facial recognition, speech-to-text, and object detection.

1.5 Introduction to Data Analytics

Data analytics is the process of examining, cleansing, transforming, and interpreting data to extract valuable insights, inform decision-making, and drive business strategies. In today's data-driven world, organizations across all industries are increasingly relying on data analytics to gain a competitive edge, optimize operations, and uncover hidden opportunities.

1.6 The Evolution of Data Analytics

We are living in an era often described as the "data revolution." The proliferation of digital technologies, the internet, and connected devices has generated an unprecedented volume of data. This data encompasses a wide range of information, including customer behavior, market trends, financial transactions, sensor readings, and social media interactions. Within this vast sea of data lies a treasure trove of knowledge waiting to be discovered.

1.7 Applications of Data Analytics

Data analytics plays a pivotal role in transforming raw data into actionable insights. It encompasses a spectrum of techniques and tools that enable organizations to:

1. Understand Patterns and Trends: Data analytics allows businesses to identify recurring patterns and trends within their data. This insight can shed light on customer preferences, market dynamics, and operational inefficiencies.

2. Predict Future Events: Predictive analytics leverages historical data to forecast future outcomes. This capability is invaluable for anticipating customer demand, optimizing inventory, and mitigating risks.

3. Optimize Decision-Making: By providing data-driven insights, analytics empowers decision-makers to make informed choices. It aids in strategy development, marketing campaigns, product development, and resource allocation.

4. Improve Efficiency: Data analytics identifies bottlenecks and areas of inefficiency in processes. This knowledge enables organizations to streamline operations, reduce costs, and enhance productivity.

5. Enhance Customer Experience: Through analyzing customer data, organizations can personalize products and services, tailor marketing efforts, and improve customer support, leading to higher satisfaction and loyalty.

1.8 Key Components of Data Analytics

1. Descriptive Analytics: Descriptive analytics focuses on summarizing historical data to provide a snapshot of past events and trends. It forms the foundation for understanding what has happened.
2. Diagnostic Analytics: Diagnostic analytics delves deeper into data to understand why certain events occurred. It helps in identifying the root causes of issues or successes.
3. Predictive Analytics: Predictive analytics uses historical data to create models that forecast future outcomes. This forward-looking approach is invaluable for planning and decision-making.
4. Prescriptive Analytics: Prescriptive analytics not only predicts future outcomes but also suggests actions to optimize those outcomes. It provides actionable recommendations based on data.

1.9 Application of these two in Sales Forecasting

Application of Data Analytics and Machine Learning in Sales Forecasting
Sales forecasting is a crucial aspect of business strategy, and the integration of data analytics and machine learning has revolutionized the accuracy and effectiveness of this process. Let's explore how these two powerful technologies are applied in the realm of sales forecasting:

Data Analytics in Sales Forecasting

1. Historical Data Analysis: Data analytics examines past sales data to identify patterns, trends, and seasonality. It helps in understanding historical sales performance, which is fundamental for forecasting.
2. Customer Segmentation: By segmenting customers based on behavior, preferences, and demographics, data analytics enables businesses to tailor sales forecasts and marketing strategies for different customer groups.
3. Market Research: Data analytics incorporates market research data to gain insights into consumer sentiment, economic conditions, and industry trends. This external data enhances the accuracy of sales forecasts.

4. Inventory Optimization: Analyzing inventory turnover rates and historical demand patterns aids in optimizing inventory levels. This ensures products are available when customers want them, reducing carrying costs and stockouts.
5. Pricing Strategy: Data analytics helps in evaluating the impact of different pricing strategies on sales volume. It identifies optimal price points and discounts to maximize revenue and profit margins.

Machine Learning in Sales Forecasting

1. Demand Prediction: Machine learning models analyze historical sales data along with various influencing factors (e.g., marketing spend, seasonality, economic indicators) to predict future demand accurately.
2. Time Series Forecasting: ML models like ARIMA and Prophet excel in handling time-dependent data. They are used to forecast sales for products with seasonality and trends.
3. Dynamic Pricing: Machine learning algorithms can dynamically adjust pricing in real-time based on demand fluctuations, competitor pricing, and other market dynamics. This maximizes revenue.
4. Customer Behavior Analysis: Machine learning can identify and analyze complex customer behavior patterns, helping businesses understand customer preferences and anticipate future buying decisions.
5. Performance Monitoring: ML models continuously monitor their own performance. When discrepancies arise between predicted and actual sales, the models can be retrained with fresh data to maintain accuracy.
6. Market Basket Analysis: Machine learning can identify product combinations frequently purchased together, enabling cross-selling and upselling opportunities.

By combining data analytics and machine learning in sales forecasting we can achieve:

1. Enhanced Accuracy: Data analytics provides valuable historical context, while machine learning models capture complex relationships, resulting in highly accurate sales forecasts.
2. Adaptability: Machine learning models adapt to changing market conditions, providing real-time insights and predictions, while data analytics helps in understanding the broader context.
3. Optimized Strategies: The combination allows businesses to optimize marketing strategies, pricing, and inventory management, resulting in improved profitability.
4. Customer-Centric Approach: Together, they enable a customer-centric approach, tailoring sales strategies to individual customer segments.

In essence, data analytics and machine learning have transformed sales forecasting into a sophisticated, data-driven process that empowers businesses to make informed decisions, enhance customer experiences, and stay competitive in dynamic markets. These technologies

have become indispensable tools for businesses looking to navigate the complexities of sales forecasting effectively. This project aims to harness the power of data analytics to tackle the challenge of sales prediction as a regression problem.

1.10 The Significance of Sales Prediction

Understanding future sales trends is fundamental to strategic planning for businesses of all sizes and industries. Whether you're a retail giant, an e-commerce startup, or a traditional brick-and-mortar store, the ability to anticipate market demand and consumer behavior is essential. Accurate sales forecasts enable companies to:

1. Resource Allocation: By knowing when and where sales will likely peak or dip, companies can allocate resources such as staff, inventory, and marketing budgets more effectively. This prevents over-investment during slow periods and avoids stockouts during high-demand seasons.
2. Inventory Management: Maintaining an optimal inventory level is a delicate balance. Predictive models can help reduce excess inventory costs and minimize losses due to stockouts, enhancing supply chain efficiency.
3. Marketing Strategies: Sales forecasts inform marketing teams about when and where to run promotions, discounts, and advertising campaigns. This precision ensures that marketing budgets are spent efficiently, targeting the right audience at the right time.
4. Financial Planning: Accurate sales predictions facilitate financial planning, helping businesses secure loans, manage cash flow, and set realistic revenue targets.
5. Competitive Advantage: In a fast-paced business environment, the ability to respond proactively to market changes gives companies a competitive edge. Accurate sales predictions can help businesses outmaneuver their competitors.

CHAPTER-2

ANALYSIS OF DATA ANALYTICS AND MACHINE LEARNING

Machine learning and data analytics are powerful techniques in the field of artificial intelligence and data science. They involve the use of algorithms and statistical models to enable computer systems to improve their performance on a specific task through learning from data. In recent years, these fields have gained significant attention due to the increasing availability of large datasets and advancements in computational technology.

Analysis of machine learning and data analytics involves a deep understanding of algorithms, data manipulation techniques, programming skills, and ethical considerations. Professionals in these fields play a vital role in extracting meaningful insights from data, enabling businesses and organizations to make informed decisions and solve complex problems.

2.1 Machine Learning and Data Analytics Analysis Steps

Analyzing data in both machine learning and data analytics involves several common steps and techniques, although the specific approaches may vary based on the goals and methodologies of each field. Figure 2.1 shows the block diagram of data science lifecycle.

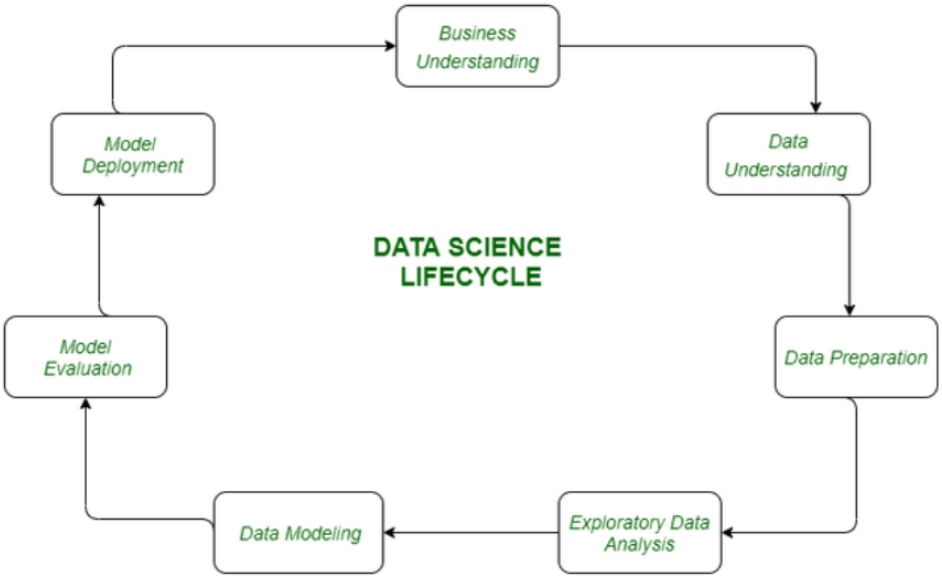


Fig 2.1 Block diagram of Data Science Lifecycle

1. Data Collection

In both fields, the first step is to gather relevant data from various sources. This could include structured data from databases, spreadsheets, or unstructured data from sources like text documents, images, or social media.

2. Data Preprocessing

Data cleaning: This involves handling missing values, outliers, and errors in the data. It ensures the dataset is of high quality.

Data transformation: Data may need to be transformed, scaled, or normalized to make it suitable for analysis and modeling.

Feature engineering: In machine learning, feature engineering is the process of selecting, creating, or transforming features (variables) to improve the performance of machine learning models.

3. Exploratory Data Analysis (EDA)

EDA involves visualizing and understanding the data before diving into modeling. Techniques like histograms, scatter plots, and summary statistics help uncover patterns and insights.

4. Data Modeling

In machine learning, this is the stage where models are selected, trained, and evaluated. Various algorithms and techniques are used to build predictive models based on the data.

5. Model Evaluation

In machine learning, models are evaluated using metrics like accuracy, precision, recall, F1-score, and ROC curves, depending on the problem type (classification, regression, etc.).

In data analytics, the focus is on understanding patterns and trends, and the evaluation might involve summary statistics, correlation analysis, or hypothesis testing.

6. Visualization

Both fields use visualization techniques to present findings and insights in a clear and interpretable manner. Tools like charts, graphs, and dashboards are commonly used for visualization.

7. Interpretation

In data analytics, the goal is often to understand historical data and make informed decisions. Interpretation involves drawing meaningful conclusions from the data to inform business strategies.

In machine learning, model interpretation is crucial to understand why a model makes specific predictions. Techniques like feature importance analysis, SHAP values, and partial dependence plots are used for model interpretation.

8. Iteration
In both fields, data analysis is often an iterative process. This means that after initial analysis, new hypotheses may be generated, additional data collected, or models retrained for improved results.

9. Deployment
In data analytics, the insights gained from data analysis can inform decision-making, strategy development, and operational improvements.
In machine learning, trained models are deployed in real-world applications to make predictions or automate decision-making processes
.
10. Monitoring
Both fields require ongoing monitoring to ensure that data quality remains high and that models (in the case of machine learning) continue to perform well in a changing environment.

11. Documentation
Comprehensive documentation of the data, analysis methods, and findings is essential for reproducibility and knowledge sharing.

It's important to note that while there is overlap between machine learning and data analytics, the primary difference lies in their objectives. Machine learning focuses on building predictive models, while data analytics aims to extract insights and make data-driven decisions. The choice of techniques and tools depends on the specific goals and requirements of each field.

2.2 Programming Frameworks Used

Python

Python is a high-level, versatile programming language known for its simplicity and readability. It has gained immense popularity in data analytics and machine learning due to its extensive libraries and packages for data manipulation, analysis, and visualization. Python's open-source nature and vibrant community make it a preferred choice for data scientists and analysts.

NumPy

NumPy (Numerical Python) is a fundamental library for numerical computing in Python. It supports multidimensional arrays, matrices, and a wide range of mathematical functions to perform operations on these arrays efficiently. NumPy is the cornerstone of many data science libraries, enabling efficient storage and manipulation of large datasets.

Pandas

Pandas is a data manipulation library in Python that offers data structures and functions to handle structured data, primarily in the form of dataframes. Dataframes are two-dimensional, tabular data structures resembling spreadsheets or SQL tables. Pandas simplifies tasks like data cleaning, transformation, indexing, and filtering, making it an essential tool for data preprocessing.

Matplotlib

Matplotlib is a widely-used data visualization library in Python. It provides many functions for creating static, interactive, and publication-quality plots and charts. Matplotlib is highly customizable, allowing users to control every aspect of their visualizations, from plot types to color schemes and annotations.

Seaborn

Seaborn is a data visualization library built on top of Matplotlib. It specializes in creating informative and aesthetically pleasing statistical visualizations. Seaborn simplifies complex visualizations like heat maps, pair plots, and violin plots, making it an excellent choice for exploring data and understanding relationships between variables.

Scikit-learn

scikit-learn is a machine learning library for Python that provides many tools and algorithms for tasks such as classification, regression, clustering, dimensionality reduction, and more. It is built on NumPy, SciPy, and Matplotlib and is designed to be user-friendly and accessible. Scikit-learn facilitates model training, evaluation, and deployment, making it a go-to choice for machine learning practitioners.

In my project, Python was the programming language used for development, NumPy enabled efficient data handling, pandas facilitated data preprocessing, Matplotlib and Seaborn were used for data visualization, and scikit-learn provided the necessary machine learning algorithms for building predictive models. Discuss specific functions and methods from these libraries that were instrumental in your project success, and highlight how they contributed to achieving objectives.

2.3 Tools Used

Jupyter Notebook

Jupyter Notebook is an open-source, web-based interactive computing environment that is widely used in data science and scientific computing. It allows you to create and share documents that contain live code, equations, visualizations, and narrative text. Here are some key features and uses:

- Interactive Coding: Jupyter Notebook provides a cell-based interface where you can write and execute code interactively. Each cell can contain code in different programming languages, including Python, R, and more.
- Data Exploration: Jupyter Notebook is excellent for data exploration and analysis. You can load datasets, perform data manipulations, visualize data, and share insights within a single document.
- Visualization: You can easily create data visualizations using libraries like Matplotlib and Seaborn, displayed in the notebook.

- Documentation: Jupyter Notebooks allow you to mix code with narrative text using Markdown. This makes it a powerful tool for creating data analysis reports, tutorials, and presentations.
- Reproducibility: Jupyter Notebooks are great for ensuring the reproducibility of your work since they capture both code and its execution results.

Google Colab Notebook

Google Colab (short for Colaboratory) is a cloud-based Jupyter Notebook platform provided by Google. It offers a free and convenient environment for running Jupyter Notebooks in the cloud. Here's why it's popular:

- No Setup Required: With Google Colab, you don't need to install anything locally. You can access Jupyter Notebooks directly through a web browser.
- Free GPU/TPU: Google Colab provides free access to Graphics Processing Units (GPUs) and Tensor Processing Units (TPUs), which are extremely useful for training machine learning models.
- Collaboration: Multiple users can collaborate on the same notebook simultaneously, making it suitable for team projects.
- Integration with Google Services: Colab seamlessly integrates with other Google services like Google Drive, making storing and sharing notebooks and data easy.
- Libraries and Dependencies: Colab comes pre-installed with many data science and machine learning libraries, saving you the trouble of manual installations.

Visual Studio Code (VS Code)

Visual Studio Code is a lightweight, open-source code editor developed by Microsoft. While it's not a full-fledged Integrated Development Environment (IDE), it's highly customizable and has many extensions available. Here's why it's popular among developers:

- Extensibility: VS Code can be customized with extensions supporting multiple programming languages, including Python. You can enhance its functionality to match your project's requirements.
- Integrated Terminal: It features an integrated terminal that allows you to run code, execute commands, and manage your project within the same interface.
- Version Control: VS Code has built-in support for version control systems like Git, making it easy to track changes and collaborate.
- Code Navigation: The editor provides powerful code navigation tools, such as intelligent code completion, debugging capabilities, and integrated documentation.
- Lightweight: It's lightweight compared to full IDEs, which makes it fast and responsive even on less powerful hardware.
- Cross-Platform: VS Code is available on Windows, macOS, and Linux, ensuring a consistent development experience across different operating systems.

In your project report, you can highlight how you used these tools to facilitate different aspects of your work. For instance, you may discuss how Jupyter Notebooks were used for data exploration and documentation, Google Colab for cloud-based collaboration and GPU-accelerated model training, and Visual Studio Code for code development, version control, and extensions tailored to your project needs.

CHAPTER-3

PROJECT ANALYSIS OF SALES FORECASTING USING MACHINE LEARNING AND DATA ANALYTICS

3.1 Dataset Overview:

- **Dataset Link:** https://www.kaggle.com/datasets/aslanahmedov/walmart-sales-forecast
- **Dataset Name:** Walmart Sales Forecasting Dataset
- **Dataset Source:** Kaggle (https://www.kaggle.com/)
- **Data Collection or Obtaining:** The dataset was likely collected or uploaded to Kaggle by individuals or organizations interested in sharing and analyzing Walmart sales data for forecasting purposes. It may contain historical sales data for various Walmart stores, allowing data scientists and analysts to develop predictive models for sales forecasting.
- **Project Link: Colab Notebook:** https://colab.research.google.com/drive/1ArUuLYCk-28TuzIrKWrooRNA4iT3lRdh?usp=sharing#scrollTo=b-Uowk35mRoF

3.2 Dataset Size and Structure:

1. Indicate the number of rows and columns in the dataset.

```
features.shape
```

```
(8190, 12)
```

Fig 3.2.1 Finding the size of features dataset

```
test.shape
```

```
(115064, 4)
```

Fig 3.2.2 Finding the size of test dataset

```
train.shape
```

```
(421570, 5)
```

Fig 3.2.3 Finding the size of train dataset

```
stores.shape
```

```
(45, 3)
```

Fig 3.2.4 Finding the size of stores dataset

2.Describe the data structure (e.g., tabular, time series, text, images).

```
features.describe()
```

	Store	Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	MarkDown4	MarkDown5	CPI	Unemployment
count	8190.000000	8190.000000	8190.000000	4032.000000	2921.000000	3613.000000	3464.000000	4050.000000	7605.000000	7605.000000
mean	23.000000	59.356198	3.405992	7032.371786	3384.176594	1760.100180	3292.935886	4132.216422	172.460809	7.826821
std	12.987966	18.678607	0.431337	9262.747448	8793.583016	11276.462208	6792.329861	13086.690278	39.738346	1.877259
min	1.000000	-7.290000	2.472000	-2781.450000	-265.760000	-179.260000	0.220000	-185.170000	126.064000	3.684000
25%	12.000000	45.902500	3.041000	1577.532500	68.880000	6.600000	304.687500	1440.827500	132.364839	6.634000
50%	23.000000	60.710000	3.513000	4743.580000	364.570000	36.260000	1176.425000	2727.135000	182.764003	7.806000
75%	34.000000	73.880000	3.743000	8923.310000	2153.350000	163.150000	3310.007500	4832.555000	213.932412	8.567000
max	45.000000	101.950000	4.468000	103184.980000	104519.540000	149483.310000	67474.850000	771448.100000	228.976456	14.313000

Fig 3.2.5 Describe the data structure of features dataset

```
stores.describe()
```

	Store	Size
count	45.000000	45.000000
mean	23.000000	130287.600000
std	13.133926	63825.271991
min	1.000000	34875.000000
25%	12.000000	70713.000000
50%	23.000000	126512.000000
75%	34.000000	202307.000000
max	45.000000	219622.000000

Fig 3.2.6 Describe the data structure of stores dataset

```
test.describe()
```

	Store	Dept
count	115064.000000	115064.000000
mean	22.238207	44.339524
std	12.809930	30.656410
min	1.000000	1.000000
25%	11.000000	18.000000
50%	22.000000	37.000000
75%	33.000000	74.000000
max	45.000000	99.000000

Fig 3.2.7 Describe the data structure of test dataset

```
train.describe()
```

	Store	Dept	Weekly_Sales
count	421570.000000	421570.000000	421570.000000
mean	22.200546	44.260317	15981.258123
std	12.785297	30.492054	22711.183519
min	1.000000	1.000000	-4988.940000
25%	11.000000	18.000000	2079.650000
50%	22.000000	37.000000	7612.030000
75%	33.000000	74.000000	20205.852500
max	45.000000	99.000000	693099.360000

Fig 3.2.8 Describe the data structure of data sets

3.3 Data Attributes:

List all the attributes (columns) in the dataset.

```
features.columns
```

```
Index(['Store', 'Date', 'Temperature', 'Fuel_Price', 'MarkDown1', 'MarkDown2',
       'MarkDown3', 'MarkDown4', 'MarkDown5', 'CPI', 'Unemployment',
       'IsHoliday'],
      dtype='object')
```

Fig 3.3.1 List all the attributes (columns) in the features dataset

```
stores.columns
```

```
Index(['Store', 'Type', 'Size'], dtype='object')
```

Fig 3.3.2 List all the attributes (columns) in the stores dataset

```
test.columns
```

```
Index(['Store', 'Dept', 'Date', 'IsHoliday'], dtype='object')
```

Fig 3.3.3 List all the attributes (columns) in the test dataset

```
train.columns
```

```
Index(['Store', 'Dept', 'Date', 'Weekly_Sales', 'IsHoliday'], dtype='object')
```

Fig 3.3.4 List all the attributes (columns) in the train dataset

3.4 Data Type of Attributes:

Describe the types of all the attributes.

```
features.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8190 entries, 0 to 8189
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Store         8190 non-null   int64
 1   Date          8190 non-null   object
 2   Temperature   8190 non-null   float64
 3   Fuel_Price    8190 non-null   float64
 4   MarkDown1     4032 non-null   float64
 5   MarkDown2     2921 non-null   float64
 6   MarkDown3     3613 non-null   float64
 7   MarkDown4     3464 non-null   float64
 8   MarkDown5     4050 non-null   float64
 9   CPI           7605 non-null   float64
 10  Unemployment  7605 non-null   float64
 11  IsHoliday     8190 non-null   bool
dtypes: bool(1), float64(9), int64(1), object(1)
memory usage: 712.0+ KB
```

Fig 3.4.1 Describe the types of all the attributes of features dataset

```
stores.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45 entries, 0 to 44
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Store   45 non-null     int64
 1   Type    45 non-null     object
 2   Size    45 non-null     int64
dtypes: int64(2), object(1)
memory usage: 1.2+ KB
```

Fig 3.4.2 Describe the types of all the attributes of stores dataset

```
test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 115064 entries, 0 to 115063
Data columns (total 4 columns):
 #   Column     Non-Null Count   Dtype
---  ------     --------------   -----
 0   Store      115064 non-null  int64
 1   Dept       115064 non-null  int64
 2   Date       115064 non-null  object
 3   IsHoliday  115064 non-null  bool
dtypes: bool(1), int64(2), object(1)
memory usage: 2.7+ MB
```

Fig 3.4.3 Describe the types of all the attributes of test dataset

```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 421570 entries, 0 to 421569
Data columns (total 5 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   Store         421570 non-null  int64
 1   Dept          421570 non-null  int64
 2   Date          421570 non-null  object
 3   Weekly_Sales  421570 non-null  float64
 4   IsHoliday     421570 non-null  bool
dtypes: bool(1), float64(1), int64(2), object(1)
memory usage: 13.3+ MB
```

Fig 3.4.4 Describe the types of all the attributes of train dataset

3.5 Steps to Solve the Sales Forecasting Problem

Step-1: Import Necessary Dependencies and Libraries

At the outset of any machine learning project, importing the necessary Python libraries and dependencies is essential. These libraries will provide tools and functions for data manipulation, analysis, visualization, and modeling.

For example, you might import numpy for numerical operations, pandas for data manipulation, matplotlib and seaborn for data visualization, and scikit-learn for machine learning algorithms. Each of these libraries serves a unique purpose in the machine learning pipeline.

```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import numpy as np
```

Fig 3.5 importing libraries of python

Step 2: Unzipping the Files for Creating a Dataset

If your dataset is stored in a compressed format (e.g., a ZIP file), you must unzip it. This step ensures you have access to the raw data required for analysis.

```
!unzip /content/walmart-recruiting-store-sales-forecasting.zip
```

```
unzip:  cannot find or open /content/walmart-recruiting-store-sales-forecasting.zip, /content/walmart-recruiting-store-sales-forecasting.zip.zip or /c
```

```
!unzip /content/test.csv.zip
```

```
Archive:  /content/test.csv.zip
  inflating: test.csv
```

```
!unzip /content/train.csv.zip
```

```
Archive:  /content/train.csv.zip
  inflating: train.csv
```

Fig 3.6 Unzipping the files

Step-3: Look at the Dataset Using Pandas DataFrame.

Once you've obtained the dataset, load it into a Pandas DataFrame. A DataFrame is a two-dimensional, tabular data structure allowing efficient data manipulation. With the DataFrame, you can perform initial exploratory data analysis (EDA). This includes tasks such as:

1. Using df.head() to view the first few rows of the dataset.

```
features = pd.read_csv('features.csv')
features.head()
```

	Store	Date	Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	MarkDown4	MarkDown5	CPI	Unemployment	IsHoliday
0	1	2010-02-05	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358	8.106	False
1	1	2010-02-12	38.51	2.548	NaN	NaN	NaN	NaN	NaN	211.242170	8.106	True
2	1	2010-02-19	39.93	2.514	NaN	NaN	NaN	NaN	NaN	211.289143	8.106	False
3	1	2010-02-26	46.63	2.561	NaN	NaN	NaN	NaN	NaN	211.319643	8.106	False
4	1	2010-03-05	46.50	2.625	NaN	NaN	NaN	NaN	NaN	211.350143	8.106	False

Fig 3.7 Viewing the first few rows of the dataset

2. Using df.info() to check data types and missing values.

```
features.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8190 entries, 0 to 8189
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Store         8190 non-null   int64
 1   Date          8190 non-null   object
 2   Temperature   8190 non-null   float64
 3   Fuel_Price    8190 non-null   float64
 4   MarkDown1     4032 non-null   float64
 5   MarkDown2     2921 non-null   float64
 6   MarkDown3     3613 non-null   float64
 7   MarkDown4     3464 non-null   float64
 8   MarkDown5     4050 non-null   float64
 9   CPI           7605 non-null   float64
 10  Unemployment  7605 non-null   float64
 11  IsHoliday     8190 non-null   bool
dtypes: bool(1), float64(9), int64(1), object(1)
memory usage: 712.0+ KB
```

Fig 3.8 checking data types and missing values

3. Employing df.describe() for summary statistics.

```
features.describe()
```

	Store	Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	MarkDown4	MarkDown5	CPI	Unemployment
count	8190.000000	8190.000000	8190.000000	4032.000000	2921.000000	3613.000000	3464.000000	4050.000000	7605.000000	7605.000000
mean	23.000000	59.356198	3.405992	7032.371786	3384.176594	1760.100180	3292.935886	4132.216422	172.460809	7.826821
std	12.987966	18.678607	0.431337	9262.747448	8793.583016	11276.462208	6792.329861	13086.690278	39.738346	1.877259
min	1.000000	-7.290000	2.472000	-2781.450000	-265.760000	-179.260000	0.220000	-185.170000	126.064000	3.684000
25%	12.000000	45.902500	3.041000	1577.532500	68.880000	6.600000	304.687500	1440.827500	132.364839	6.634000
50%	23.000000	60.710000	3.513000	4743.580000	364.570000	36.260000	1176.425000	2727.135000	182.764003	7.806000
75%	34.000000	73.880000	3.743000	8923.310000	2153.350000	163.150000	3310.007500	4832.555000	213.932412	8.567000
max	45.000000	101.950000	4.468000	103184.980000	104519.540000	149483.310000	67474.850000	771448.100000	228.976456	14.313000

Fig 3.9 Describe the dataset

4. Visualizing data distributions and relationships using Matplotlib and Seaborn.

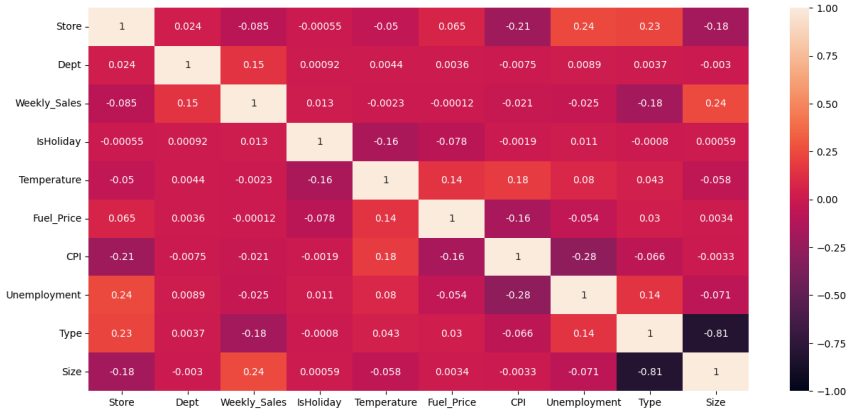


Fig 3.10 Visualizing data distributions and relationships

Step 4: Create a DataFrame of Features for Modeling

In this step, you select the relevant features (columns) from the dataset input to your machine-learning models. These features are typically referred to as independent variables.

Careful feature selection is crucial as it impacts model performance. When deciding which features to include, you should consider domain knowledge and feature importance analysis.

Step-5: Data Preprocessing or Cleaning

Data preprocessing is a critical step that ensures the data is in a suitable format for machine learning. Common preprocessing tasks include:

1. Handling missing data using methods like imputation or removal.

```
(df.isnull().sum()/len(df))*100
```

```
Store            0.000000
Dept             0.000000
Date             0.000000
Weekly_Sales     0.000000
IsHoliday        0.000000
Temperature      0.000000
Fuel_Price       0.000000
MarkDown1       64.257181
MarkDown2       73.611025
MarkDown3       67.480845
MarkDown4       67.984676
MarkDown5       64.079038
CPI              0.000000
Unemployment     0.000000
Type             0.000000
Size             0.000000
dtype: float64
```

Fig 3.11 Handling missing values

2. Removing duplicate records if they exist.
3. Scaling or normalizing features to ensure uniform scales.
4. Handling categorical variables through encoding (e.g., one-hot encoding for nominal data).
5. Handling date and time features if present in the dataset.

These preprocessing steps aim to improve data quality and enhance the performance of machine learning algorithms.

Step-6: Removing Outliers

Outliers are data points that deviate significantly from most data. They can skew model predictions and should be addressed. Techniques for outlier handling include visualization plots, scatter plots), statistical methods (z-scores, IQR), and domain knowledge-based approaches.

IQR Method analysis:

The Interquartile Range (IQR) method is a statistical technique used to identify and remove outliers from a dataset. Outliers are data points that significantly deviate from the central tendency of the data and can distort statistical analyses. The IQR method is particularly useful because it defines outliers based on the spread of the data rather than relying on fixed thresholds. Here's a detailed explanation of the IQR method for removing outliers:

Step 1: Calculate the Interquartile Range (IQR)

The IQR is a measure of statistical dispersion and is defined as the difference between the third quartile (Q3) and the first quartile (Q1) of a dataset. It represents the middle 50% of the data.

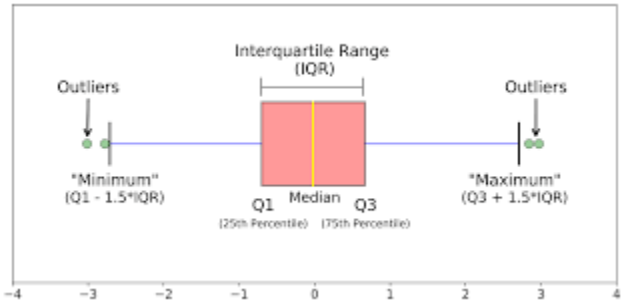


Fig 3.12 Finding outliers using IQR

Mathematically, the IQR is calculated as

$IQR = Q3 - Q1$,

Where,

- Q1 (First Quartile) is the 25th percentile of the data, meaning that 25% of the data points are below this value.
- Q3 (Third Quartile) is the 75th percentile of the data, meaning that 75% of the data points are below this value.

Step 2: Define the Lower and Upper Boundaries

To identify potential outliers, you must define lower and upper boundaries based on the IQR. Outliers are typically defined as data points that fall below the lower boundary or above the upper boundary. These boundaries are calculated as follows:

- Lower Boundary: $Q1 - (k \times IQR)$
- Upper Boundary: $Q3 + (k \times IQR)$

Here, k is a user-defined constant determining the range for considering data points as outliers. Common choices for k include 1.5 and 3, although it can vary depending on the level of stringency desired.

Step 3: Identify Outliers

Once you have calculated the lower and upper boundaries, you can scan the dataset to identify data points that fall below the lower boundary or above the upper boundary. These data points are considered outliers.

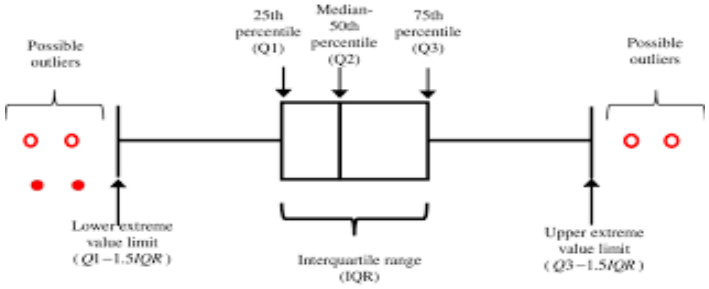


Fig 3.13 Outlier Detection

Step 4: Decide How to Handle Outliers

After identifying outliers, you must decide how to handle them. Here are some common strategies:

- Remove Outliers: You can remove the identified outliers from the dataset. This is suitable when you believe the outliers are due to data entry or measurement errors.
- Transform Data: In some cases, you might prefer to transform the data to make it more resistant to outliers. For example, you can apply a logarithmic transformation or use a robust statistical method.
- Use Robust Statistical Techniques: Instead of removing outliers, you can use statistical methods that are robust to outliers. For example, using the median instead of the mean for central tendency estimation can be less influenced by outliers.
- Flag Outliers: Instead of removing outliers, you can flag them for further investigation or analysis. This approach retains the data while allowing you to assess the impact of outliers on your results.

Step 5: Apply the Chosen Strategy

Implement the chosen strategy for handling outliers. This step may involve removing or transforming data points or using robust statistical techniques in subsequent analyses.

Step 6: Reassess and Validate

After applying the chosen strategy, reassess and analyze your dataset to ensure it aligns with your research objectives. Validate the impact of your outlier-handling strategy on the results of your analysis.

The IQR method is a robust and data-driven approach for identifying and handling dataset outliers. It is handy when dealing with data that may not conform to a specific distribution and when a flexible approach to outlier detection is required.

```
# Removing Outliers:

columns = ['Temperature', 'Fuel_Price', 'CPI', 'Unemployment', 'Size']

Q3 = df[columns].quantile(.75)
Q1 = df[columns].quantile(.25)
IQR = Q3 - Q1
UL = Q3 + 1.5*IQR
LL = Q1 -  1.5*IQR

for column in columns:
    df[column] = np.where(df[column] > UL[column], UL[column], np.where(df[column] < LL[column], LL[column], df[column]))
```

Fig 3.14 Removing Outliers

Step-7: Convert Categorical Features to Numerical Features

Machine learning algorithms typically use numerical data.

Original Data

Team	Points
A	25
A	12
B	15
B	14
B	19
B	23
C	25
C	29

Label Encoded Data

Team	Points
0	25
0	12
1	15
1	14
1	19
1	23
2	25
2	29

Fig 3.15 Label Encoded Data

Therefore, categorical features need to be converted into a numerical format. Common techniques include one-hot encoding, label encoding, or ordinal encoding, depending on the nature of the data.

Step-8: Checking Multicollinearity

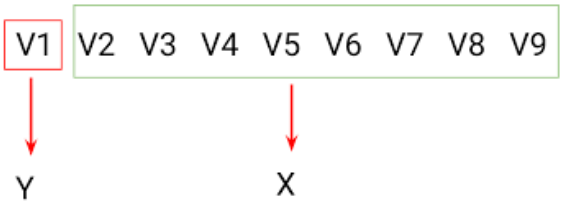
Multicollinearity occurs when independent variables in a regression model are highly correlated. This can lead to unstable coefficient estimates. To address multicollinearity, you may use correlation analysis or variance inflation factor (VIF) calculations to identify and mitigate the issue.

VIF Analysis for Removing Multicollinearity:

The Variance Inflation Factor (VIF) is a statistical technique used to assess and quantify multicollinearity in regression analysis. Multicollinearity occurs when two or more independent variables in a regression model are highly correlated. This can pose problems because it can make it challenging to interpret the individual effect of each independent variable on the dependent variable, leading to unstable coefficient estimates.

The VIF method helps identify which independent variables contribute to multicollinearity and provides a measure of the severity of this issue. Here's how the VIF method works:

Step 1: Fit a Multiple Linear Regression Model:



Fitting a multiple linear regression model using your dataset's independent variables (features). In this model, one of the variables serves as the dependent variable, and the others are treated as independent variables.

Step 2: Calculate VIF for Each Independent Variable:
For each independent variable in the regression model, calculate its VIF score using the following formula:

$$VIF = \frac{1}{1 - R_i^2}$$

The VIF score quantifies how much the variance of the estimated coefficient for the independent variable is increased due to multicollinearity. If a VIF is equal to 1, it indicates no multicollinearity, while higher values suggest increasing multicollinearity.

Step 3: Interpretation and Decision:
Evaluate the VIF scores for each independent variable. A commonly used threshold to detect multicollinearity is a VIF value of 5 or 10, although the specific threshold can vary depending on the context and the degree of tolerance for multicollinearity.
If a variable has a high VIF (e.g., VIF > 5 or 10), it suggests that it is highly correlated with other independent variables in the model. In such cases, consider addressing multicollinearity.

Step 4: Addressing Multicollinearity:

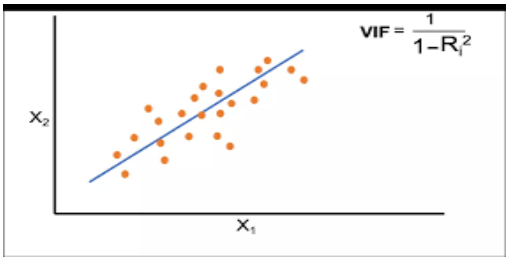


Fig 3.16 Multicollinearity

If you identify variables with high VIF values, you have several options to address multicollinearity:

a. Remove one or more highly correlated variables: One approach is to remove one or more of the highly correlated variables from the model. This should be done thoughtfully, considering the impact on the overall model and the research question.

b. Combine correlated variables: In some cases, you can create a new variable that combines the information from highly correlated variables. This can be achieved through principal component analysis (PCA) or factor analysis.

c. Domain Knowledge: Rely on domain knowledge to decide which variables are most important and should be retained in the model, even if they have high VIF values.

The VIF method aims to improve the stability and interpretability of your regression model by reducing the impact of multicollinearity. By identifying and addressing multicollinearity, you can obtain more reliable and meaningful results from your regression analysis.

```
from statsmodels.stats.outliers_influence import variance_inflation_factor

vif_data = pd.DataFrame()
vif_data['Features'] = df.columns
vif_data['VIF'] = [variance_inflation_factor(df.values, i) for i in range(len(df.columns))]
vif_data
```

Fig 3.17

Step-9: Separating Dependent and Independent Variables

In machine learning, you must distinguish between the dependent variable (target or label) and the independent variables (features). This separation allows you to train your models effectively.

```
x = df.drop(['Weekly_Sales'], axis = 1)
y = df['Weekly_Sales']
```

```
x.columns
```

```
Index(['Store', 'Dept', 'IsHoliday', 'Temperature', 'Fuel_Price', 'CPI',
       'Unemployment', 'Type'],
      dtype='object')
```

```
y.head()
```

```
Date
2010-02-05    24924.50
2010-02-05    50605.27
2010-02-05    13740.12
2010-02-05    39954.04
2010-02-05    32229.38
Name: Weekly_Sales, dtype: float64
```

Fig 3.18 Separating Dependent and Independent Variables

Step 10: Separate the Dataset into Training and Validation

Before building and training your machine learning models, it's crucial to divide your dataset into training and validation sets. The training set is used to train the model, while the validation set is used to evaluate its performance.

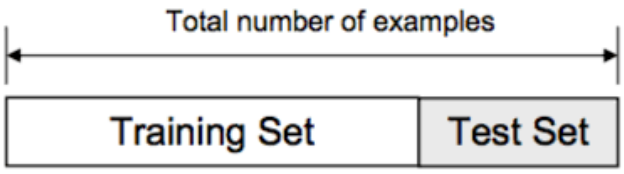


Fig 3.19 Train and Test Set

Common splits include a 70-30 or 80-20 split between training and validation data. I have used the 70-30 split, which means training data is 70% and testing data is 30%.

```
from sklearn.model_selection import train_test_split
x_train, x_val, y_train, y_val = train_test_split(x, y, test_size = 0.3, random_state = 10)
```

```
print("x Train Shape :",x_train.shape)
print("x Val Shape   :",x_val.shape)
print("y Train Shape :",y_train.shape)
print("y Val Shape   :",y_val.shape)
```

```
x Train Shape : (295099, 8)
x Val Shape   : (126471, 8)
y Train Shape : (295099,)
y Val Shape   : (126471,)
```

Step-11: Linear Regression Modeling

Linear regression is a foundational machine learning algorithm for predicting continuous target variables. It assumes a linear relationship between the independent and dependent variables.

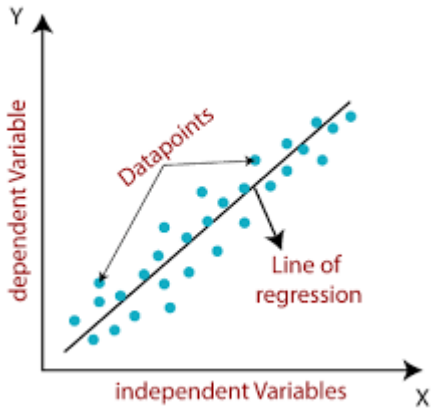


Fig 3.20 Linear Regression

The model is trained to learn the coefficients for each feature, which are used to make predictions.

```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()

lr.fit(x_train, y_train)
y_pred = lr.predict(x_val)
```

```
lr.score(x_val, y_val)
```

```
0.06149084741656563
```

Fig 3.21 Train the model through Linear regression

Step-12: Decision Tree Modeling

Decision trees are versatile machine-learning models that handle regression and classification tasks.

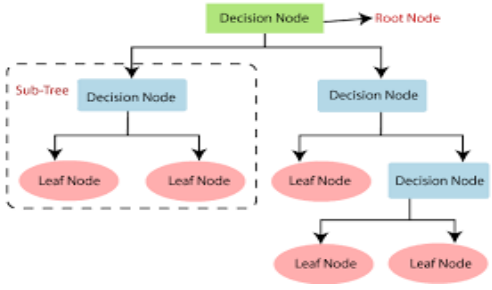


Fig 3.22 Decision Tree Algorithm

They create a tree-like structure of decision rules based on the data. Decision tree models are interpretable and can be visualized, making them useful for understanding feature importance.

```
from sklearn.tree import DecisionTreeRegressor
```

```
dt = DecisionTreeRegressor()
dt_model = dt.fit(x_train, y_train)
y_pred_dt = dt_model.predict(x_val)
```

Fig 3.23 Train the model through Decision Tree regression

Step-13: Random Forest Modeling

Random Forest is an ensemble learning method that combines multiple decision trees to improve predictive accuracy and reduce overfitting.

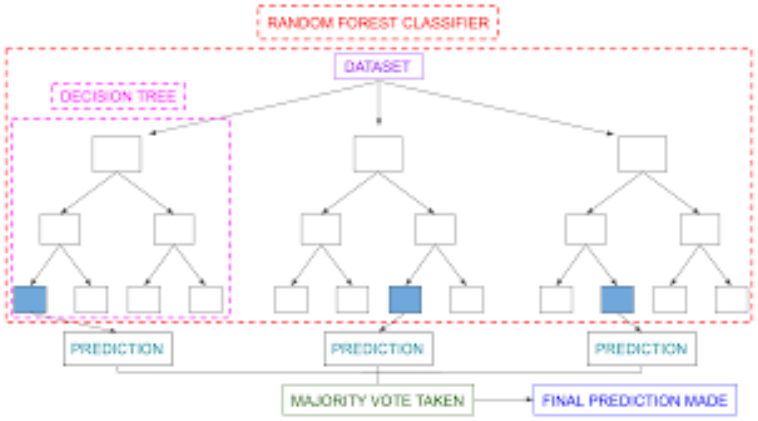


Fig 3.24 Random Forest Algorithm

It aggregates the predictions of individual decision trees to make more robust and accurate predictions.

```
from sklearn.ensemble import RandomForestRegressor
```

```
rf = RandomForestRegressor()
rf_model = rf.fit(x_train, y_train)
y_pred_rf = rf_model.predict(x_val)
```

Fig 3.24 Train the model through Random Forest

Each of these steps is a crucial component of the machine learning pipeline, contributing to the success of your sales prediction project. These steps collectively ensure the data is prepared,

features are appropriately selected, models are trained and evaluated, and the project objectives are met.

Step-14: Model Testing

I used the following two metrics for model testing.

R-squared (R^2) for model validation:

R-squared, often referred to as the coefficient of determination, is a statistical measure that assesses the goodness of fit of a regression model. It quantifies the proportion of the variance in the dependent variable (target) explained by the model's independent variables (features). R-squared values range from 0 to 1, where:

- $R^2 = 0$: The model explains none of the variance in the target variable. It provides no predictive value and essentially predicts the mean of the target for all observations.
- $R^2 = 1$: The model explains all the variance in the target variable. It perfectly fits the data, capturing all patterns and relationships.

Here's how R-squared is calculated:

$$\textit{Coefficient of Determination} \rightarrow \quad R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

$$\textit{Sum of Squares Total} \rightarrow \quad SST = \sum (y - \bar{y})^2$$

$$\textit{Sum of Squares Regression} \rightarrow \quad SSR = \sum (y' - \bar{y'})^2$$

$$\textit{Sum of Squares Error} \rightarrow \quad SSE = \sum (y - y')^2$$

Calculate the Total Sum of Squares (SST): SST represents the total variance in the target variable. It measures how much the actual values of the target deviate from their mean.

Calculate the Residual Sum of Squares (SSE): SSE represents the unexplained variance or the remaining variance after the model's predictions are subtracted from the actual target values.
Calculate R-squared (R^2):

- R-squared is a value between 0 and 1, where higher values indicate a better fit of the model to the data.
- R-squared can be interpreted as the proportion of the total variance in the target variable explained by the model.
- For example, an R^2 of 0.80 means that the model explains 80% of the variance in the target, while the remaining 20% is unexplained.

- **Root Mean Squared Error (RMSE) for model validation:**

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

RMSE is a metric that quantifies the average magnitude of the errors (residuals) between the model's predictions and the actual target values. It measures the model's accuracy in predicting the target variable. RMSE is expressed in the same units as the target variable, making it interpretable in the problem context. The lower the RMSE, the better the model's predictive accuracy.

Here's how RMSE is calculated:
Step-1: Calculate the Squared Residuals: Calculate the squared difference between the actual target and predicted values for each observation.
Step 2: Calculate the Mean of the Squared Residuals. Take the average of the squared residuals.
Step-3: Calculate RMSE:

- RMSE is the square root of the mean squared error (MSE).
- RMSE measures the typical magnitude of errors made by the model. Smaller RMSE values indicate better predictive accuracy.

In summary, R-squared (R^2) quantifies the proportion of variance explained by the model, with values closer to 1 indicating better fit. Root Mean Squared Error (RMSE) measures the typical magnitude of prediction errors, with smaller values indicating better predictive accuracy. Both metrics are crucial for assessing and comparing the performance of regression models.

CHAPTER 4

COMPARATIVE ANALYSIS

Below is a detailed comparative analysis of models based on various aspects as we studied earlier:

4.1 Model Performance

- Random Forest: Random Forest typically provides higher accuracy and better predictive performance than Decision Trees and Linear Regression. It is an ensemble method that combines multiple decision trees, reducing overfitting and improving generalization.
- Decision Tree: Decision Trees can perform well on simple datasets but are prone to overfitting on complex data. They are more interpretable than Random Forests but may not generalize as effectively.
- Linear Regression: Linear Regression is suitable for linear relationships between variables. Its performance depends on the linearity and independence assumptions, which may not always hold. It may not capture complex patterns in the data as effectively as tree-based models.

4.2 Model Interpretability

- Random Forest: Random Forest models are less interpretable than Decision Trees or Linear Regression. Understanding the decision process of individual trees within the ensemble is challenging.
- Decision Tree: Decision Trees are highly interpretable. You can easily visualize the decision rules and understand how the model makes predictions. This makes them useful for explaining the reasoning behind predictions.
- Linear Regression: Linear Regression models are also interpretable. The coefficients of the linear equation directly show the relationship between independent and dependent variables. It's easy to interpret the impact of each feature on the target variable.

4.3 Handling Non-Linearity

- Random Forest: Random Forest can capture non-linear relationships in the data effectively due to its ensemble nature and the use of multiple decision trees.
- Decision Tree: Decision Trees can model non-linear relationships but may overfit noisy data. Pruning techniques can be used to mitigate this issue.
- Linear Regression: Linear Regression assumes a linear relationship between variables. While it can model some non-linear relationships with feature engineering, it may not handle complex non-linearity as well as tree-based models.

4.4 Robustness to Outliers

- Random Forest: Random Forest is robust to outliers because it combines predictions from multiple trees, reducing the impact of individual outliers
- Random Forest: Random Forest is robust to outliers because it combines predictions from multiple trees, reducing the impact of individual outliers.
- Decision Tree: Decision Trees can be sensitive to outliers, especially when the depth of the tree is not controlled. Pruning can help mitigate this sensitivity.
- Linear Regression: Linear Regression can be sensitive to outliers because it seeks to minimize the sum of squared errors. Outliers can disproportionately influence the model's coefficients.

4.5 Scalability

- Random Forest: Random Forest can handle large datasets but may require more computational resources due to the ensemble of trees. Parallelization can be used to speed up training.
- Decision Tree: Decision Trees are computationally efficient and scalable, making them suitable for large datasets.
- Linear Regression: Linear Regression is computationally efficient and scales well to large datasets, provided the assumptions hold.

4.6 Use Cases

- Random Forest: Random Forest is suitable for various tasks, including classification and regression, and is particularly well-suited for complex, non-linear problems.
- Decision Tree: Decision Trees are useful for tasks where interpretability is crucial, and the relationship between variables is relatively simple.
- Linear Regression: Linear Regression is appropriate for tasks where the relationship between variables is assumed to be linear, and interpretability is essential.

Table 4.1 Evaluation Metrics

SELECTED MODEL	RMSE SCORE	R-SQUARE
Linear Regression	483013587.3134137	-14.759850236179558
Decision Tree	7589.5932756326965	0.8906254539106669
Random Forest	5500.32296063104	0.938818461672514

The table 4.1 provided contains the RMSE (Root Mean Squared Error) scores and R-squared (R^2) values for three regression models: Linear Regression, Decision Tree, and Random Forest.

These metrics are used to assess the performance of regression models, and they provide insights into how well each model fits the data and makes predictions. Let's analyze the differences in these values and draw conclusions based on the results:

Linear Regression

- RMSE Score: 483,013,587.31
- R-squared (R^2) Value: -14.76
- Interpretation: The Linear Regression model has a very high RMSE score, indicating that it has a large average prediction error. Additionally, the negative R^2 value (-14.76) suggests that the model performs poorly and is likely not a good fit for the data. Negative R^2 values often occur when the model is a poor fit and performs worse than a simple horizontal line (the mean of the target variable).

Decision Tree

- RMSE Score: 7,589.59
- R-squared (R^2) Value: 0.891
- Interpretation: The Decision Tree model has a much lower RMSE score than Linear Regression, indicating better predictive accuracy. The positive R^2 value (0.891) indicates that the model explains a significant portion of the variance in the target variable, suggesting a good fit to the data. However, it's essential to consider whether the model may be overfitting the data.

Random Forest

- RMSE Score: 5,500.32
- R-squared (R^2) Value: 0.939
- Interpretation: The Random Forest model outperforms both Linear Regression and Decision Tree models. It has the lowest RMSE score, indicating the smallest prediction errors on average. The high positive R^2 value (0.939) suggests that the Random Forest model explains a substantial portion of the variance in the target variable and is a good fit to the data. It's a strong candidate for predictive modeling.

4.7 Model Interpretations

- Based on the provided results, the Random Forest model is the best-performing model among the three. It has the lowest RMSE and the highest R-squared value, indicating superior predictive accuracy and a better fit to the data.
- The Decision Tree model also performs well compared to Linear Regression, with a significantly lower RMSE and a positive R-squared value. However, assessing whether it may be overfitting the data is important.
- Linear Regression performs poorly in this context, with an extremely high RMSE and a negative R-squared value, indicating that it's unsuitable for the data.

- When choosing a model, it's essential to consider factors such as overfitting, interpretability, and computational complexity in addition to performance metrics. In this case, Random Forest appears to be the most promising model for further analysis and prediction tasks. Still, thorough model evaluation, validation, and fine-tuning should also be conducted to ensure its reliability and generalization to new data.

In conclusion, the choice of model depends on the specific characteristics of your data and your project goals. Random Forest tends to be a robust choice for achieving high accuracy, especially when the data is complex and non-linear. Decision Trees offer interpretability, making them valuable when transparency is essential. Linear Regression is appropriate when a linear relationship between variables is well-supported, and understanding feature impacts is critical.

CONCLUSION

In conclusion, our work on sales forecasting using machine learning and data analytics has revolutionized how companies predict market trends and client requests. We have gained important insights into previous sales patterns by thorough analysis, experimentation, and the use of sophisticated algorithms. We have also opened the path for more precise, effective, and pro-active sales forecasting approaches.

One of the most important lessons learned from this project is the effectiveness of predictive modeling in streamlining resource allocation, inventory management, and general company strategies. We may now confidently make judgments, reduce risks, and take advantage of new opportunities by utilizing the power of machine learning algorithms.

Furthermore, the integration of data analytics has provided a deeper understanding of customer behavior, enabling personalized marketing strategies and enhancing customer satisfaction. The synergy between machine learning and data analytics has not only streamlined the sales forecasting process but has also enhanced the overall operational efficiency of the organization.

This study has been both an academic venture and a real-world illustration of how machine learning and data analytics have the power to revolutionize the field of sales forecasting. Future attempts might build on the information and experience obtained via this project, promoting more research and invention in the areas of predictive analytics and business intelligence.

In conclusion, this project's journey reinforces the value of data-driven decision-making and highlights its function as a driver of organizational development and competitiveness. Businesses can confidently traverse the complexity of the market and make strategic decisions that promote success and long-term growth by embracing the potential of machine learning and data analytics.

REFERENCES

1. https://www.tutorialspoint.com/python/index.htm

2. https://www.geeksforgeeks.org/numpy-tutorial/

3. https://www.learndatasci.com/tutorials/python-pandas-tutorial-complete-introduction-for-beginners/

4. https://www.simplilearn.com/tutorials/python-tutorial/data-visualization-in-python

5. https://www.datacamp.com/tutorial/machine-learning-python

6. https://www.mygreatlearning.com/blog/seaborn-tutorial/

7. https://www.analyticsvidhya.com/blog/2021/10/everything-you-need-to-know-about-linear-regression/

8. https://blog.paperspace.com/decision-trees/

9. https://www.kaggle.com/code/pouryaayria/a-complete-ml-pipeline-tutorial-acu-86

10. https://towardsdatascience.com/understanding-random-forest-58381e0602d2