FUTURE SALES PREDICTION

Phase 5 submission document

Project Title: Future Sales Prediction

Phase 5: Project Documentation & Submission

Topic: In this phase you will document your project and prepare it for submission.

Future Sales Prediction

Problem Statement:

The problem is to create a predictive model for a retail company that can forecast future sales based on historical sales data. The goal is to help the company optimize its inventory management and make informed business decisions. This involves analyzing past sales trends, understanding the factors that influence sales, and using this information to predict future sales accurately.

Introduction:

- Predicting future sales is the heartbeat of successful business strategies. It involves the art and science of using historical data, market trends, and advanced analytical tools to forecast the demand for products or services in the days, months, or years to come.
- Sales prediction is not merely a crystal ball gazing exercise; it's a process that harnesses the power of data analytics, machine learning, and statistical modeling to unravel patterns and insights hidden within vast amounts of information. By analyzing past sales performance, consumer behavior, economic indicators, and various influencing factors, businesses can anticipate market trends and consumer demands, aiding in making well-informed decisions.

- The accuracy of future sales prediction can significantly impact inventory management, resource allocation, and the overall success of a business. In today's digital age, organizations are increasingly relying on sophisticated predictive models and algorithms to make proactive and data-driven decisions, allowing them to adapt swiftly to changing market conditions and stay ahead in the competitive landscape.

- However, the challenge lies in navigating uncertainties and unexpected variables that can influence sales trends. Future sales prediction requires a

balance between data-driven insights and an understanding of the dynamic nature of markets, consumer behavior, and external influences.

- In essence, the ability to predict future sales is a cornerstone of strategic planning, enabling businesses to steer operations, marketing, and production in the right direction. As technology continues to evolve, refining predictive models becomes an ongoing journey toward greater accuracy and adaptability, ensuring businesses stay agile and responsive in an ever-changing market.

Data source:

Dataset link:

[Amazon Top 50 Bestselling Books 2009 - 2022 (kaggle.com)]

Dataset:

	A	B	C	D	E	F	G	H
1	Name	Author	User Ratin	Reviews	Price	Year	Genre	
2	Act Like a	Steve Har	4.6	5013	17	2009	Non Fiction	
3	Arguing w	Glenn Bec	4.6	798	5	2009	Non Fiction	
4	Breaking [	Stephenie	4.6	9769	13	2009	Fiction	
5	Crazy Love	Francis Ch	4.7	1542	14	2009	Non Fiction	
6	Dead And	Charlaine	4.6	1541	4	2009	Fiction	
7	Diary of a	Jeff Kinne	4.8	3837	15	2009	Fiction	
8	Divine Sou	Zhi Gang S	4.6	37	6	2009	Non Fiction	
9	Dog Days (	Jeff Kinne	4.8	3181	12	2009	Fiction	
10	Eat This N	David Zinc	4.5	720	1	2009	Non Fiction	
11	Eat This, N	David Zinc	4.3	956	14	2009	Non Fiction	
12	Eclipse (T	Stephenie	4.7	5505	7	2009	Fiction	
13	Eclipse (T	Stephenie	4.7	5505	18	2009	Fiction	
14	Glenn Bec	Glenn Bec	4.6	1365	11	2009	Non Fiction	
15	Going Rog	Sarah Pali	4.6	1636	6	2009	Non Fiction	
16	Good to G	Jim Collin	4.5	3457	14	2009	Non Fiction	
17	Have a Lit	Mitch Albo	4.8	1930	4	2009	Non Fiction	
18	I, Alex Cro	James Pat	4.6	1320	7	2009	Fiction	
19	Liberty an	Mark R. Le	4.8	3828	15	2009	Non Fiction	
20	Mastering	Julia Child	4.8	2926	27	2009	Non Fiction	
21	New Moo	Stephenie	4.6	5680	10	2009	Fiction	
22	Olive Kitte	Elizabeth	4.2	4519	12	2009	Fiction	
23	Outliers: T	Malcolm G	4.6	10426	20	2009	Non Fiction	
24	Publicatio	American	4.5	8580	46	2009	Non Fiction	
25	Sookie Sta	Charlaine	4.7	973	25	2009	Fiction	
26	Strengths	Gallup	4	5069	17	2009	Non Fiction	
27	Super Fre	Steven D.	4.5	1583	18	2009	Non Fiction	
28	The 5000 Y	W. Cleon	4.8	1680	12	2009	Non Fiction	

Here's is a list of tools and software commonly used in the process:

1. **Programming Language:**
   Python is the most popular language for machine learning due to its extensive libraries and frameworks. You can use libraries like Numpy, Pandas, scikit-learn, and more
2. **Integrated Development Environment(IDE):**
   Choose an IDE for coding and running machine learning experiments. Some popular options include Jupyter Notebook, Google Colab, or traditional IDEs like PyCharm.

3. Machine Learning Libraries:

- ✓ You'll need various machine learning libraries, including:
- ✓ Scikit-learn for building and evaluating machine learning models.
- ✓ TensorFlow or PyTorch for deep learning, if needed.
- ✓ XGBoost, LightGBM, or CatBoost for gradient boosting models.

4. Data Visualization Tools:

Tools like Matplotlib, Seaborn, or Plotly are essential for data

exploration and visualization.

5. Data Preprocessing Tools:

Libraries like pandas help with data cleaning, manipulation, and preprocessing.

6. Data Collection and Storage:

Depending on your data source, you might need web scraping tools (e.g., BeautifulSoup or Scrapy) or databases (e.g., SQLite, PostgreSQL) for data storage.

7. Version Control:

Version control systems like Git are valuable for tracking changes in your code and collaborating with others.

8. Notebooks and Documentation:

Tools for documenting your work, such as Jupyter Notebooks or Markdown for creating README files and documentation.

1. DESIGN THINKING AND PRESENT IN FORM OF DOCUMENT

1. Empathize:

- Begin by understanding the retail company's specific pain points and objectives related to inventory management and sales forecasting.
- Conduct interviews or surveys with stakeholders, including inventory managers, sales teams, and decision-makers, to gather their insights and requirements.
- Explore the challenges they face in managing inventory efficiently and making data-driven decisions.

2. Define:

- Clearly define the problem statement based on the insights gathered. For example, "Develop a sales forecasting system to predict monthly sales for each product category in order to reduce overstock and understock situations."
- Identify the key performance indicators (KPIs) that will measure the success of the solution, such as inventory turnover rate or forecast accuracy.

3. Ideate:

- Brainstorm potential solutions and approaches with a cross-functional team.
- Consider the types of data needed, such as historical sales data, product attributes, external factors (e.g., holidays, promotions), and any relevant market data.
- Explore various machine learning and forecasting models that could be suitable for the task.

4. Prototype:

- Create a small-scale prototype or proof of concept to test the feasibility of the chosen approach.
- Use a subset of historical data to build an initial model and evaluate its performance.
- Gather feedback from stakeholders on the prototype to refine the approach.

5. Test:

- Conduct thorough testing of the model using a validation dataset to assess its accuracy and reliability.

- Evaluate different models and algorithms to determine which one performs best for the specific business problem.
- Iterate on the model design based on test results and feedback.

6. Implement:

- Develop a full-scale solution that includes data pipelines, model training, and a user interface for accessing predictions.
- Ensure that the system is capable of handling large volumes of historical and real-time data efficiently.
- Deploy the system in a production environment, taking into account scalability and security considerations.

7. Feedback and Iterate:

- Continuously collect feedback from end-users, inventory managers, and decision-makers regarding the accuracy and usefulness of the sales forecasts.
- Monitor the system's performance in real-world scenarios and address any issues promptly.
- Periodically retrain the model with new data to keep it up to date and improve accuracy.

8. Scale and Optimize:

- As the system proves its value, consider scaling it to cover more product categories or regions within the company.
- Optimize the system's efficiency and cost-effectiveness over time by fine-tuning algorithms and data processing pipelines.

9. Educate and Train:

- Provide training to relevant staff members on how to use the sales forecasting system effectively.
- Educate decision-makers on how to interpret and act upon the predictions to optimize inventory management.

10. Celebrate Success:

- Acknowledge and celebrate the successes achieved through the implementation of the sales forecasting system, such as reduced inventory costs, improved product availability, and data-driven decision-making.

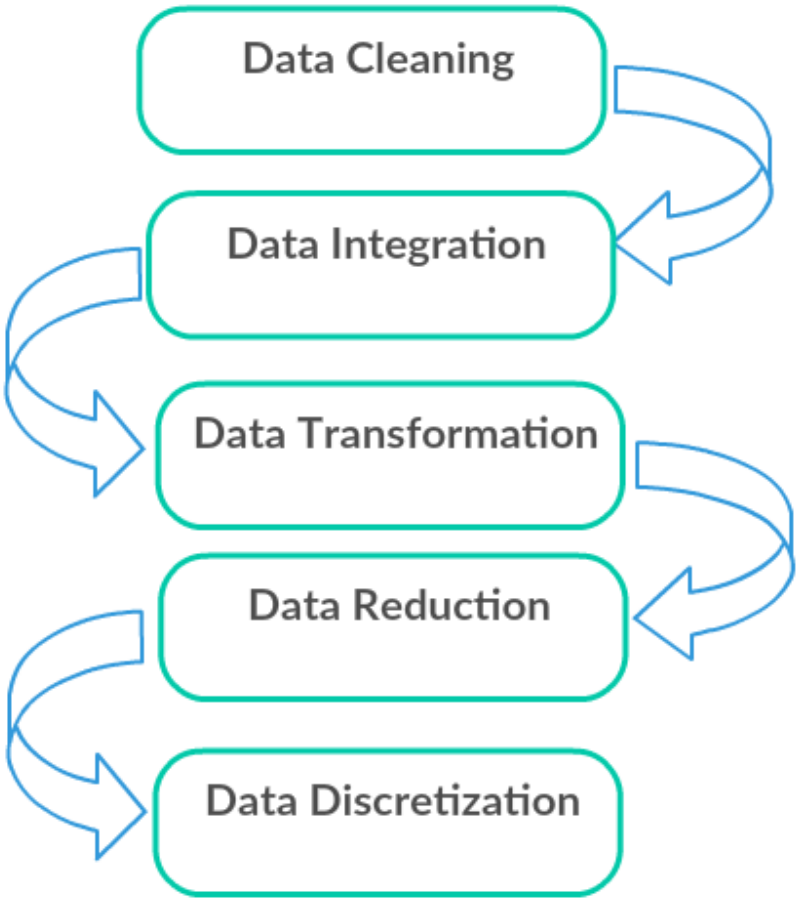
3. DESIGN INTO INNOVATION

1. Data Collection:

Data collection is a critical step in creating accurate predictive models for future sales. To predict future sales, you'll need various types of data that can offer insights into historical sales patterns, market trends, customer behaviour, and other relevant factors. Here are some crucial data collection areas:

2. Data Preprocessing:

Clean the data by handling missing values, outliers, and encoding categorical variables. Standardize or normalize numerical features as necessary



Python Program:

```
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv("/content/bestsellers with categories.csv")
df.describe()

 Check for missing values
print(df.isnull().sum())

 Check the statistics of the DataFrame

 Drop the 'User Rating' column from 'x'
x = df.drop('User Rating', axis=1)

 Correct the syntax to drop the 'Reviews' column from 'y'
y = df['Reviews']

 Fill missing values in 'x' using forward fill (ffill)
x.fillna(method='ffill', inplace=True)

 Apply one-hot encoding to categorical columns in 'x'
x = pd.get_dummies(x, columns=['Author', 'Year', 'Genre'])
print(x)

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

 Identify numeric columns for scaling
numeric_columns = x.select_dtypes(include= ['int64',
'float64']).columns

 Split the data into training and testing sets
x_train, x_test, y_train, y_test = train_test_split(x, y,
test_size=0.2, random_state=42)

 Initialize the StandardScaler
scaler = StandardScaler()

 Fit and transform the training data for numeric columns
x_train[numeric_columns] =
scaler.fit_transform(x_train[numeric_columns])

 Transform the testing data for numeric columns using the same scaler
```

```
x_test[numeric_columns] = scaler.transform(x_test[numeric_columns])

features = ['Name', 'Author', 'User Ratings', 'Reviews', 'Price',
'Year', 'Genre']
x_train_df = pd.DataFrame(x_train, columns=features)
y_train_df = pd.DataFrame({'Reviews': y_train})
print("pre-processed Data:")
print(x_train_df.head())
print("\n Target Data:")
print(y_train_df.head())

import matplotlib.pyplot as plt
import seaborn as sns
data = pd.read_csv('/content/bestsellers with categories.csv')
summary_stats = data.describe()
print("Summary Statistics:")
print(summary_stats)
coorelation_matrix = data.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(coorelation_matrix, annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap")
plt.show()
plt.figure(figsize=(8,6))

sns.histplot(data['Year'],kde=True,bins=30)
plt.title("Distribution of the publication year")
plt.xlabel("publication year")
plt.ylabel("Count")
plt.show()
plt.figure(figsize=(10,6))

sns.countplot(x='Genre', data=data,
order=data['Genre'].value_counts().index)
plt.title("Count of Books by Genre") Correct the parameter name
'rotation'
plt.xticks(rotation=90)
plt.show()

plt.figure(figsize=(2,3))
sns.catplot(x='Genre', y='User Rating', hue='Author', kind='strip',
data=data)
plt.xticks(rotation=90) Rotate x-axis labels for better readability
plt.show()

plt.figure(figsize=(10,8))
sns.boxplot(x='Genre',y='Price',data=data)
plt.title("Book price by Genre")
plt.show()
```

Output:

Checking missing values

Name 0

Author 0

User Rating 0

Reviews 0

Price 0

Year 0

Genre 0

dtype: int64

checking for outliers:

	Name	Reviews	Price
0	*Act Like a Lady, Think Like a Man: What Men Re...*	*5013*	*17*
1	*Arguing with Idiots: How to Stop Small Minds a...*	*798*	*5*
2	*Breaking Dawn (The Twilight Saga, Book 4)*	*9769*	*13*
3	*Crazy Love: Overwhelmed by a Relentless God*	*1542*	*14*
4	*Dead And Gone: A Sookie Stackhouse Novel (Sook...*	*1541*	*4*
..	*...*	*...*	*...*
695	*The Wonderful Things You Will Be*	*20920*	*9*
696	*Ugly Love: A Novel*	*33929*	*10*
697	*Verity*	*71826*	*11*
698	*What to Expect When You're Expecting*	*27052*	*13*
699	*Where the Crawdads Sing*	*208917*	*10*

	Author_Abraham Verghese	Author_Adam Gasiewski	Author_Adam Mansbach \
0	0	0	0
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0

..	...	...	...
695	0	0	0
696	0	0	0
697	0	0	0
698	0	0	0
699	0	0	0

	Author_Adam Silvera	Author_Adam Wallace	Author_Adir Levy \
0	0	0	0
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
..	...	...	...
695	0	0	0
696	0	0	0
697	0	0	0
698	0	0	0
699	0	0	0

	Author_Admiral William H. McRaven	...	Year_2015	Year_2016	Year_2017 \
0	0	...	0	0	0
1	0	...	0	0	0
2	0	...	0	0	0
3	0	...	0	0	0
4	0	...	0	0	0
..	...	...	...	...	...
695	0	...	0	0	0
696	0	...	0	0	0

697	0	...	0	0	0	
698	0	...	0	0	0	
699	0	...	0	0	0	

	Year_2018	Year_2019	Year_2020	Year_2021	Year_2022	Genre_Fiction
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	0	0	0	0	0	1
3	0	0	0	0	0	0
4	0	0	0	0	0	1
..	...	...	...	...	...	...
695	0	0	0	0	1	1
696	0	0	0	0	1	1
697	0	0	0	0	1	1
698	0	0	0	0	1	0
699	0	0	0	0	1	1

	Genre_Non Fiction
0	1
1	1
2	0
3	1
4	0
..	...
695	0
696	0
697	0

698 1

699 0

[700 rows x 324 columns]

pre processed - Data:

	Name	Author	User	Ratings \
82	The Five Dysfunctions of a Team: A Leadership ...		NaN	NaN
51	Autobiography of Mark Twain, Vol. 1		NaN	NaN
220	Knock-Knock Jokes for Kids		NaN	NaN
669	Principles for Dealing with the Changing World...		NaN	NaN
545	Unicorn Coloring Book: For Kids Ages 4-8 (US E...		NaN	NaN

	Reviews	Price	Year	Genre
82	-0.664098	-0.699680	NaN	NaN
51	-0.776559	0.122609	NaN	NaN
220	-0.644803	-0.905253	NaN	NaN
669	-0.72712	0.842113	NaN	NaN
545	-0.543977	-0.905253	NaN	NaN

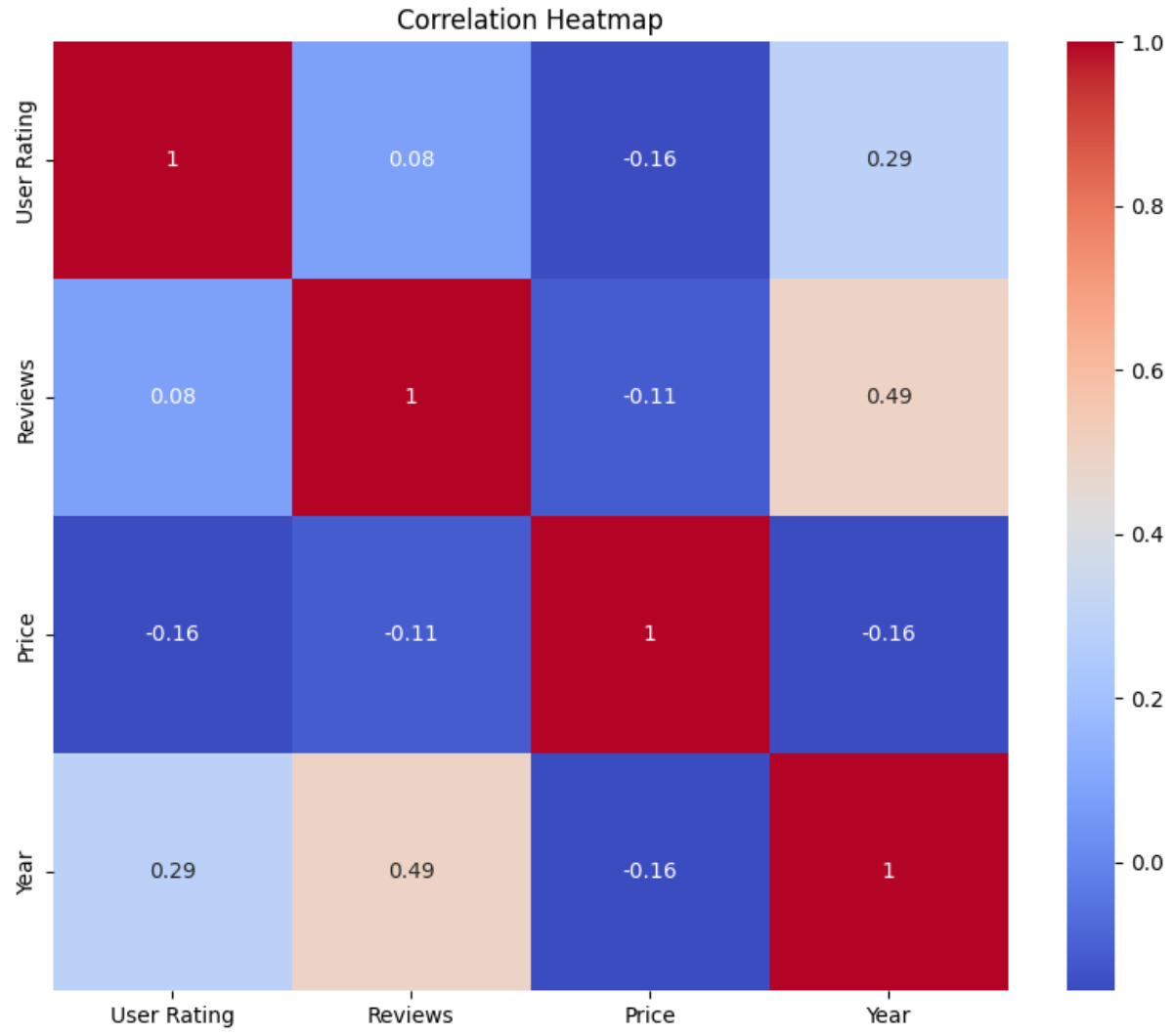
Target Data:

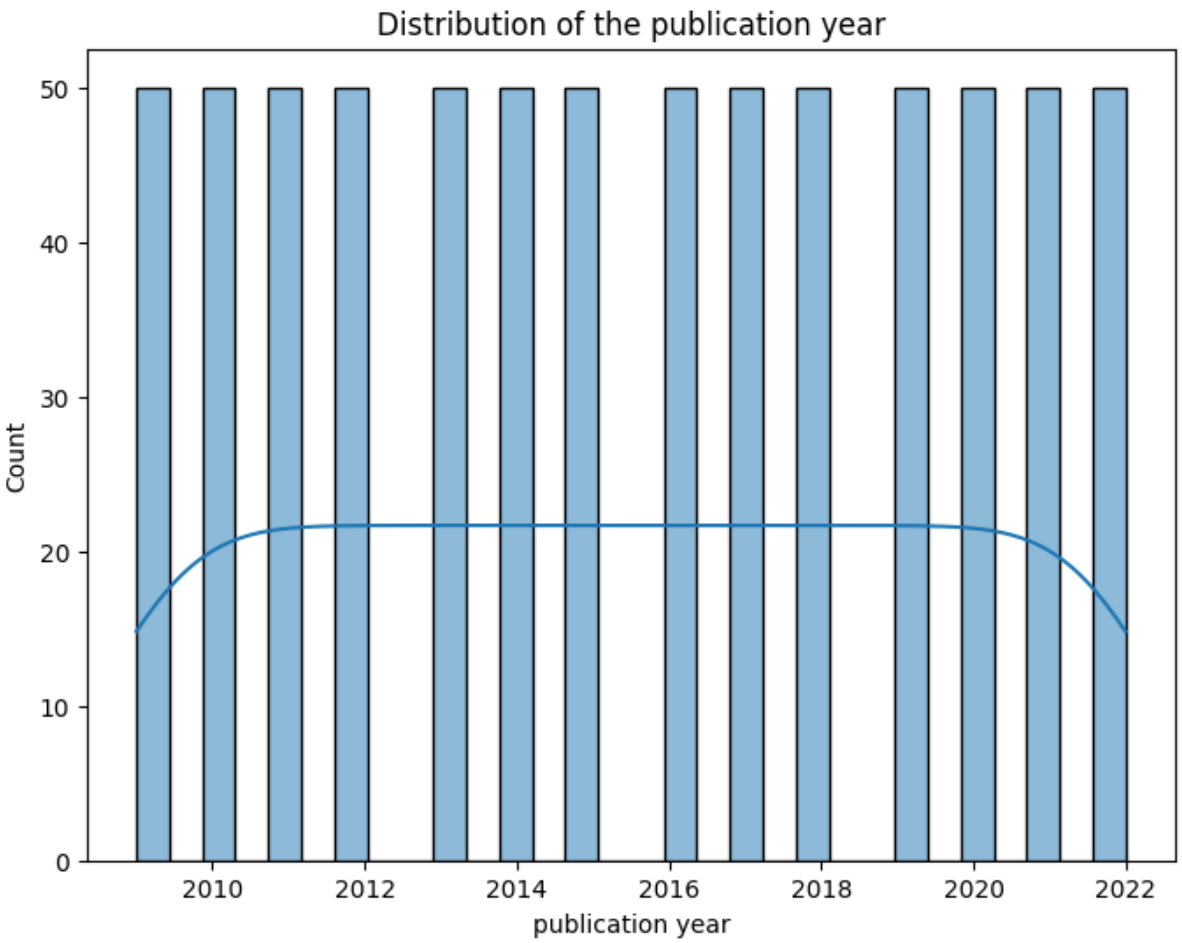
Reviews

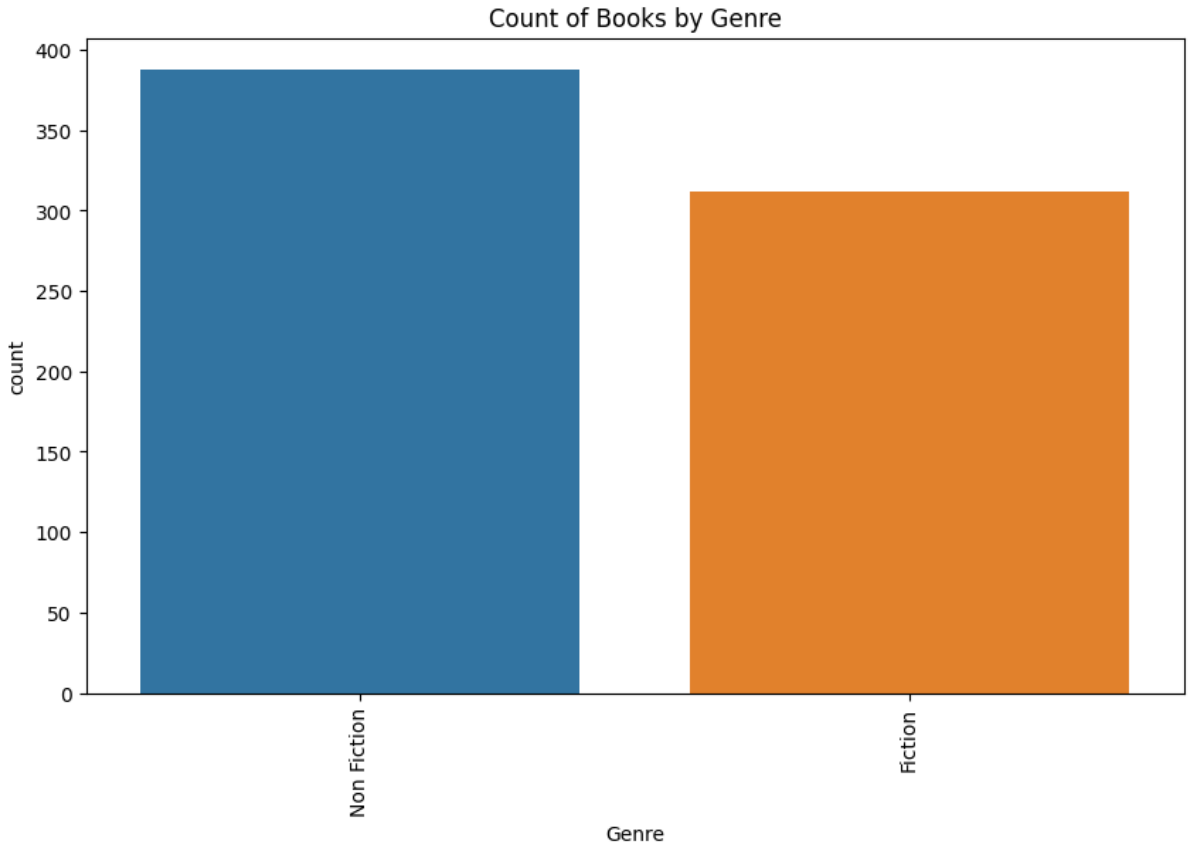
Summary Statistics:

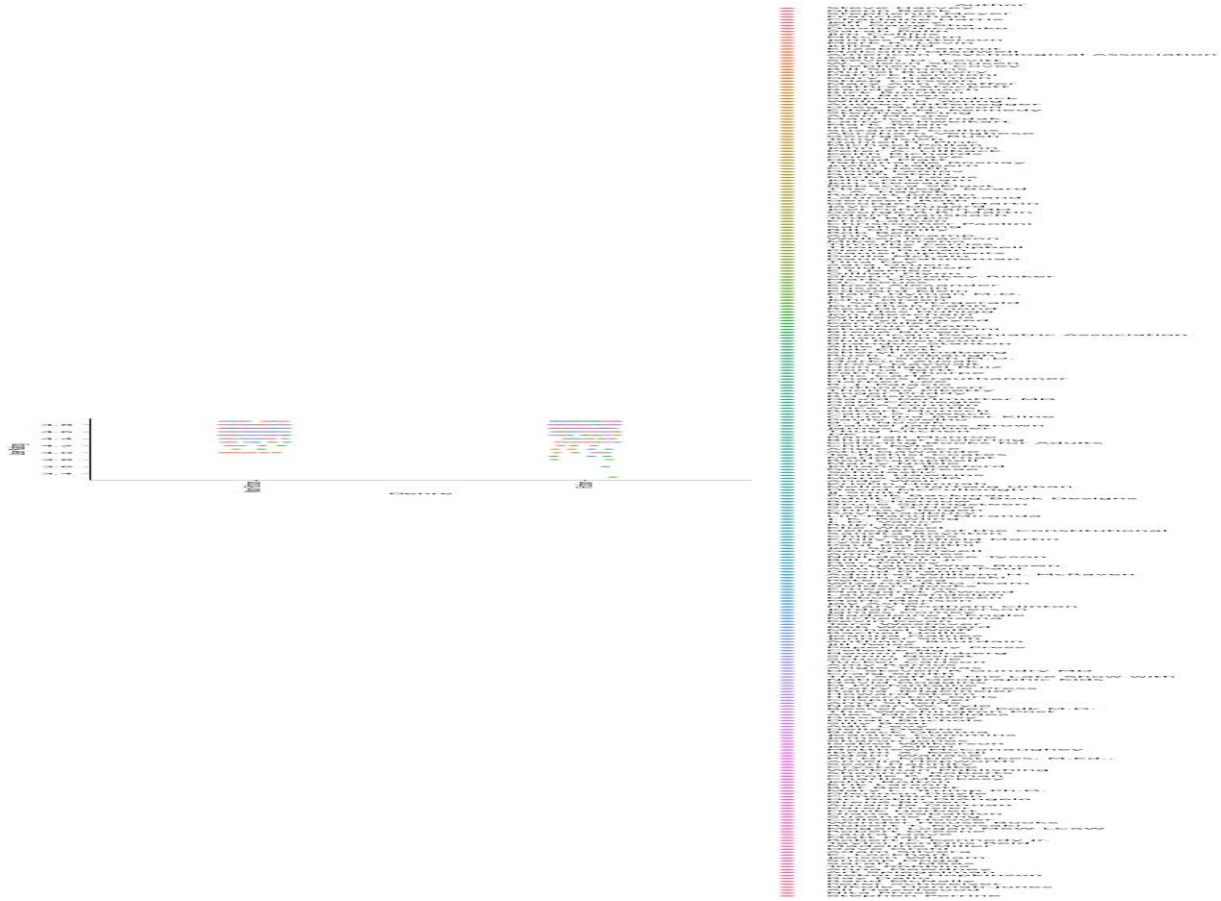
	User Rating	Reviews	Price	Year
count	700.000000	700.000000	700.000000	700.000000
mean	4.639857	19255.195714	12.700000	2015.500000
std	0.218586	23613.443875	9.915162	4.034011
min	3.300000	37.000000	0.000000	2009.000000

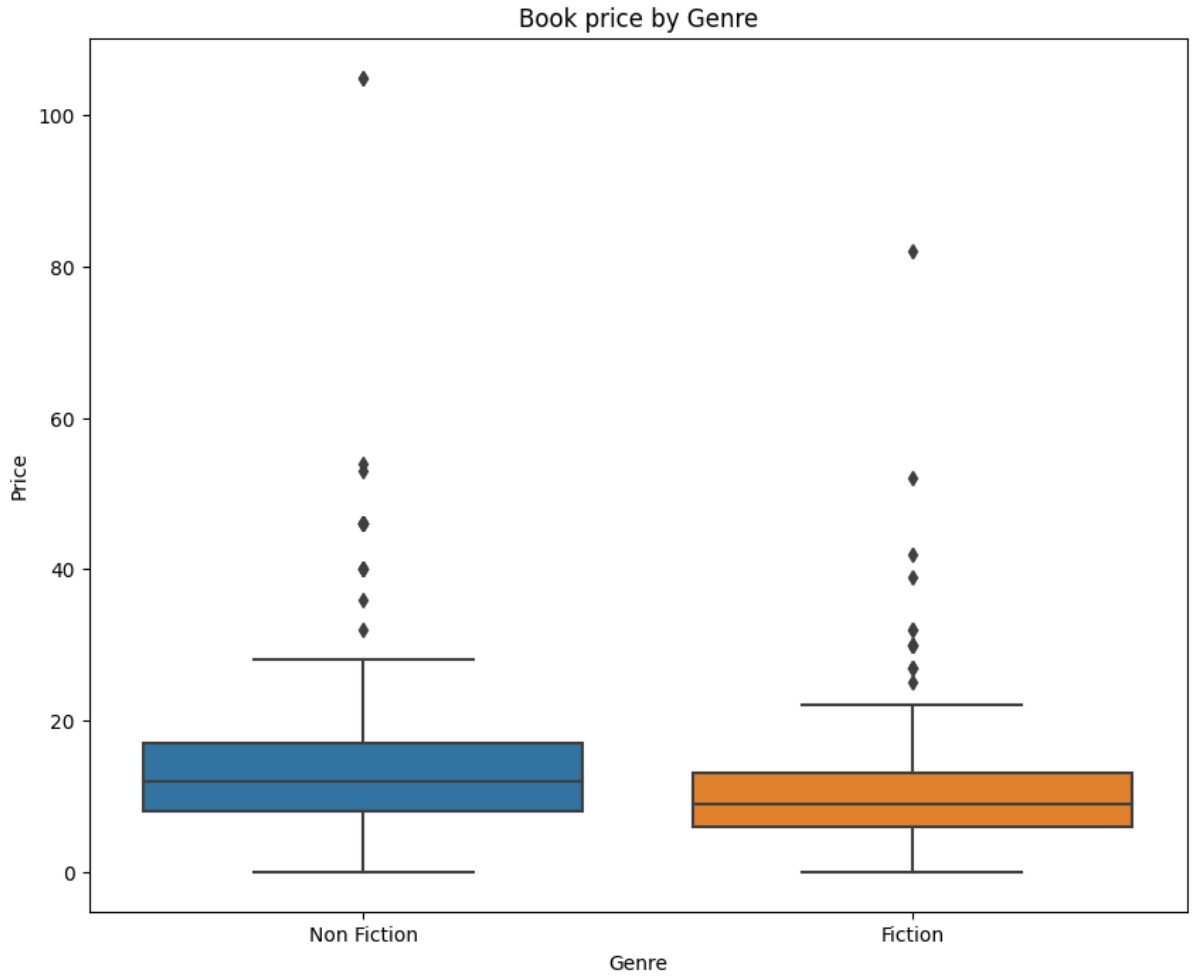
25%	4.500000	4987.250000	7.000000	2012.000000
50%	4.700000	10284.000000	11.000000	2015.500000
75%	4.800000	23358.000000	15.000000	2019.000000
max	4.900000	208917.000000	105.000000	2022.000000




Distribution of the publication year
Count
0
10
20
30
40
50
2010
2012
2014
2016
2018
2020
2022
publication year


Count of Books by Genre
400
350
300
250
200
150
100
50
0
count
Non Fiction
Fiction
Genre





3. Feature Engineering:

Feature engineering for LSTM involves creating sequences of data. LSTM models are capable of learning patterns from sequential data. In the case of sales forecasting, you might do the following:

Sequence Creation:

Organize your sales data into sequences. Each sequence could represent a window of historical sales data.

Normalization:

Normalize the data to ensure the LSTM model converges faster. You can use Min-Max scaling or standardization.

Input Features and Targets:

Create input features (X) and corresponding targets (y). For example, X could be a sequence of past sales values, and y could be the next sales value.

Look-Back Period:

Decide on the look-back period, which is the number of past time steps the model should consider when making a prediction.

4. Feature Selection:

Selecting relevant features for sales prediction involves understanding the factors influencing sales. Common features include historical sales data, seasonality, marketing efforts, economic indicators, and product characteristics. Techniques like correlation analysis, feature importance from machine learning models, and domain knowledge help identify significant predictors. Focus on relevant historical data, market trends, promotional activities, and product attributes to enhance the predictive accuracy.

5. Model Selection:

For future sales prediction, consider models like linear regression for simple trends, ARIMA for time-series patterns, or machine learning methods (random forests, gradient boosting) for complex relationships. Choose based on data characteristics, problem complexity, and desired accuracy vs. interpretability trade-offs.

6. Model Trainning:

To train an LSTM model for top-selling book sales prediction:

1. Preprocess data: Collect historical sales data, create sequences of sales with features like publication date.
2. Split data into training and testing sets.
3. Normalize data.
4. Build an LSTM model with appropriate architecture.
5. Compile the model with loss function and optimizer.
6. Train the model on the training data, tuning hyperparameters.
7. Evaluate the model using metrics like MSE on the testing data.
8. Use the trained model to make future sales predictions based on new input sequences.

7. Evaluation Metrics:

For evaluating book sales prediction models, use regression metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2) score.

MAE measures the average prediction error, MSE emphasizes larger errors, RMSE provides a more interpretable error measure, and R^2 measures the model's explanatory power.

Lower MAE, MSE, and RMSE and higher R^2 indicate better model performance.

Evaluate your model using these metrics to assess its accuracy and predictive power.

Python program for feature engineering ,model training and Evaluation

```
Import necessary libraries
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt

Load your dataset
data = pd.read_csv('/bestsellers with categories.csv')  Replace with your dataset file

Feature Engineering
 Normalize numerical features
scaler = MinMaxScaler()
numerical_cols = ['User Rating', 'Reviews', 'Price', 'Year']
data[numerical_cols] = scaler.fit_transform(data[numerical_cols])

 Encode categorical feature 'Genre' (you can use one-hot encoding)
data = pd.get_dummies(data, columns=['Genre'])

 Define your target variable
target_col = 'User Rating'  Replace with the actual variable you want to predict

Create sequences of data
sequence_length = 10 Adjust as needed
X = []
y = []
for i in range(len(data) - sequence_length):
    X.append(data[numerical_cols +
list(data.columns[8:])].iloc[i:i+sequence_length].values)
    y.append(data[target_col].iloc[i+sequence_length])

X = np.array(X)
y = np.array(y)

 Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

Build the LSTM model
model = Sequential()
model.add(LSTM(50, input_shape=(X_train.shape[1], X_train.shape[2])))
```

```
model.add(Dense(1)) Adjust for the number of output neurons based on your problem

 Compile the model
model.compile(loss='mean_squared_error', optimizer='adam')

 Train the model
model.fit(X_train, y_train, epochs=10, batch_size=12, validation_data=(X_test, y_test))

 Evaluate the model
test_loss = model.evaluate(X_test, y_test)
print(f'Test Loss: {test_loss}')

 Make predictions
y_pred = model.predict(X_test)
print("y_prediction:",y_pred)


 Visualize the results
plt.figure(figsize= (12, 6))
plt.plot(y_test, label='Actual')
plt.plot(y_pred, label='Predicted')
plt.legend()
plt.show()
from sklearn.metrics import mean_squared_error,r2_score
mse=mean_squared_error(y_test,y_pred)
r2=r2_score(y_test,y_pred)
print(“Metrics:”)
print("MEAN SQUARED ERROR:",mse)
print("R2-SCORE:",r2)
```

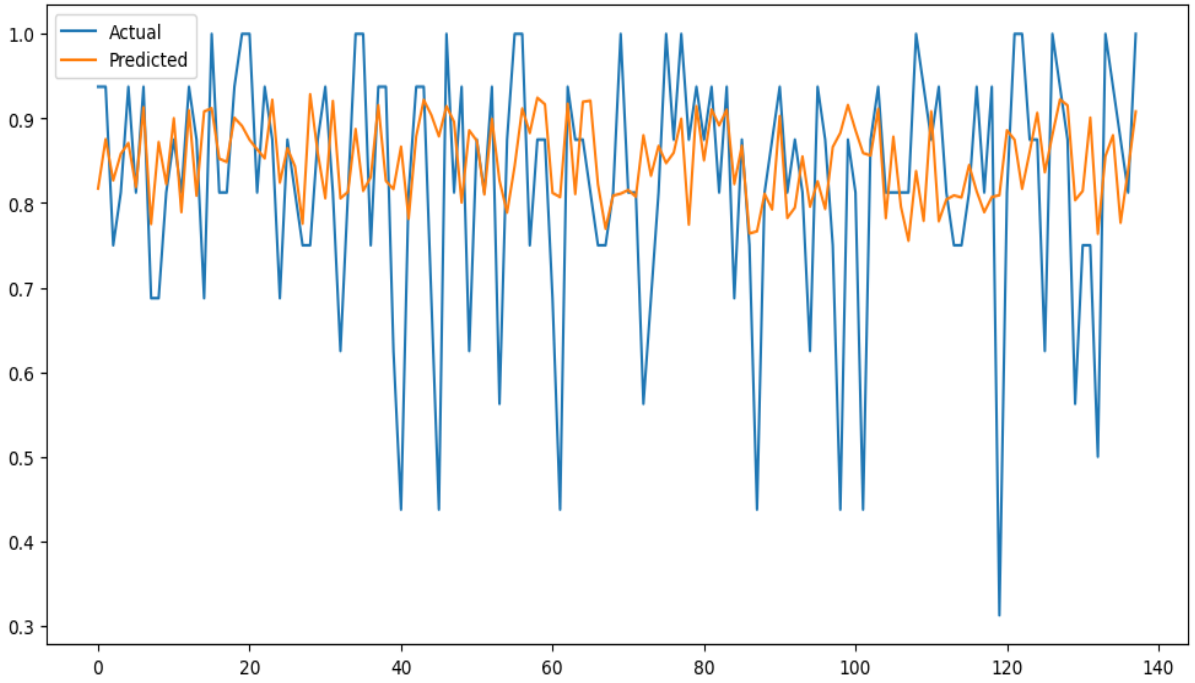
Output:

Model Training:

```
Epoch 1/10
46/46 [==============================] - 3s 17ms/step - loss: 0.0711 - val_loss: 0.0242
Epoch 2/10
46/46 [==============================] - 0s 7ms/step - loss: 0.0188 - val_loss: 0.0203
Epoch 3/10
46/46 [==============================] - 0s 6ms/step - loss: 0.0183 - val_loss: 0.0210
Epoch 4/10
46/46 [==============================] - 0s 7ms/step - loss: 0.0182 - val_loss: 0.0195
Epoch 5/10
```

46/46 [==============================] - 0s 6ms/step - loss: 0.0184 - val_loss: 0.0199
Epoch 6/10
46/46 [==============================] - 0s 7ms/step - loss: 0.0179 - val_loss: 0.0207
Epoch 7/10
46/46 [==============================] - 0s 6ms/step - loss: 0.0182 - val_loss: 0.0198
Epoch 8/10
46/46 [==============================] - 0s 6ms/step - loss: 0.0179 - val_loss: 0.0192
Epoch 9/10
46/46 [==============================] - 0s 7ms/step - loss: 0.0179 - val_loss: 0.0204
Epoch 10/10
46/46 [==============================] - 0s 7ms/step - loss: 0.0181 - val_loss: 0.0198
5/5 [==============================] - 0s 4ms/step - loss: 0.0198
Test Loss: 0.019775712862610817
5/5 [==============================] - 0s 6ms/step
y_prediction: [[0.8171518]
[0.8755395]
[0.82673275]
[0.8581269]
[0.8709007]
[0.8189731]
[0.91319305]
[0.7752176]
[0.8721105]
[0.82245994]
[0.9001946]
[0.789292]
[0.90959436]
[0.8086507]
[0.90822536]
[0.9120763]
[0.85243064]
[0.8485834]
[0.9008284]
[0.89121807]
[0.8747343]
[0.8635044]
[0.8528505]
[0.92204547]
[0.82419074]
[0.86484116]
[0.84327435] [0.7756054] [0.9286332]……..]

Actual Vs Predicted



Metrics:
MEAN SQUARED ERROR: 0.019775712297464994

R2-SCORE: 0.03566001942258268

Conclusion:

In conclusion, a successful future sales prediction project hinges on selecting appropriate models (e.g., regression, time series, machine learning), understanding and utilizing relevant features (historical sales, marketing efforts, economic indicators), and prioritizing accuracy while balancing interpretability. Integrating domain knowledge and iterative model refinement are crucial for optimal forecasting results in sales prediction projects.