Region Wise Product Category Sales Prediction

BDA mini project submitted in partial fulfillment of the requirements
of the degree of

B. E. Computer Engineering

By

Samit Fernandes	**03**	**212031**
Jaden Franco	**04**	**212032**

Name of the Guide: Ms. Jayashri Mittal
Designation: Assistant Professor

Department of Computer Engineering
St. Francis Institute of Technology
(Engineering College)

An Autonomous Institute, Affiliated to University of Mumbai

2024-2025

CERTIFICATE

This is to certify that the mini project entitled "**Region Wise Product Category Sales Prediction"** is a bonafide work of **Samit Fernandes, 03 and Jaden Franco, 04** submitted to the University of Mumbai in partial fulfillment of the requirement for the BDA subject in final year of Computer Engineering.

Ms. Jayashri Mittal
Guide

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Samit Fernandes 03

Jaden Franco 04

Date:

Abstract

In today’s dynamic marketplace, understanding regional variations in product category sales is crucial for businesses aiming to optimize inventory management, enhance marketing strategies, and maximize revenue. This project presents a comprehensive approach to predicting sales based on geographic and categorical factors using a large dataset of sales records. Utilizing advanced machine learning techniques, particularly the Decision Tree Regression model, we analyze the impact of various factors such as state, product category, and month on sales performance. The dataset, sourced from Indian sales records, underwent rigorous preprocessing, including handling missing values, log transformation of sales figures, and one-hot encoding of categorical variables. The model's performance was evaluated using metrics such as Mean Squared Error (MSE) and R^2 score, demonstrating a robust predictive capability. Furthermore, we developed an interactive web application using Streamlit, allowing users to visualize sales trends and make informed predictions. This project not only aids businesses in forecasting sales more accurately but also serves as a valuable tool for data-driven decision-making in the retail sector.

Contents

List of Figures

List of Abbreviations

Sr. No.	Abbreviation	Expanded form
1	BDA	Big Data Analysis
2	MSE	Mean Squared Error
3	RMSE	Root Mean Squared Error
4	R^2	R-squared
5	B2B	Business-to-Business
6	CRM	Customer Relationship Management

Chapter 1
Introduction

1.1 Description

Sales prediction is a critical aspect of business operations, impacting inventory management, marketing strategies, and overall financial planning. Accurate sales forecasts enable companies to anticipate demand, optimize stock levels, and enhance customer satisfaction. The increasing complexity of market dynamics necessitates the development of sophisticated analytical tools that can provide insights into regional sales trends and product category performance. This project focuses on the Indian market, analyzing sales data to identify patterns and predict future sales across different states and product categories.

1.2 Problem Formulation

Despite the availability of extensive sales data, many businesses struggle to leverage this information effectively for predictive analytics. The primary challenges include:

- Data Complexity: Sales data is often influenced by multiple variables, including geographical location, seasonal trends, and consumer behavior.
- Lack of Predictive Models: Many existing models do not adequately account for regional variations or the specific nuances of product categories, leading to inaccurate forecasts.
- Data Quality Issues: Missing values and data inconsistencies can significantly impact the reliability of sales predictions.
- To address these challenges, this project aims to develop a predictive model that accurately forecasts sales at a regional level, incorporating various product categories and temporal factors.

1.3 Motivation

The motivation behind this project stems from the increasing need for businesses to adapt to rapidly changing market conditions. As competition intensifies, organizations must leverage data analytics to make informed decisions and maintain a competitive edge. By providing accurate sales predictions, businesses can optimize their inventory, reduce operational costs, and improve overall profitability. Additionally, understanding regional sales dynamics enables companies to tailor their marketing strategies and product offerings, thereby enhancing customer satisfaction and driving sales growth.

1.4 Proposed Solution

The proposed solution involves the following key components:

1. Data Collection: Gathering historical sales data across various states and product categories.
2. Data Preprocessing: Cleaning and transforming the dataset to address missing values and ensure data consistency. This includes log transformation of the sales figures to normalize the distribution.
3. Feature Engineering: Utilizing one-hot encoding for categorical variables to facilitate model training.
4. Model Development: Implementing a Decision Tree Regression model to predict sales based on the identified features.
5. Model Evaluation: Assessing the model's performance using metrics such as Mean Squared Error (MSE) and R-squared (R^2).
6. Visualization: Developing an interactive web application using Streamlit for users to visualize sales trends and make predictions based on user inputs.

1.5 Scope of The Project

The scope of this project encompasses the following aspects:

- Regional Focus: The analysis will concentrate on sales data from different states in India, allowing for a nuanced understanding of regional market dynamics.

- Product Categories: The project will include various product categories, providing insights into category-specific sales trends and performance.
- Temporal Analysis: Sales predictions will be made on a monthly basis, considering seasonal effects and other time-related factors.
- User Interaction: The developed web application will enable users to upload sales data, predict future sales, and visualize key insights, making it accessible to stakeholders with varying levels of technical expertise.
- Limitations: While the project aims to provide accurate predictions, it will acknowledge the inherent limitations of machine learning models, including data quality issues and potential overfitting.

Chapter 2
Review of Literature

The paper "Predicting and Defining B2B Sales Success with Machine Learning" investigates how machine learning can predict sales success for a Fortune 500 paper and packaging company. The research aimed to identify key factors influencing sales and develop an accurate predictive model to improve sales outcomes.Using data from the company's Salesforce.com CRM system, the study tested several models, with the random forest model achieving the highest accuracy. This model identified important variables like opportunity duration and task count, which significantly influence sales success.While the study advances predictive sales modeling, a key limitation is the reliance on CRM data quality. Inconsistent or incomplete data entry by sales teams can reduce the accuracy of predictions, highlighting the need for improved data practices within the company. Despite this, the study provides valuable insights into enhancing B2B sales forecasting. [1]

The paper "Sales Prediction based on Machine Learning" by Zixuan Huo evaluates different predictive models to forecast e-commerce sales using Walmart's sales data. The study tests two linear models, three machine learning models, and two deep learning models to predict daily sales over the next 28 days.The research finds that while adding information like price and calendar data improves model performance, complex machine learning and deep learning models do not significantly outperform simpler models such as linear regression. The study highlights that machine learning and deep learning models might not have an advantage in sales prediction for this dataset.A limitation of the study is the relatively small dataset, which may affect the generalizability of the results. The performance of machine learning models on larger datasets could vary, and this aspect is suggested for future research. Despite this, the study provides valuable insights into improving sales predictions in e-commerce. [2]

Chapter 3

System Analysis

3.1 Functional Requirements

The functional requirements of the "Region Wise Product Category Sales Prediction" system outline the essential features and functionalities that must be implemented to meet user needs effectively. These include:

1. Data Input:The system must allow users to upload historical sales data in a specified format (e.g., CSV). The system must validate the uploaded data for completeness and correctness.
2. Data Processing:The system must preprocess the uploaded data by handling missing values and performing necessary transformations (e.g., log transformation). The system must apply one-hot encoding to categorical variables for model compatibility.
3. Sales Prediction:The system must provide users with the capability to input specific parameters (e.g., region, product category, month) to generate sales predictions. The system must utilize a Decision Tree Regression model to forecast sales based on the input parameters.
4. Visualization:The system must present the prediction results in a user-friendly format, including graphical representations of sales trends. The system must allow users to visualize historical sales data and predicted sales side by side.
5. User Interface:The system must provide a simple and intuitive web interface for user interactions. The system must include help and guidance sections to assist users in navigating the application.
6. Reporting: The system must generate reports summarizing the prediction results and insights drawn from the analysis.

3.2 Non Functional Requirements:

Non-functional requirements define the quality attributes of the system that are crucial for its usability, performance, and reliability. The following non-functional requirements are identified for this project:

3.2.1 Performance Requirements

1. Response Time: The system should provide predictions within 5 seconds of input submission, ensuring a responsive user experience.
2. Scalability: The system must handle datasets with up to 100,000 records without significant performance degradation.
3. Availability: The system should be operational 24/7, with a target uptime of 99.9%.

3.2.2 Software Quality Attributes

1. Usability: The user interface must be designed for ease of use, allowing users with minimal technical expertise to navigate and utilize the system effectively.
2. Reliability: The system should consistently produce accurate sales predictions, with a target accuracy rate of at least 85%.
3. Maintainability: The codebase must be well-documented and modular, allowing for easy updates and enhancements in the future.
4. Security: The system must implement security measures to protect user data and ensure that sensitive information is not exposed.

3.3 Specific Requirements:

Hardware :

1. Server Requirements:

   Processor: Minimum quad-core processor (Intel i5 or equivalent).

   RAM: Minimum of 16 GB RAM.

   Storage: At least 500 GB SSD for data storage and application deployment.

   Network: High-speed internet connection for data access and user interaction.

2. User Device Requirements:

   Any device (desktop, laptop, tablet) capable of running a modern web browser.

Software :

1. Operating System:Server should run on modern operating system (eg, Windows Server)
2. Programming Languages:Python (for model development and data processing).
3. Frameworks and Libraries:
   a. Streamlit (for creating the web application interface).
   b. Pandas (for data manipulation).
   c. Scikit-learn (for implementing the machine learning model).
   d. Matplotlib or Plotly (for data visualization).
4. Development Tools:Integrated Development Environment like PyCharm or VS Code.

3.4 Use-Case Diagrams and description

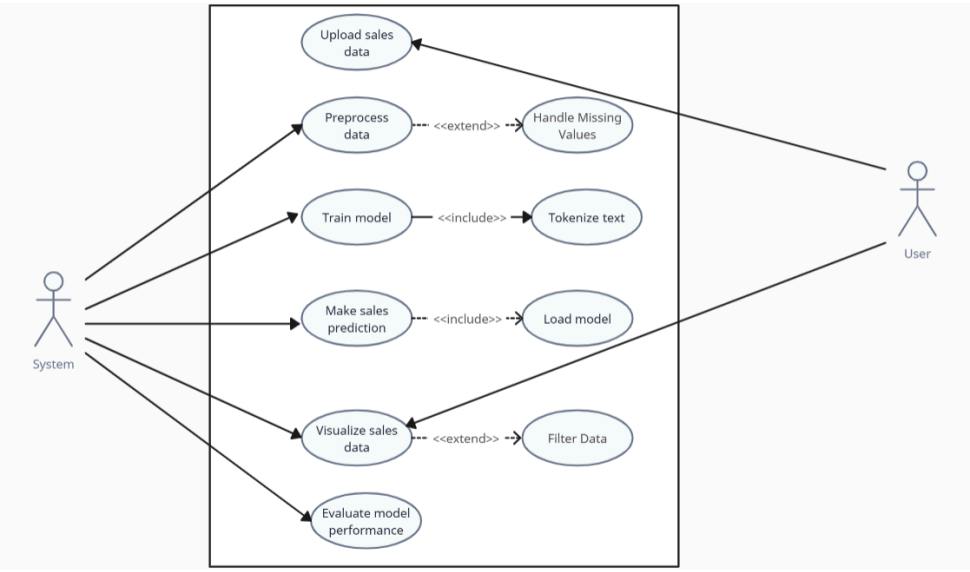

Figure 1: Use case diagram for sales prediction

The use case diagram for the sales prediction system includes two actors: User and System. The User uploads the dataset, selects the criteria for sales prediction (such as state, product category, and month), and views the predicted results. The System handles dataset preprocessing (with tasks like encoding and scaling), performs sales predictions using the trained model, and optionally visualizes the data.

Chapter 4

Analysis Modeling

4.1 Class Diagram

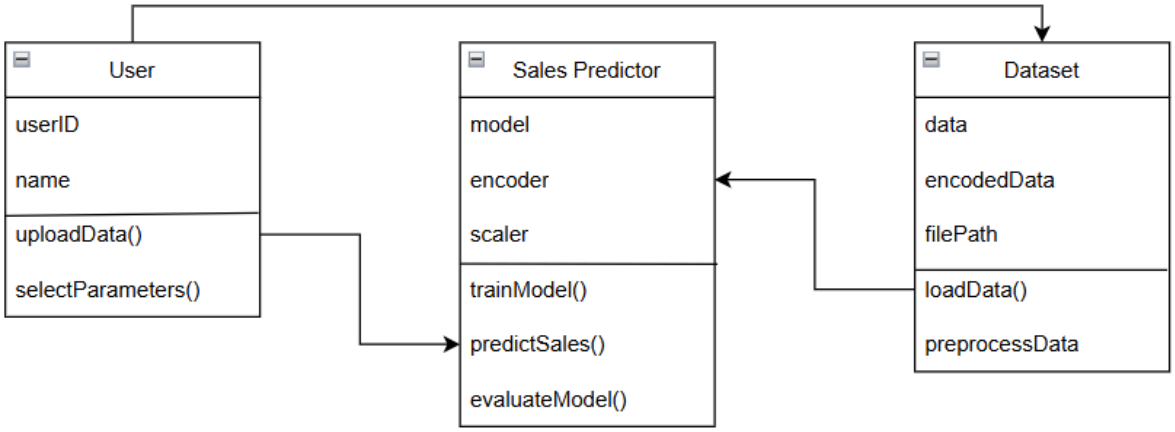

Figure 2: Class diagram for sales prediction

The class diagram for the sales prediction system consists of three main classes: User, SalesPredictor, and Dataset. The User class represents the individual interacting with the system and has attributes like userID and role, along with methods to upload data and request predictions. The Dataset class holds the sales data, with attributes like state, productCategory, month, and sales, and methods to preprocess, clean, and transform the data (like handling missing values and applying one-hot encoding). The SalesPredictor class handles the core functionality of sales prediction, utilizing attributes like model, encoder, and scaler. It has methods for loading models, predicting sales, and evaluating results.

4.2 Functional Modeling

Data Flow Diagram

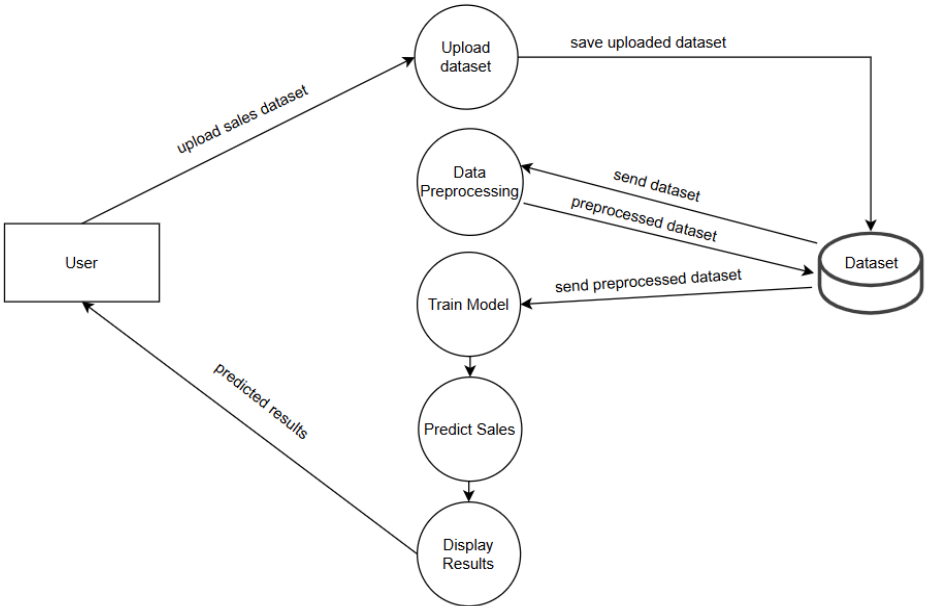

Figure 3: Data Flow Diagram for sales prediction

The data flow diagram represents the process for a region-wise sales prediction project. It begins with the user uploading a sales dataset, which is saved and sent to a data preprocessing module to clean and prepare the data for analysis. Once the dataset is preprocessed, it is used to train a machine learning model, specifically using a Decision Tree Regressor, which learns from the data to predict future sales. The trained model generates sales predictions, focusing on different regions, and the results are displayed back to the user, providing insights into future sales trends.

Chapter 5

Design

5.1 Architectural Design

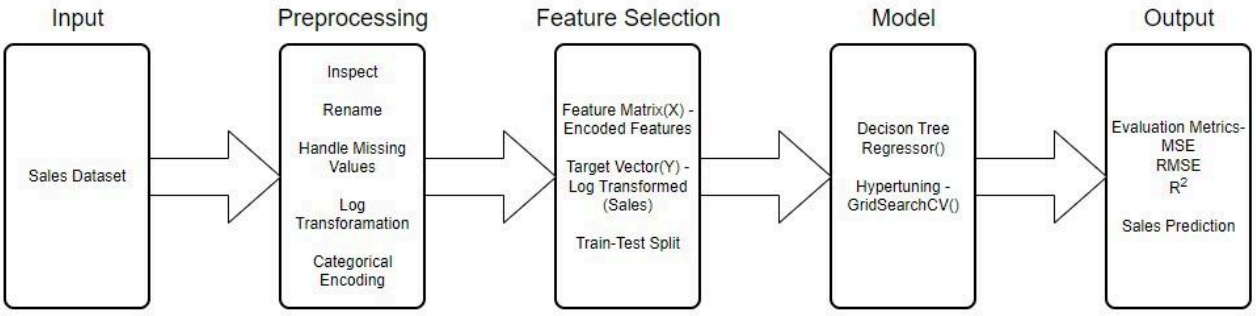

Figure 4: Architectural diagram for sales prediction

The architecture for the "Region Wise Product Category Sales Prediction" project outlines a streamlined process. Starting with a Sales Dataset as input, the data undergoes Preprocessing steps such as inspection, renaming columns, handling missing values, applying log transformation, and encoding categorical features. In the Feature Selection phase, relevant features (X) and the target variable (Y) are identified and the dataset is split into training and testing sets. The Model used is a Decision Tree Regressor with hyperparameter tuning performed using GridSearchCV. Finally, the Output includes evaluation metrics like MSE, RMSE, and R^2, leading to accurate sales predictions for the given regions and product categories. This architecture provides an effective framework for sales forecasting.

5.2 User Interface Design

The user interface (UI) design for the "Region Wise Product Category Sales Prediction" project emphasizes usability, accessibility, and aesthetic appeal to enhance user engagement and experience. The application is developed using Streamlit, which provides a streamlined framework for building interactive web applications with minimal effort. The design focuses

on presenting critical information in an intuitive layout, allowing users to navigate through various functionalities seamlessly.

The main components of the user interface include:

Dashboard Overview: The landing page features an overview of sales data visualizations, showcasing total sales by state, product category distributions, and trends over time. This section provides users with immediate insights into sales performance.

Data Upload Functionality: Users can easily upload their sales data through a dedicated file uploader. The interface guides users on the accepted file format, ensuring smooth data input and preprocessing.

Sales Prediction Section: A clear and concise form allows users to input parameters such as state, product category, and month for which they wish to predict sales. This section includes dropdown menus that auto-populate based on the uploaded data, reducing the risk of user error.

Visualization Tools: Interactive charts and graphs are integrated into the UI to present sales trends and comparisons effectively. Users can highlight specific states, product categories, or months to focus their analysis, making the interface more dynamic and informative.

Predictive Results Display: Upon clicking the "Predict" button, users receive instant feedback on estimated sales figures, enhancing the application's responsiveness. This section is designed to clearly display the predicted sales along with relevant metrics, fostering user understanding.

Reset Functionality: A reset button is included to allow users to clear inputs and start fresh without having to reload the application. This feature enhances usability by providing an efficient way to explore different scenarios.

Overall, the UI design prioritizes clarity and interactivity, ensuring that users can leverage the analytical capabilities of the application without facing barriers. By adopting a user-centered approach, the design effectively meets the needs of stakeholders in the retail sector, empowering them to make informed decisions based on data-driven insights.

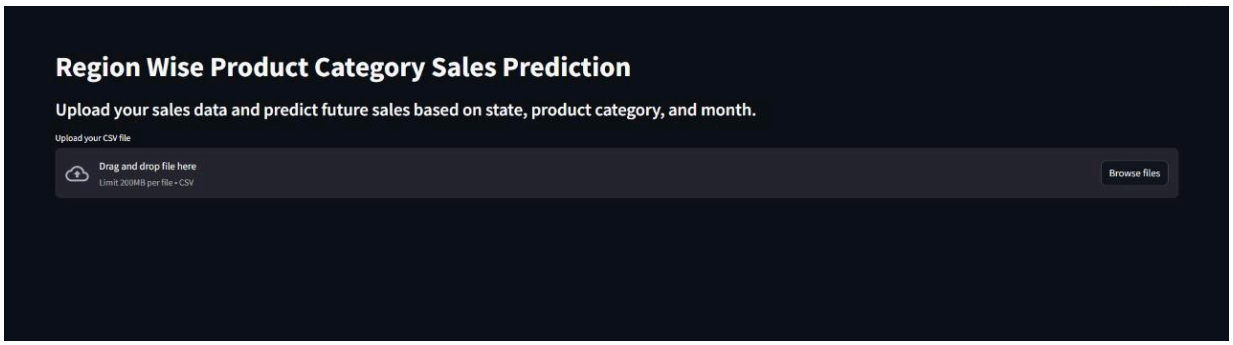

Figure 5: Home page for sales prediction

The home page of website where users are introduced to the system and its features, offering options for uploading sales data and predicting sales.

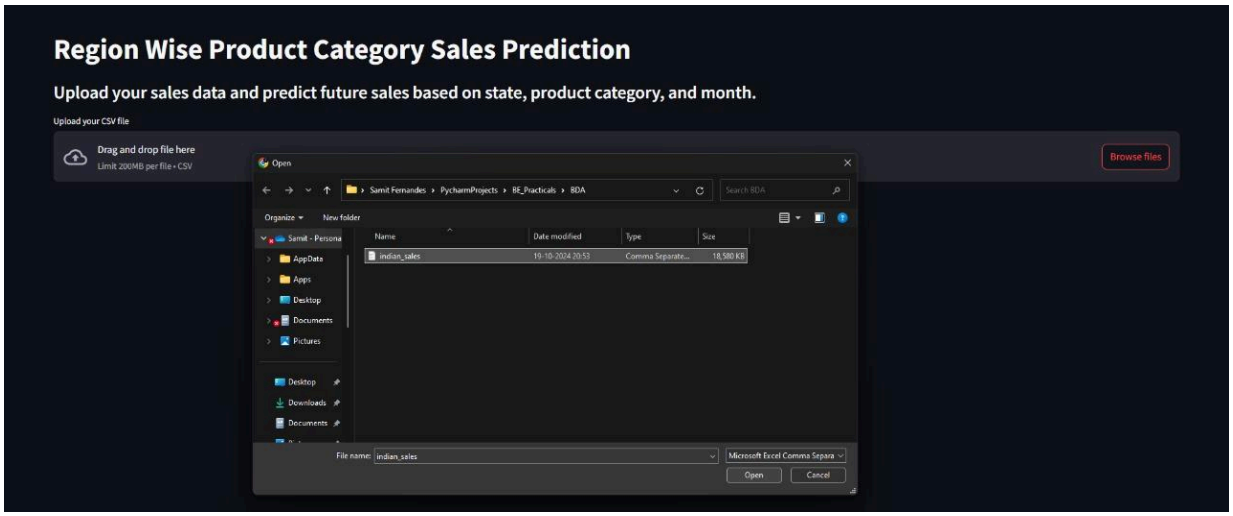

Figure 6: Upload sales dataset

Users can upload their sales dataset in CSV format, providing necessary data for predictions.

Figure 7: Sales dataset uploaded

Displays the dataset has been successfully uploaded, showing key information about the data.

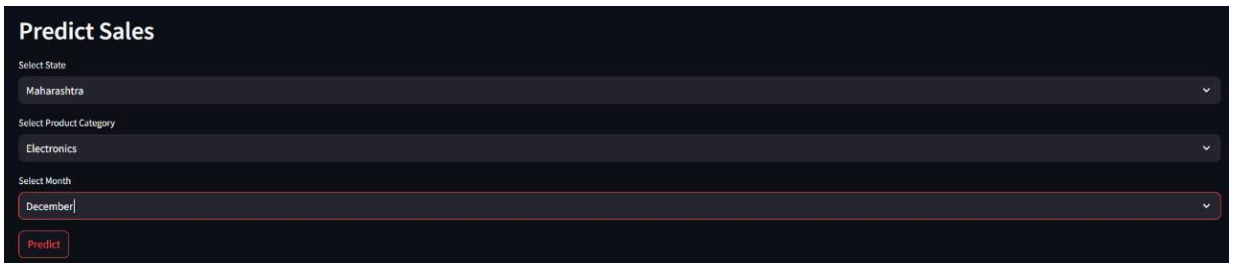

Figure 8: Select parameters for sales prediction

Users can choose specific parameters such as region, product category, and month to generate sales predictions.

Figure 9: Predicted estimate sales result

It displays predicted sales value for the selected parameters, showing an estimated sales output.

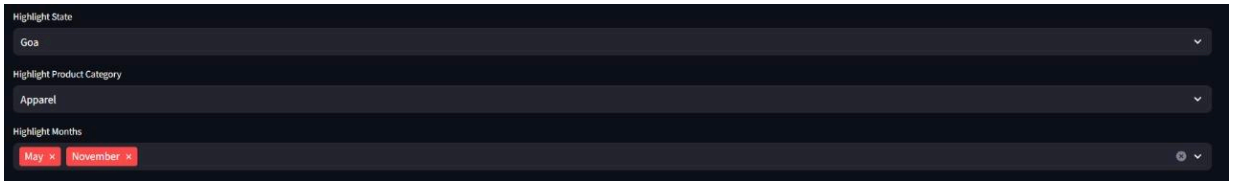

Figure 10: Select Region and product category to visualize data

This interface allows users to filter sales data by selecting specific region(Goa) and product category(Apparel).

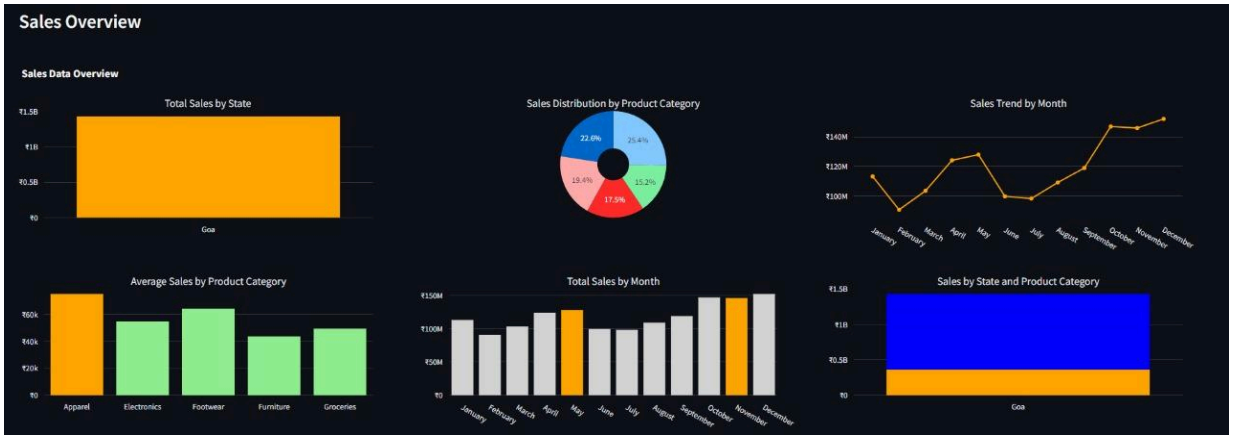

Figure 11: Visualize sales dataset for selected parameter(Apparel)

A page that offers various visualizations of the uploaded sales dataset, allowing users to explore trends and patterns.

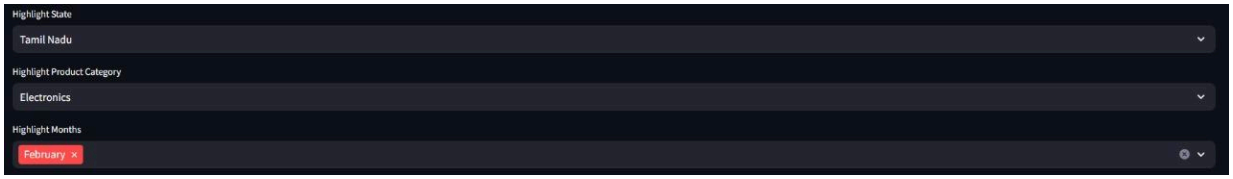

Figure 12: Select Region(Tamil Nadu) and product category to visualize data

This interface allows users to filter sales data by selecting specific region(Tamil Nadu) and product category(Electronics).

Figure 13: Visualize sales dataset for selected parameter(Electronics)

A page that offers various visualizations of the uploaded sales dataset, allowing users to explore trends and patterns.

Chapter 6

Implementation

6.1 Algorithm

Decision Tree Regression:

A Decision Tree Regressor is a non-linear machine learning algorithm used for predicting continuous values (sales data). It works by splitting the data into smaller and smaller subsets based on the input features, creating a tree-like model where each internal node represents a decision rule, and each leaf node represents the predicted outcome.

Working of Decision Tree Regression:

1. Data Splitting: At each node, the algorithm evaluates different features (e.g., state, product category, month) and creates a split based on a condition that minimizes the prediction error. The tree continues to grow by splitting the dataset into smaller regions at each step.
2. Prediction: For a given input, the algorithm traverses the decision tree by applying the learned rules (e.g., if the feature value is less than or greater than a threshold). It reaches a leaf node, where the average of the target variable (log-transformed sales) in that region is returned as the prediction.

6.2 Working Of The Project

The implementation of the "Region Wise Product Category Sales Prediction" project consists of several key steps that encompass data preprocessing, model training, and the development of a user interface using Streamlit. Below is a detailed breakdown of each step along with the corresponding code segments:

Step 1: Load and Preprocess Data

```
#Import Libraries
import pandas as pd
import numpy as np
```

```
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error, r2_score
import joblib

#Load Dataset

data = pd.read_csv("indian_sales.csv")
print(data.info())
data.rename(columns={"Sales (₹)": "Sales"}, inplace=True)
data.fillna(0, inplace=True)

#Transform Sales Data

data["log_Sales"] = np.log(data["Sales"] + 1)

#One-Hot Encode Categorical Variables

categorical_cols = ["State", "Product_Category", "Month"]
encoder = OneHotEncoder(sparse_output=False, drop='first')
encoded_features = encoder.fit_transform(data[categorical_cols])
encoded_df = pd.DataFrame(encoded_features,
columns=encoder.get_feature_names_out(categorical_cols))
data = pd.concat([data, encoded_df], axis=1)

#Define Features and Target Variable

feature_columns = encoded_df.columns.tolist()
X = data[feature_columns]
y = data["log_Sales"]
```

Step 2: Split the Data and Scale Features

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1234)


scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(X_train)

X_test_scaled = scaler.transform(X_test)
```

Step 3: Train the Model

```
dt = DecisionTreeRegressor()
param_grid = {
    'max_depth': [5, 10, 15],
    'min_samples_split': [2, 4, 8]
}
```

```
grid_search = GridSearchCV(estimator=dt, param_grid=param_grid, cv=3,
scoring='neg_root_mean_squared_error')
grid_search.fit(X_train_scaled, y_train)

best_model = grid_search.best_estimator_
```

Step 4: Make Predictions and Evaluate the Model

```
predictions = best_model.predict(X_test_scaled)

mse = mean_squared_error(y_test, predictions)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, predictions)
print("Mean Squared Error (MSE):", mse)
print("Root Mean Squared Error (RMSE):", rmse)
print("R²:", r2)

#Save the Model
joblib.dump(best_model, 'model.pkl')
joblib.dump(encoder, 'encoder.pkl')
joblib.dump(scaler, 'scaler.pkl')
print("Model, encoder, and scaler saved as model.pkl, encoder.pkl, and scaler.pkl")
```

Step 5: Develop User Interface with Streamlit

```
import streamlit as st
import joblib
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots

#Load and Preprocess User Data
def load_and_preprocess_data(file):
    data = pd.read_csv(file)
    data.rename(columns={"Sales (₹)": "Sales"}, inplace=True)
    data.fillna(0, inplace=True)
    month_order = ["January", "February", "March", "April", "May", "June",
                   "July", "August", "September", "October", "November", "December"]
    data["Month"] = pd.Categorical(data["Month"], categories=month_order, ordered=True)
    return data

#Create Visualizations
def create_charts(data, selected_state=None, selected_product_category=None,
selected_months=None):

#Sales Prediction Function
def predict_sales(model, encoder, scaler, state, product_category, month):

#User Interface
```

```
st.set_page_config(page_title="Sales Prediction App", layout="wide")
st.title("Sales Prediction App")
uploaded_file = st.file_uploader("Upload your CSV file", type=["csv"])
if uploaded_file is not None:
    # Load and preprocess data, input selections, and prediction logic
```

This step-by-step implementation details the entire process, from data loading and preprocessing to model training and creating an interactive web application, enabling users to predict sales based on various input parameters and visualize the results effectively.

Chapter 7
Conclusion

In this chapter, we reviewed the implementation of the "Region Wise Product Category Sales Prediction" project within the context of big data analysis. By leveraging a comprehensive dataset and applying robust preprocessing techniques, including one-hot encoding and handling missing values, we effectively prepared the data for modeling. The decision tree regression model, optimized through GridSearchCV, demonstrated strong predictive capabilities, with evaluation metrics such as RMSE and R^2 score affirming its performance in capturing complex sales patterns across different regions and product categories.

The incorporation of a Streamlit application facilitated an interactive user experience, allowing stakeholders to visualize sales trends and make informed predictions in real time. This project not only highlights the power of big data analytics in enhancing decision-making processes within the retail sector but also sets the stage for future enhancements, such as exploring advanced modeling techniques and incorporating additional data sources. Overall, the insights gained from this analysis could significantly impact strategic planning and operational efficiency in sales management.

References

[1] S. Mortensen, M. Christison, B. Li, A. Zhu and R. Venkatesan, "Predicting and Defining B2B Sales Success with Machine Learning," 2019 Systems and Information Engineering Design Symposium (SIEDS), Charlottesville, VA, USA, 2019, pp. 1-5, doi: 10.1109/SIEDS.2019.8735638.

[2] Z. Huo, "Sales Prediction based on Machine Learning," 2021 2nd International Conference on E-Commerce and Internet Technology (ECIT), Hangzhou, China, 2021, pp. 410-415, doi: 10.1109/ECIT52743.2021.00093.

[3] S. Cheriyan, S. Ibrahim, S. Mohanan and S. Treesa, "Intelligent Sales Prediction Using Machine Learning Techniques," 2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE), Southend, UK, 2018, pp. 53-58, doi: 10.1109/iCCECOME.2018.8659115.

Acknowledgements

A project is always a coordinated, guided and scheduled team effort aimed at realizing a common goal. We are grateful and gracious to all those people who have helped and guided us through this project and make this experience worthwhile. We wish to sincerely thank our Principal Dr. Sincy George and our CMPN HOD Dr. Kavita Sonawane for giving us this opportunity to prepare a project in the Final Year of Computer Engineering. We are highly indebted to our institute, St. Francis Institute of Technology and the Department of Computer Engineering for providing us with this learning opportunity with the required resources to accomplish our task so far. We would also like to express our deep gratitude to our assigned mentor, Ms. Jayashri Mittal, who constantly guided and supervised us, and also furnished essential information concerning the project. This work would not have been possible without her necessary insights and intellectual suggestions that have helped us achieve so much. We would like to thank our teacher Ms. Jayashri Mittal who approved the topic for our BDA mini project and gave us some valuable suggestions to make our project better. We also take the opportunity to thank all teaching and non-teaching staff for their endearing support and cooperation.