

Sales Prediction Using Machine Learning Algorithms

Report by: Sayush Khadka

Table of Contents

1. Introduction:	1
1.1. Artificial Intelligence:	1
1.2. Machine Learning:	3
1.3. Chosen Topic:	5
2. Background:	7
2.1. Research on the selected topic (Problem domain):	7
2.1.1. Problem scenario with manual sales forecasting:	8
2.1.2. How can Machine Learning be used to Predict Sales?	8
2.2. Review and analysis of existing work and research work already done on the topic (Research papers):	9
2.2.1. Sales Prediction of Market using Machine Learning	9
2.2.2. Machine-Learning-Based Restaurant Sales Forecasting	11
2.2.3. Sales Prediction System Using Machine Learning	13
2.2.4. Comparative Analysis of Supervised Machine Learning Techniques for Sales Forecasting	15
2.2.5. Forecasting of Automobile Sales Based on Support Vector Regression Optimized by the Grey Wolf Optimizer Algorithm	17
2.3. Review and analysis of existing work and research work already done on the topic (Similar projects):	19
2.3.1. How to Forecast Sales using Machine Learning	19
2.3.2. Sales Prediction with Machine Learning	21
2.3.3. Walmart — Store Sales Forecasting	23
3. Solution:	26
3.1. Proposed approach and solution to solve the problem:	26
3.1.1. Using Machine Learning for Sales forecasting:	26
3.1.2. Machine learning as a solution:	27
3.2. Algorithms used:	29
3.2.1. Linear Regression:	31
3.2.2. KNN (K-Nearest Neighbor Regressor)	33
3.3. Pseudocode of the proposed solution:	35
3.4. Diagrammatic representations of the solution (flowchart)	37
3.5. Tools and libraries:	38

3.5.1. Programming language:	38
3.5.2. Tool or platform:	38
3.5.3. Libraries:	38
3.6. Machine learning model as a solution:.....	40
3.6.1. Importing necessary libraries:	40
3.6.2. Importing the dataset:	41
3.6.3. Analysing and understanding the dataset:	41
3.6.4. Converting the column heading to the same case letters:.....	42
3.6.5. Dataframe and its columns:.....	43
3.6.6. Total unique values in columns:	45
3.6.7. Checking missing/null values in the columns:	45
3.6.8. Visually checking the missing/null values (visual representation):	46
3.6.9. Handling missing/null values (item_weight):.....	47
3.6.10. Checking dataset for further possible cleaning and cleaning it:.....	52
3.6.11. Checking and removing outliers:	53
3.6.12. Some important visualizations:.....	55
3.6.13. Dropping useless columns:	58
3.6.14. Encoding:	58
3.6.15. Converting columns to lowercase after encoding:.....	60
3.6.17. Feature splitting:.....	61
3.6.18. Sales prediction using Linear Regression:	62
3.6.19. Sales prediction using KNN Regressor:	63
3.7. Comparison of the models used in this project:	65
3.7.1. The performance metrics:	65
3.7.2. Comparison table:	67
4. Conclusion:	69
4.1. Analysis of the work done:	69
4.2. How the solution addresses real-world solution:.....	69
4.3. Further work:.....	70
Bibliography	71

Table of tables:

Table 1 Dataframe and its columns.....	44
Table 2 Comparison table of the models.....	67

Table of figures:

Figure 1 Artificial Intelligence (Rijal, 2021)	1
Figure 2 Key components of AI (Kanade, 2022)	2
Figure 3 Machine Learning (Selig, 2022)	3
Figure 4 Types of machine learning (MathWorks, n.d.)	4
Figure 5 Sales forecast	5
Figure 6 How to Forecast Sales using Machine Learning (Naeem, 2022)	19
Figure 7 How to Forecast Sales using Machine Learning (importing dataset) (Naeem, 2022)	20
Figure 8 How to Forecast Sales using Machine Learning (model fit) (Naeem, 2022) ...	20
Figure 9 Sales Prediction with Machine Learning (Kharwal, 2021)	21
Figure 10 Sales Prediction with Machine Learning (importing libraries and dataset) (Kharwal, 2021)	21
Figure 11 Sales Prediction with Machine Learning (Model fit) (Kharwal, 2021)	22
Figure 12 Walmart — Store Sales Forecasting (Alves, 2021)	23
Figure 13 Walmart — Store Sales Forecasting (importing libraries and dataset) (Alves, 2021)	24
Figure 14 Walmart — Store Sales Forecasting (Model fit) (Alves, 2021)	25
Figure 15 Linear Regression (Saint, 2020)	31
Figure 16 Distance functions (KNN - continuous variables) (saedsayad, n.d.)	33
Figure 17 Distance functions (KNN- categorical variables) (saedsayad, n.d.)	34
Figure 18 Flowchart	37
Figure 19 Importing necessary libraries	40
Figure 20 Importing the dataset	41
Figure 21 Information of the dataset	41
Figure 22 Some more information of the dataset	42
Figure 23 converting the column heading to same case	42
Figure 24 Total unique values in columns	45
Figure 25 Checking missing/null values in the columns	45
Figure 26 Visually checking the missing/null values (visual representation)	46
Figure 27 Handling missing/null values (item_weight)	47
Figure 28 Calling the function null_value_count_column_item	48
Figure 29 Assigning values to uniqueltems variable	48
Figure 30 Filling in the missing/null values	49
Figure 31 Handled missing values of item_weight	50
Figure 32 Handling missing/null values from column outlet_size	50
Figure 33 Missing/null values have been handled	51
Figure 34 Checking dataset for further possible cleaning	52
Figure 35 Further cleaning (item_fat)	52
Figure 36 Checking outliers	53
Figure 37 Outliers	53
Figure 38 Removing outliers	53

Figure 39 Outliers have been removed	54
Figure 40 Univariate analysis	55
Figure 41 Bivariate analysis	56
Figure 42 Heat Map.....	56
Figure 43 Keeping column (outlet_indentifier)	57
Figure 44 Keeping column (outlet_indentifier)2	57
Figure 45 Dropping item_identifier	58
Figure 46 Encoding	58
Figure 47 Dropping some columns after encoding	59
Figure 48 Converting columns to lowercase after encoding	60
Figure 49 Feature splitting.....	61
Figure 50 Sales prediction using Linear Regression Model	62
Figure 51 Sales prediction using KNN Regressor	63
Figure 52 Sales prediction using KNN Regressor (continued)	63
Figure 53 Formula for R-Squared (R2) (Chugh, 2020).....	65
Figure 54 Formula for Mean Absolute Error (MAE) (Chugh, 2020)	66
Figure 55 Formula for Root Mean Square Error (RMSE) (Chugh, 2020)	66
Figure 56 Where (for performance metrics) (Chugh, 2020).....	66

1. Introduction:

1.1. Artificial Intelligence:

Artificial intelligence is the ability of machines, particularly computer systems, to imitate human intellectual functions. Expert systems, machine learning, natural language processing, and speech recognition are some examples of uses for AI. (Burns, 2022).

AI is now widely used in a wide range of applications, with varying degrees of sophistication. A common use of AI is in chatbots that can be found on websites or in smart speakers, as well as recommendation systems that can suggest future purchases (e.g., Alexa or Siri). Automating manufacturing processes, reducing various forms of cognitive labour duplication, and forecasting the weather and financial markets are all possible with AI (e.g., tax accounting or editing). AI is also used to play games, drive self-driving cars, process language, and do a variety of other things (Frankenfield, 2022).

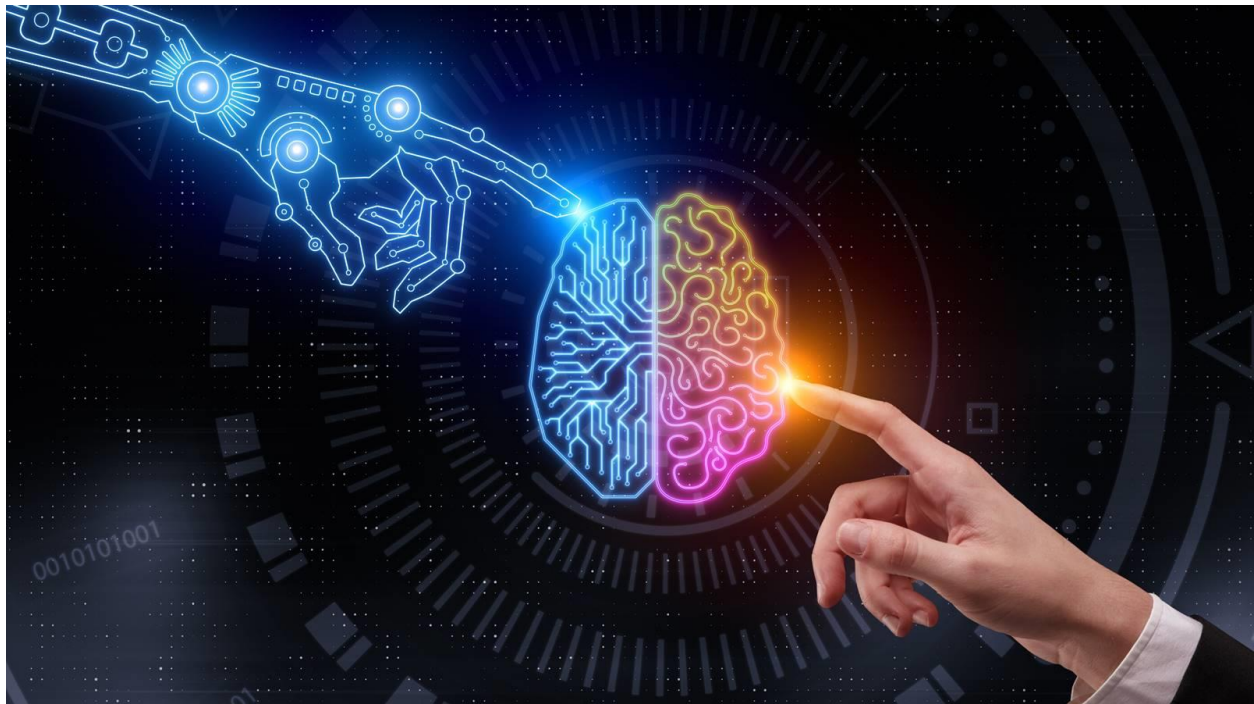


Figure 1 Artificial Intelligence (Rijal, 2021)

KEY COMPONENTS OF AI

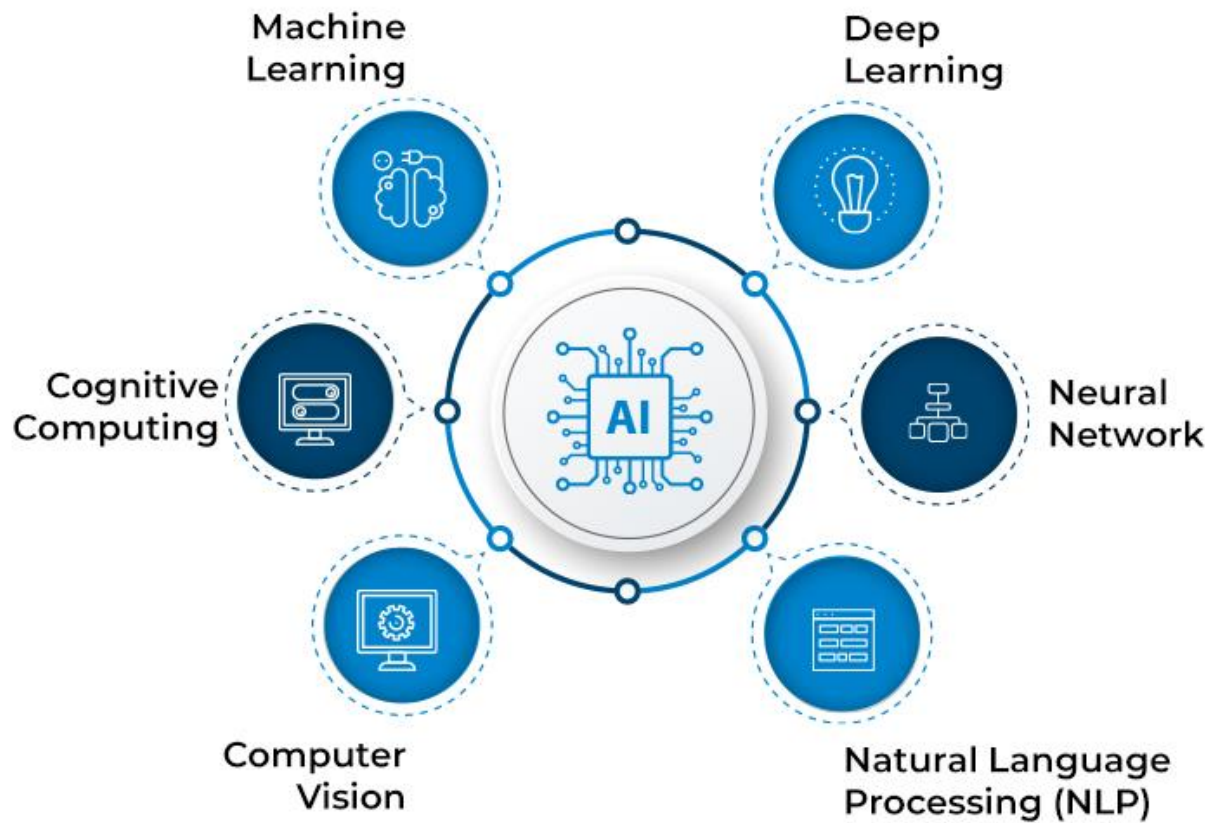


Figure 2 Key components of AI (Kanade, 2022)

1.2. Machine Learning:

A component of artificial intelligence (AI) called machine learning (ML) enables software applications to improve their propensity to anticipate outcomes without having to explicitly train them to do so. By using historical data as input, machine learning algorithms forecast new output values (Burns, 2021).

In this area of artificial intelligence, algorithms and data are used to replicate human learning, enabling robots to become better over time, increase the accuracy of their classification and prediction systems, or explore new data-driven insights. It uses three methods to function: first, combining data and algorithms to find patterns and classify data sets; second, assessing accuracy using an error function; and third, optimizing the fit of the data points into the model (Coursera, 2022).

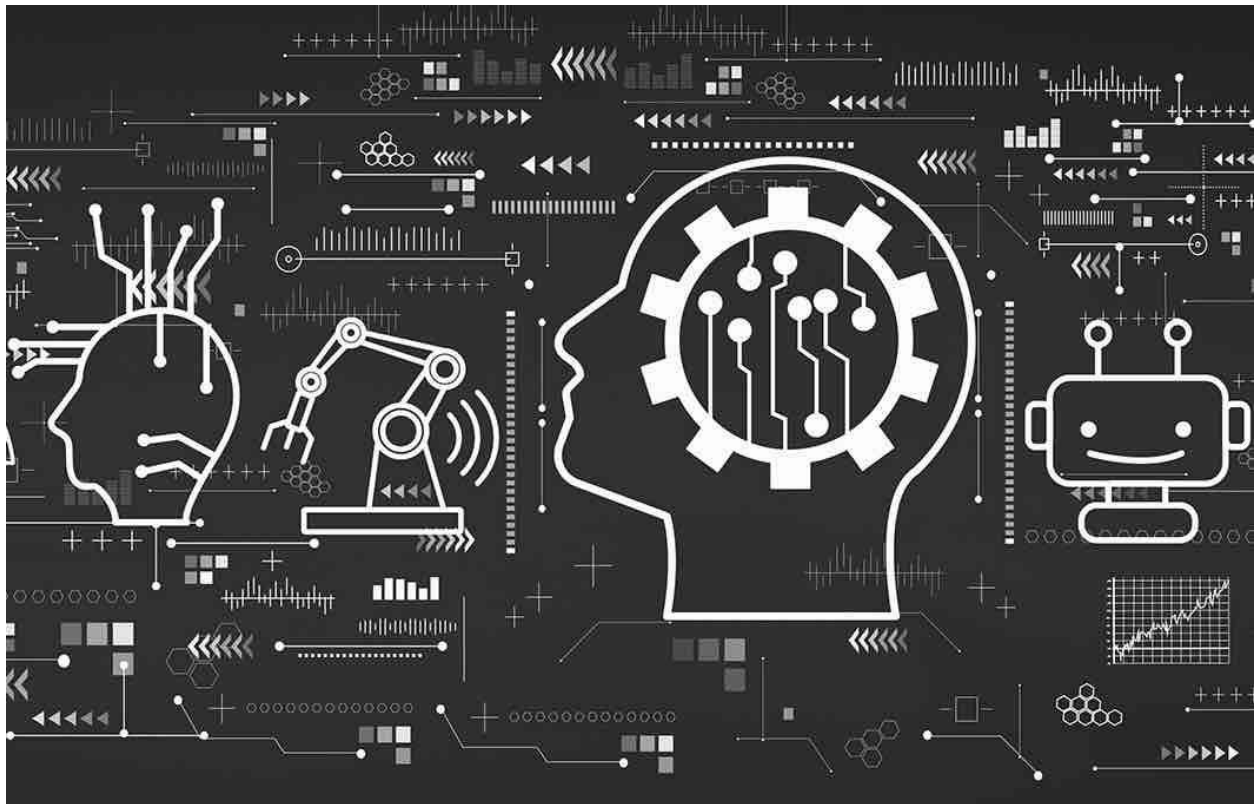


Figure 3 Machine Learning (Selig, 2022)

Types of machine learning:

- a) Supervised learning: The algorithm is fed information to aid in learning while it is being "supervised," which is why this sort of machine learning is known as "supervised" learning. The system uses the remaining pieces of information provided as input features, and the output provided is labelled data (Coursera, 2022).
- b) Unsupervised learning: These machine learning algorithms are trained on unlabelled data. The program goes over the data sets looking for any meaningful links. Both the predictions or suggestions that algorithms provide and the raw data that they utilize to learn are pre-set (Burns, 2021).
- c) Reinforcement learning: Reinforcement learning is the machine learning technique that most closely matches human learning. The deployed algorithm or operative picks up new skills by interacting with its environment and earning incentives, whether favourable or unfavourable. Common algorithms include deep adversarial networks, Q-learning, and temporal difference (Coursera, 2022).

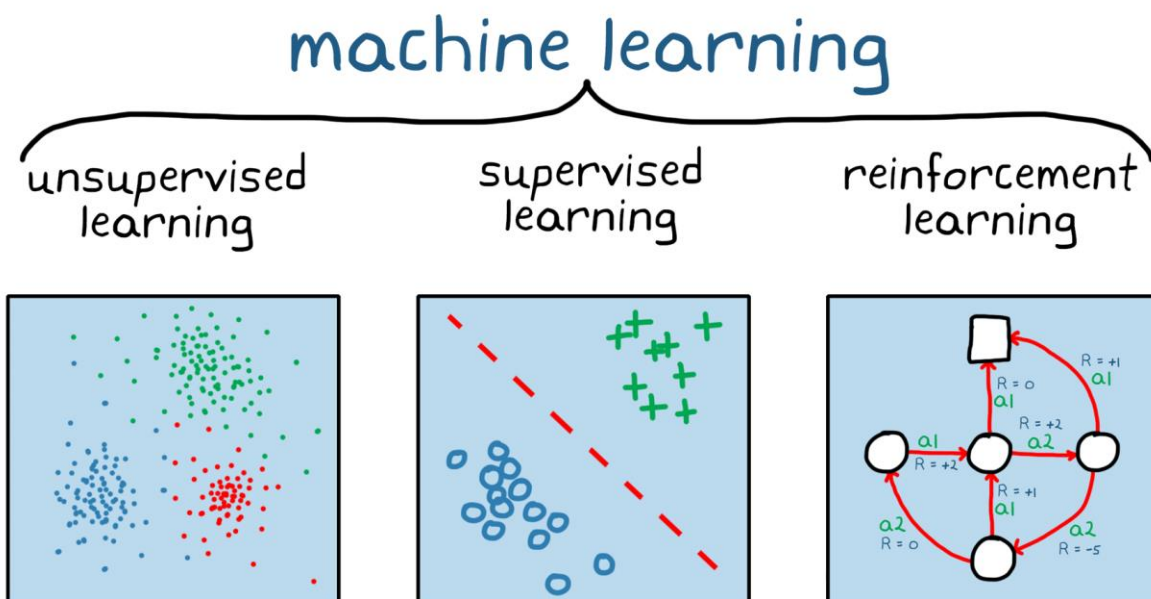


Figure 4 Types of machine learning (MathWorks, n.d.)

1.3. Chosen Topic:

Sales forecast

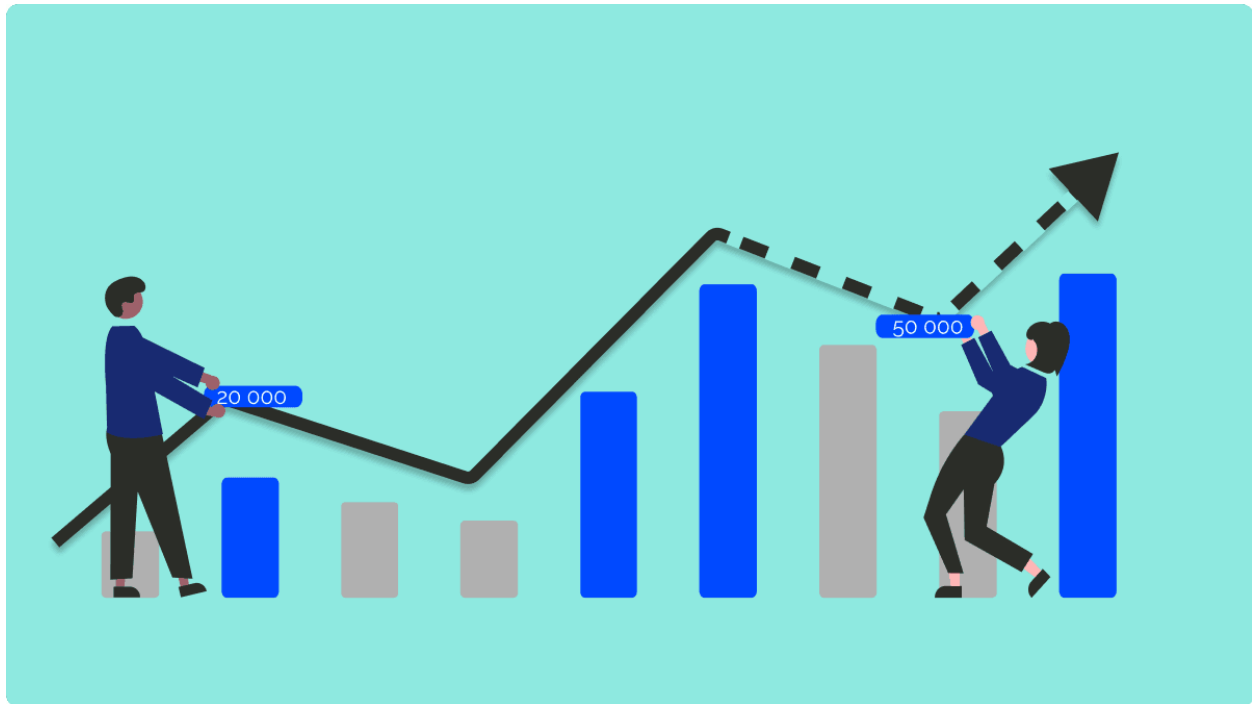


Figure 5 Sales forecast

A sales forecast is used to estimate future sales revenue. The health of the sales pipeline, market trends, and prior performance are frequently considered in sales estimates. A tool used by firms to forecast weekly, monthly, quarterly, and annual sales totals is the sales forecast (Bishop, 2020).

A sales forecast is essentially a prediction of how the market will react to a company's marketing endeavours. (Anaplan, n.d.).

Sales forecasting has always been a crucial area of study. To keep marketing teams working effectively, all vendors must now anticipate needs and opportunities effectively. In today's fast-paced world, executing this work manually would take time and increase the risk of serious mistakes that would lead to poor management of the company. The business sectors are crucial to the global economy because they are expected to produce enough commodities in the proper quantities to meet demand (Bajaj, 2020).

Importance of the **Sales forecast**:

- a) The future is considered in forecasts. It is impossible to stress the importance of a company establishing an accurate sales prediction (Anaplan, n.d.).
- b) Privately held businesses have more faith in their operations when leaders can put their trust in forecasts (Anaplan, n.d.).
- c) Accurate predictions enhance the market credibility of publicly traded corporations (Anaplan, n.d.).
- d) The entire organization gains from forecasting sales (Anaplan, n.d.).
- e) Sales estimates are used by production to plan its cycles, and by finance to develop budgets for hiring and capacity planning (Anaplan, n.d.).
- f) Forecasts aid in the planning of territory and targets for sales operations, channel and partner plans for sales strategy, and manufacturing capacity planning for the supply chain. (Anaplan, n.d.).

2. Background:

2.1. Research on the selected topic (Problem domain):

One of the essential components of effective sales forecasting is accuracy. The measurements and data collection required for your new product must be done with the utmost accuracy and precision. Having too much inventory is more likely if demand is overestimated. Underestimating the sales demand for your new product, on the other hand, puts you at risk of having to pay high shipping costs or having buyers walk away empty-handed (Wong, n.d.).

There are two main **approaches to predicting sales**:

- i) Manual Sales Forecasting: The conventional approach entails manually analysing the market and competing products and generating conclusions from the information gathered. All necessary forecasting tasks while using manual sales forecasting is completed relying on human (Wong, n.d.).
- ii) Machine learning-based sales forecasting: One of the biggest developments in the supply chain is machine-based sales forecasting, which may increase customer interaction and produce far more accurate demand predictions without human participation (Wong, n.d.).

2.1.1. Problem scenario with manual sales forecasting:

- i) Time-consuming: It takes a long time to forecast manually. The longer humans work on sales forecasting, the more likely it is that they will miss genuine sales opportunities (Wong, n.d.).
- ii) Biased and inaccurate: As a business owner, it's possible that one won't make a fully objective sales projection for their products. It will ultimately prove to be biased and inaccurate (Wong, n.d.).
- iii) Error-prone Human Mind: In addition, since humans make mistakes, adding some numbers incorrectly has been said to result in a work-doubling error (Wong, n.d.).
- iv) Limited data for forecasting: Because humans are limited in how much labour they can accomplish, it is nearly impossible for a single person or group of people to use all manual prediction techniques for projecting sales within a set time constraint.

2.1.2. How can Machine Learning be used to Predict Sales?

Machine learning examines millions of data points using several approaches, such as clustering and regression, before making predictions. Data points include things like demographic information, behavioural trends, and previous transactions. Businesses utilize machine learning algorithms to predict sales and revenue. This is done by using data from previous transactions to predict consumer behaviour. Businesses may create accurate estimates and prepare for impending occurrences by doing this. (Smolic, 2022).

2.2. Review and analysis of existing work and research work already done on the topic (Research papers):

2.2.1. Sales Prediction of Market using Machine Learning

Analysing market baskets is a practical method for locating complimentary commodities. Retailers can benefit from market basket analysis's statistics on related sales based on a particular group of commodities. Buyers of bread frequently purchase a variety of items associated with bread, such as milk, butter, or jam. It seems to sense that these establishments are conveniently situated close to one another in a mall so that customers may get to them quickly. To remind clients of related items and guide them logically away from the centre, such linked sets of goods must also be displayed side by side. Market basket analysis reveals the goods that are usually bought together and aids in planning supermarket layouts and marketing campaigns. To understand market consumer behaviour, it is required to evaluate data mining techniques. Regarding product positioning, pricing, endorsements, profitability, and whether any successful products exist without any key associated attributes, informed decisions can be made with simplicity. Similar products can be found, allowing them to be presented to various customers or placed close to one another (Soham Patangia, 2020).

Literature review:

With the help of connected devices, sensors, and mobile apps, the retail sector can be used as a beneficial testbed for big data tools and applications. The application of big data to retail operations is being researched. Historical sales data and loyalty programs can be used to obtain customer insights for operational planning. External data can be used for pricing and demand forecasts, including rival prices and environmental factors. But getting started with big data exploitation is not simple. One challenge is the physical capacity of the supply chain to respond to real-time changes captured by big data. Other challenges include a lack of qualified workers, a lack of supplier support, integration issues with IT, management issues with information sharing and process

integration, and problems with IT integration. It is suggested to develop a data maturity profile for retail businesses and to list prospective areas for more investigation. (Soham Patangia, 2020).

It is required to establish the rule to acquire the new knowledge to make it easier to recognize the value of the percentage from each of the datum when using specific algorithms, such as apriori. The frequency of occurrence of the data on the item set will be determined with the aid of this rule. This study compared market basket analysis with the apriori algorithm to market basket analysis without the use of an algorithm to get new knowledge. The comparable indicators were the concept, the regulation-development process, and the finalized rule. Although the rule was developed utilizing different procedures in both methodologies, the comparison revealed that the idea remained the same (Soham Patangia, 2020).

2.2.2. Machine-Learning-Based Restaurant Sales Forecasting

Introduction:

By analysing a wide range of machine learning (ML) techniques on a curated, real-world dataset, current research in this study is aimed to be broadened. Even though the outmoded ARIMA method for restaurant analysis has been the subject of substantial research. A strategy that is both simple and thorough to compare a wide range of models is employed. An ideal machine learning (ML) model will execute a prediction assignment and catch minute details like vacations while retaining performance when the forecast window is extended from one day to one week. It will receive training with the ideal quantity of features. How a set of forecasting models with low performance can be transformed from a raw dataset is described. The objective is to determine the set of techniques and models that yields the best results and to determine whether modern recurrent neural network architectures, such as LSTM, GRU, or TFT, perform better than conventional forecasting machine learning models, such as decision trees or simple linear regression (Schmidt, 2022).

Literature review:

Recent articles show that comparing machine learning (ML) techniques to forecast curated restaurant sales is a popular research topic. Additionally, two more current forecasting using ML problems that don't involve restaurants are looked at. Although there are a few significant ways in which the methodology differs from other recent forecasting publications—differences that are discussed—it is like those used in model training and feature engineering. The first key contrast between research approaches is the forecasting horizon window used. Since only one publication extended the forecast horizon beyond a one-time step, we see this research's main contribution as anticipating results for one week. The importance of consistent data represents yet another departure from previous studies. A stable dataset has always been crucial when working with time series because it prevents patterns or seasonality from being learned. Instead, each

instance is kept independent and may be anticipated based on its merit rather than suggested resemblance. However, there is only one study that mentions this stationary condition. The authors simply trained models using data that appeared to have no apparent trend in common, choosing not to further investigate the data. The stationary condition and evaluate its significance across various datasets as an addition to earlier studies is considered. The engineering of weather as a part of the feature set is the main aspect in which this work differs from prior publications. Even though this is not used in this research, only one newspaper doesn't use weather models as feature labels for predicting. Weather can be thought of as a potential upgrade in the future to attempt and improve results even more, despite not being a part of the job. The models chosen for testing are the key differentiator, which is to be emphasized in the research (Schmidt, 2022).

Some studies just look at standard methods like support vector machines, decision trees, and linear regression without discussing recurrent neural networks (RNN). In one paper, RNN models are the only family of statistical or machine learning models that are provided. RNN and conventional models are used sparingly in the latter two papers, which use a range of models. There are no results for the new TFT model, which is prominently featured in this paper and was first published in 2020, in any of the articles. RNN and ML models are compared to these authors to ensure a more accurate comparison is tested (Schmidt, 2022).

2.2.3. Sales Prediction System Using Machine Learning

Introduction:

Sales forecasting is crucial in many industries and aids in corporate growth by permitting the setting of long-term objectives. Sales forecasting is a crucial part of company planning and wise decision-making since it helps businesses better schedule their operations. There has only ever been one prediction model employed in previous research on sales forecasting. However, no one model can deliver the greatest results for every kind of product. The accurate data mining classification prediction results of this investigation. The decision tree technique can help vendors increase their revenues. As fundamental classification models, many prediction models are available. The outcomes of the prediction model were contrasted with those of other independent models. The results show that the prediction model's accuracy is higher than the accuracy of a single model. Sales forecasts are used by the warehousing department's sales and marketing to choose where to put the warehouse. Future sales patterns can also be predicted more precisely using sales data (Mansi Panjwani, 2020).

Literature review:

In these publications, a new predictive algorithm, observable data, and qualitative analysis of prior data prediction models using the Markov model to forecast the hidden values are presented as ways to improve the predictability of the model. The experimental results of this study demonstrate that the grey DNN model, a cutting-edge advancement in artificial intelligence, successfully forecasts sales volume. Sales forecasting is a crucial component of many professions, including economics, which analyses market trends and projects the potential size of the country's market for commodities. On data mining in small to medium-sized firms, however, there is a study. putting categorization processing to use for resource forecasting, forecasting of electric power, etc. Effective corporate planning and decision-making depend on accurate sales forecasting, which enables organizations to estimate sales and create informed plans. For offline businesses, sales

forecasting is essential. To predict future sales and establish strategies for the company's sales, statistical approaches like regression or a range of various models are generally used. Forecasting techniques frequently vary from one another and just consider the products' historical sales data. However, in circumstances where vast quantities were only pertinent to short-term sales, the prediction of a specific time series should be based on the company's previous sales (Mansi Panjwani, 2020).

The prevalence of internet use influences a variety of aspects of daily life, including business. Small and medium-sized businesses used new media to their advantage to classify their worldwide operations as internet commerce. The creation of an e-commerce sales prediction model based on data was necessary because there were no known practical implementations of how to estimate sales using historical transaction data. They developed a sales prediction model for small and medium-sized businesses that used historical sales data to project future sales, but the predicted sales weren't what they had anticipated because there wasn't enough historical sales data, and the model wasn't very good. Sales forecasting is to predict future sales for companies including supermarkets, grocers, restaurants, bakeries, and patisseries. By helping the company reduce the stock of products whose sales are projected to drop and increase it for products whose sales are anticipated to rise, sales forecasting helps the business increase sales and the representation of the sales output variable. The company, however, is unable to maintain the anticipated sales over the long term (Mansi Panjwani, 2020).

2.2.4. Comparative Analysis of Supervised Machine Learning Techniques for Sales Forecasting

Introduction:

The retail industry is one of the most significant and quickly developing commercial areas in the data science field due to the volume of data it creates, and the variety of optimization challenges it poses. Optimal prices, suggestions, discounts, and stock levels are just a few issues that can be resolved with the help of various data analysis approaches. Projecting the sales of commodities can be challenging in the fast-paced, constantly evolving commercial environment of today. Merchants could help reduce operating costs and boost sales by making only a small number of improvements to the sales forecast. Additionally, it could increase customer satisfaction (Stuti Raizada, 2021).

After reviewing the sales forecasting literature, the methodology was established. Several machine-learning techniques, including linear regression, random forest regression, KNN regression, SVR regression, and additional tree regression, were used to train the model. After discussing the outcomes of each model, a conclusion has finally been made (Stuti Raizada, 2021).

Literature review:

Microsoft Time Series Algorithm provides regression methods that are optimized for forecasting continuous data, such as product sales or demand over time. To predict continuous variables like product sales and demand over time, regression methods can be utilized. Microsoft's time series algorithm will be helpful in this process. One of the major advantages of the method is that, unlike the decision tree algorithm, it won't need any new additional columns to forecast trends. Any irregularities that we might have in the sales/demand could be predicted by the model based on the underlying data set that is supplied (Stuti Raizada, 2021).

To produce predictions, one merely needs to update the data that the model is using as input; as data accumulates, the model will consider this. The Microsoft Time Series algorithm's capacity for cross-prediction is one of its unique features. The technique can be used for two different but related series of data, and the resulting model will be able to predict the outcome of one series based on how the other series behaves by understanding their co-relation. As an example, consider the claim in the issue stated that "Observed sales of one car can affect expected sales of another car." Working of the Approach: To improve prediction accuracy, the method initially employs the Autoregressive Tree Models with Cross Prediction (ARTXP) and Autoregressive Integrated Moving Average (ARIMA) techniques. Long-term forecasts will be made using ARIMA, while short-term predictions will be made using the ARTXP algorithm (Stuti Raizada, 2021).

2.2.5. Forecasting of Automobile Sales Based on Support Vector Regression Optimized by the Grey Wolf Optimizer Algorithm

Introduction:

More and more consumer behaviour data are being included into different forecasting issues as the Internet and big data continue to grow, greatly increasing prediction accuracy. Due to market and environmental changes, automobile sales will fluctuate as the main form of transportation. The economy, the transportation sector, and dealers' ability to quickly change their marketing strategy can all be affected by accurate vehicle sales forecasts. Several elements, such as the product's inherent qualities, the economy, legislation, and other elements, might have an impact on the decision to purchase a vehicle (Qu F, 2022).

The sample data also show characteristics from several sources, enormous complexity, and considerable volatility. In this study, the monthly sales of autos were estimated using the Support Vector Regression (SVR) model, which has global optimization, a simple structure, good generalization capabilities, and is ideal for multi-dimensional, small sample data. The Grey Wolf Optimizer (GWO) program also improves the parameters to improve forecast accuracy (Qu F, 2022).

Literature review:

The elements that influence auto sales are first examined and determined using the grey correlation analysis approach. Second, it is employed in the construction of the GWO-SVR model for forecasting auto sales. Third, the suggested model is contrasted with the other four widely used approaches utilizing data from Suteng and Kaluola in the Chinese vehicle segment, which were employed in the experimental investigation. Finally, a few managerial implications are presented for the consideration (Qu F, 2022).

First, little research has been done that considers online user reviews; instead, estimates of auto sales have mostly relied on prior sales, the Baidu index, and the Google index. This article analyses information from customer star ratings to investigate how user emotion affects the forecast of vehicle sales. Additionally, a lot of research on sales

forecasting concentrates on the broader market or projects sales for a single year, which is less beneficial for consumers. This study makes projections for certain models or brands to offer more accurate reference opinions. Third, the number of automobiles sold can be influenced by factors beyond the car itself, such as the economy, costs, and raw materials, in addition to the attributes of the car itself. The sample data sources are diverse, the data is extremely complex, and the sales data exhibit a significant level of volatility and nonlinearity, as can be seen. The SVR-GWO prediction model, which uses the GWO algorithm to choose the optimal SVR model parameters, is suggested by this research to enhance prediction performance while accepting multi-source data and capturing nonlinear changes in data. (Qu F, 2022).

2.3. Review and analysis of existing work and research work already done on the topic (Similar projects):

2.3.1. How to Forecast Sales using Machine Learning

[Home](#) » [Data Science](#) » How to Forecast Sales using Machine Learning

How to Forecast Sales using Machine Learning

By [Awais Naeem](#) / [Data Analysis](#), [Data Science](#), [Regression](#) / [Leave A Comment](#) / April 8, 2022



Figure 6 How to Forecast Sales using Machine Learning (Naeem, 2022)

```
1 | store_sales = pd.read_csv('store_sale.csv')
2 | store_sales.head(10)
```

Output:

```
01 |      Date      store item  sales
02 | 0  2013-01-01    1  1    13
03 | 1  2013-01-02    1  1    11
04 | 2  2013-01-03    1  1    14
05 | 3  2013-01-04    1  1    13
06 | 4  2013-01-05    1  1    10
07 | 5  2013-01-06    1  1    12
08 | 6  2013-01-07    1  1    10
09 | 7  2013-01-08    1  1     9
10 | 8  2013-01-09    1  1    12
11 | 9  2013-01-10    1  1     9
```

Firstly, let's check if there are any null values in the dataset:

```
1 | store_sales.info()
```

Figure 7 How to Forecast Sales using Machine Learning (importing dataset) (Naeem, 2022)

To train Linear Regression, we can simply call it using scikit-learn and pass the training data. Moreover, we can use the 'predict' function of the linear regression model to get the predicted outputs using the test data:

```
1 | linreg_model = LinearRegression()
2 | linreg_model.fit(X_train, y_train)
3 | linreg_pred = linreg_model.predict(X_test)
```

To transform the predicted values back to their original scale, we need to call the 'inverser_transform' function of the MinMaxScaler. For that, we need to create a test set matrix containing all the input features of the test data and the predicted output instead of the real output:

```
1 | linreg_pred = linreg_pred.reshape(-1,1)
2 | linreg_pred_test_set = np.concatenate([linreg_pred,X_test], axis=1)
3 | linreg_pred_test_set = scaler.inverse_transform(linreg_pred_test_set)
```

Figure 8 How to Forecast Sales using Machine Learning (model fit) (Naeem, 2022)

2.3.2. Sales Prediction with Machine Learning



Figure 9 Sales Prediction with Machine Learning (Kharwal, 2021)

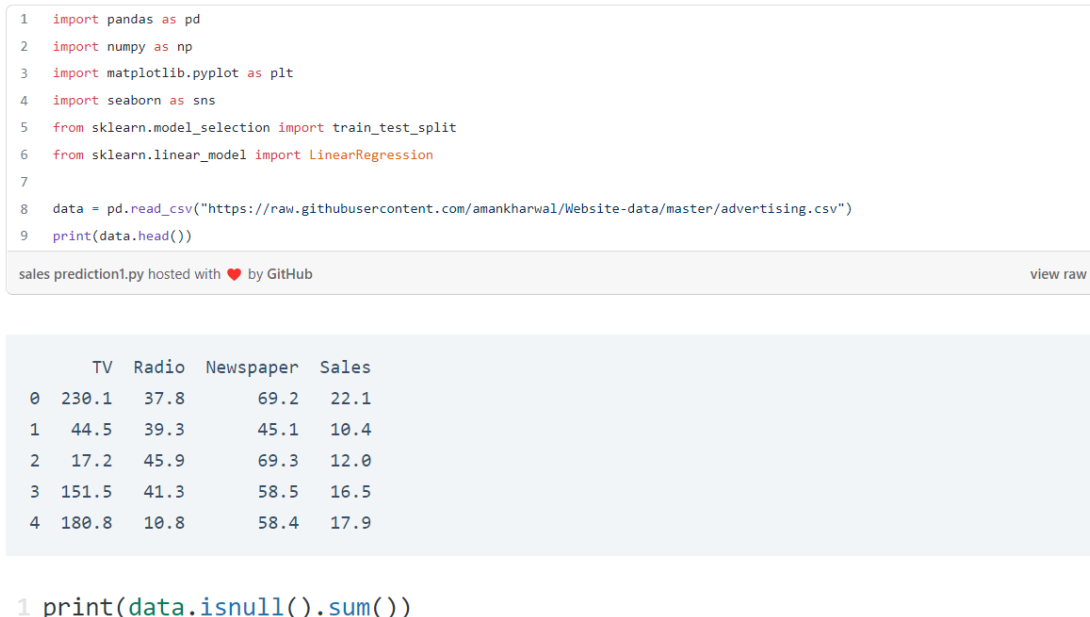
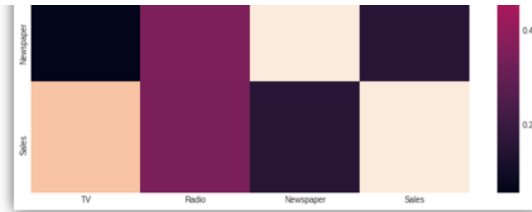


Figure 10 Sales Prediction with Machine Learning (importing libraries and dataset) (Kharwal, 2021)



Now let's prepare the data to fit into a machine learning model and then I will use a [linear regression](#) algorithm to train a sales prediction model using Python:

```
1 x = np.array(data.drop(["Sales"], 1))
2 y = np.array(data["Sales"])
3 xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.2, random_state=42)
4 model = LinearRegression()
5 model.fit(xtrain, ytrain)
6 ypred = model.predict(xtest)
7
8 data = pd.DataFrame(data={"Predicted Sales": ypred.flatten()})
9 print(data)
```

sales prediction3.py hosted with ❤ by GitHub

[view raw](#)

Figure 11 Sales Prediction with Machine Learning (Model fit) (Kharwal, 2021)

2.3.3. Walmart — Store Sales Forecasting



Sergio Alves

Aug 7, 2021 · 5 min read · [Listen](#)



Walmart — Store Sales Forecasting

Forecasting based on Decision Trees & Random Forest



For this Machine Learning project, I used the "Walmart Recruiting —

Figure 12 Walmart — Store Sales Forecasting (Alves, 2021)

```
import opendatasets as od
import os
from zipfile import ZipFile

import numpy as np # linear algebra
import pandas as pd # data processing
import seaborn as sns; sns.set(style="ticks", color_codes=True)
import matplotlib.pyplot as plt
```

Hosted on [Jovian](#) [View File](#)

Here we download the datasets from Kaggle.

```
dataset_url = 'https://www.kaggle.com/c/walmart-recruiting-store-sales-forec
od.download('https://www.kaggle.com/c/walmart-recruiting-store-sales-forecas
```

Skipping, found downloaded files in "./walmart-recruiting-store-sales-forecasting" (use force=True to force download)

Hosted on [Jovian](#) [View File](#)

Figure 13 Walmart — Store Sales Forecasting (importing libraries and dataset) (Alves, 2021)

DecisionTreeRegressor.

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

Hosted on [Jovian](#) [View File](#)

```
tree = DecisionTreeRegressor(random_state=0)
```

Hosted on [Jovian](#) [View File](#)

Now, we fit our model to the training data.

```
%%time
tree.fit(train_inputs, train_targets)
```

CPU times: user 2.17 s, sys: 26.8 ms, total: 2.2 s
Wall time: 2.19 s
DecisionTreeRegressor(random_state=0)



Hosted on [Jovian](#)  15 |  [View File](#)

Figure 14 Walmart — Store Sales Forecasting (Model fit) (Alves, 2021)

3. Solution:

3.1. Proposed approach and solution to solve the problem:

Machine Learning-Based Forecast: **A Superior Approach?**

3.1.1. Using Machine Learning for Sales forecasting:

- i) Find a Model: A machine learning algorithm is selected according to the nature of the dataset. Mainly regression model like linear regression is preferred and mostly used for sales forecasting (Smolic, 2022).
- ii) Collect Data: Machine learning requires data to produce sales projections, and the more the better. Clean the data and remove the outliers (Smolic, 2022).
- iii) Create and test the model: The model created in the first stage will be used to make forecasts. Usually, a fresh dataset with previous transaction data, demographic data, and other relevant details is created. This fresh dataset will be examined by the algorithm using the previously created model (Smolic, 2022).
- iv) Analyse results: The last step is to analyse the results of the forecast. By examining these outcomes, businesses may evaluate the efficiency of their machine-learning algorithms. (Smolic, 2022).

3.1.2. Machine learning as a solution:

i) Unlimited Data: Machine-based forecasting analyses millions of data points while concurrently taking into consideration an infinite number of demand elements by combining big data, cloud computing, and learning algorithms (Wong, n.d.).

ii) Accurate planning: By implementing machine learning algorithms into their data and customizing them to client needs, businesses can enhance their goods and services. They can also predict consumer behaviour more accurately, which will help them plan better (Smolic, 2022).

iii) Planning of Supply based on Demand: With the help of an accurate sales prediction based on tested databases, businesses can ascertain the interest in current goods or services and estimate that interest for future undertakings. Businesses can create revenue without having a surplus by adjusting supply to fit demand using this projection (Team, 2022).

iv) Higher Accuracy Level: The forecast is more accurate and precise because of machine-based analysis, which makes use of more data. The forecast will include details that might be added up, like product components, packaging, raw material valuation, third-party financial data, and other pertinent information. Additionally, automated software is considerably less likely to make a mistake than human intelligence because it has a much lower error rate (Wong, n.d.).

v) Building Marketing Plans: Sales projections are of the utmost significance to any company's marketing department. Depending on how well or poorly the projection seems, marketers can then step up their marketing, better target customers, rethink their product positioning, or re-evaluate the market. Forecasts serve as a benchmark that will serve as the objective of the campaigns and assist in budgeting marketing budgets on a weekly or annual basis (Team, 2022).

vi) A Classier Approach: Pattern identification, a technique used in machine learning forecasting, uses a variety of algorithms that can adapt to all types of data and are suitable for different demand forms (Wong, n.d.).

vii) It can decrease costs: When done effectively, demand forecasting can help in streamlining processes to increase productivity along the entire supply chain. Because it is feasible to predict customer requests and their timing more accurately, surplus inventory levels are decreased, increasing overall profitability (Galt, n.d.).

viii) Measuring company health: When a company grows, sales forecasts transform from investment pullers to indicators of the overall health of the company. Even Wall Street evaluates a company's success based on how readily it meets quarterly sales projections. They think that delivering lower-than-expected sales conveys to stakeholders that the company is performing poorly and possibly even that the management lacks ownership (Team, 2022).

ix) Resource allocation: Thanks to accurate sales forecasting, businesses may successfully manage their cash flow and allocate resources for future expansion (Mahalingam, n.d.).

3.2. Algorithms used:

The above-mentioned solution is carried out following the use of different machine learning algorithms.

The algorithms used for **Sales Forecast** in this project are:

- i) Linear Regression
- ii) KNN (K-Nearest Neighbor Regressor)

The overall solution also includes the accuracy comparison of the above-mentioned algorithms.

The link between dependent and independent variables is analysed using regression. Thus, regression algorithms support the forecasting of continuous variables such as real estate prices, market trends, climatic conditions, and oil and gas prices (Terra, 2022).

Terminology regarding Regression:

- i) Dependent Variable: The dependent variable is the primary variable in a regression study that we seek to comprehend or forecast. Another name for it is the target variable (JavaTPoint, n.d.).
- ii) Independent Variable: An independent variable, also referred to as a predictor, is any element that affects the dependent variables or that is used to predict their values (JavaTPoint, n.d.).
- iii) Outliers: An observation is considered an outlier if it is exceptionally high or extremely low in comparison to other observed values. The outcome might be harmed by an outlier; thus, it should be avoided (JavaTPoint, n.d.).
- iv) Multicollinearity: When independent variables have a larger correlation with one another than with other variables, it is said that multicollinearity has occurred. The dataset

shouldn't contain it because it makes it difficult to determine which variable has the biggest impact (JavaTPoint, n.d.).

v) Overfitting and Underfitting: The issue with overfitting is when our system performs well on the training dataset but poorly on the test dataset. An algorithm is said to be underfitted when it performs poorly even with training data (JavaTPoint, n.d.).

3.2.1. Linear Regression:

Simple Linear Regression:

A technique in machine learning that operates on supervised learning is linear regression. Regression creates a value for the goal prediction based on independent factors. Its primary application is to establish a link between variables and forecasting (Saint, 2020).

It performs the function of predicting the value of a dependent variable (y) based on an identified independent variable (x). Therefore, this regression technique discovers a linear relationship between x (the input) and y (the output) (output) (Saint, 2020).

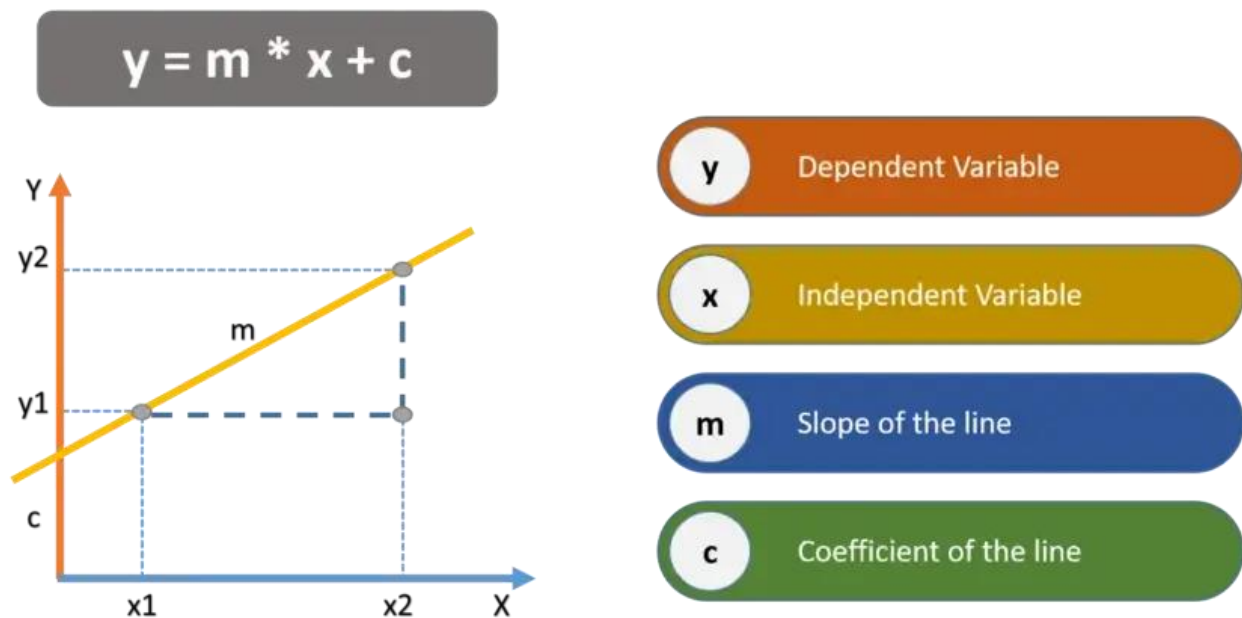


Figure 15 Linear Regression (Saint, 2020)

Linear Regression,

Mathematically,

$$Y = aX + b$$

Here,

Y = dependent variables (target variables),

X= Independent variables (predictor variables),

a and b are the linear coefficients

Multiple Linear Regression:

To forecast the outcome of a response variable, a statistical method called multiple linear regression (MLR), commonly referred to as multiple regression, makes use of several explanatory factors. To represent the linear relationship between the explanatory (independent) and response (dependent) variables, multiple linear regression is used (Hayes, 2022).

$$\hat{Y} = a + b_1X_1 + b_2X_2$$

- X_1 and X_2 are the independent variables.
- a is the Y-intercept
- b_1 is the net change in Y for each unit change in X_1 holding X_2 constant. It is called a partial regression coefficient, a net regression coefficient, or just a regression coefficient.

Figure 16 Multiple Linear Regression Formula

Multiple Linear Regression has been carried out in this project.

3.2.2. KNN (K-Nearest Neighbor Regressor)

K nearest neighbor is a simple method that stores all the relevant samples and uses a similarity metric to forecast the numerical target (e.g., distance functions). The non-parametric KNN technique has been utilized in statistical estimation and pattern identification since the early 1970s. (saedsayad, n.d.).

KNN regression can be easily implemented by calculating the numerical target's average of the K nearest neighbors. The inverse distance-weighted average of the K nearest neighbors is a different approach. Both KNN classification and KNN regression employ the same distance functions (saedsayad, n.d.).

Distance functions

Euclidean	$\sqrt{\sum_{i=1}^k (x_i - y_i)^2}$
Manhattan	$\sum_{i=1}^k x_i - y_i $
Minkowski	$\left(\sum_{i=1}^k (x_i - y_i)^q \right)^{1/q}$

Figure 17 Distance functions (KNN - continuous variables) (saedsayad, n.d.)

These three-distance metrics can only be used with continuous variables. The Hamming distance, which counts the number of times identical symbols appear differently in two strings of the same length, must be used when dealing with categorical data (saedsayad, n.d.).

Hamming Distance

$$D_H = \sum_{i=1}^k |x_i - y_i|$$

$$x = y \Rightarrow D = 0$$

$$x \neq y \Rightarrow D = 1$$

X	Y	Distance
Male	Male	0
Male	Female	1

Figure 18 Distance functions (KNN- categorical variables) (saedsayad, n.d.)

Examining the data first is the most straightforward way to choose the best value for K. Typically, a large K value obscures the distinct boundaries inside the feature space while decreasing total noise to improve precision. Another method for choosing a suitable K value in the past is to use cross-validation, which involves comparing your K value to one from a different data set. K ought to be at least 10 for most datasets. This produces significantly better results than 1-NN (saedsayad, n.d.).

3.3. Pseudocode of the proposed solution:

COLLECT a dataset

IMPORT libraries

IMPORT the dataset

PRE-PROCESSING

CHECK dataset is pre-processed

IF dataset is not pre-processed

HANDLE missing values in the dataset

REPLACE or **REMOVE** null values from the dataset

REMOVE outliers

ENDIF

EXTRACT the independent and the dependent variables

SPLIT the dataset into a train set and a test set

FEATURE SCALE the data

ASSIGN machine learning models in a variable

FIT the model into the training dataset

EVALUATE the model

TUNE parameter

MAKE prediction

3.4. Diagrammatic representations of the solution (flowchart)

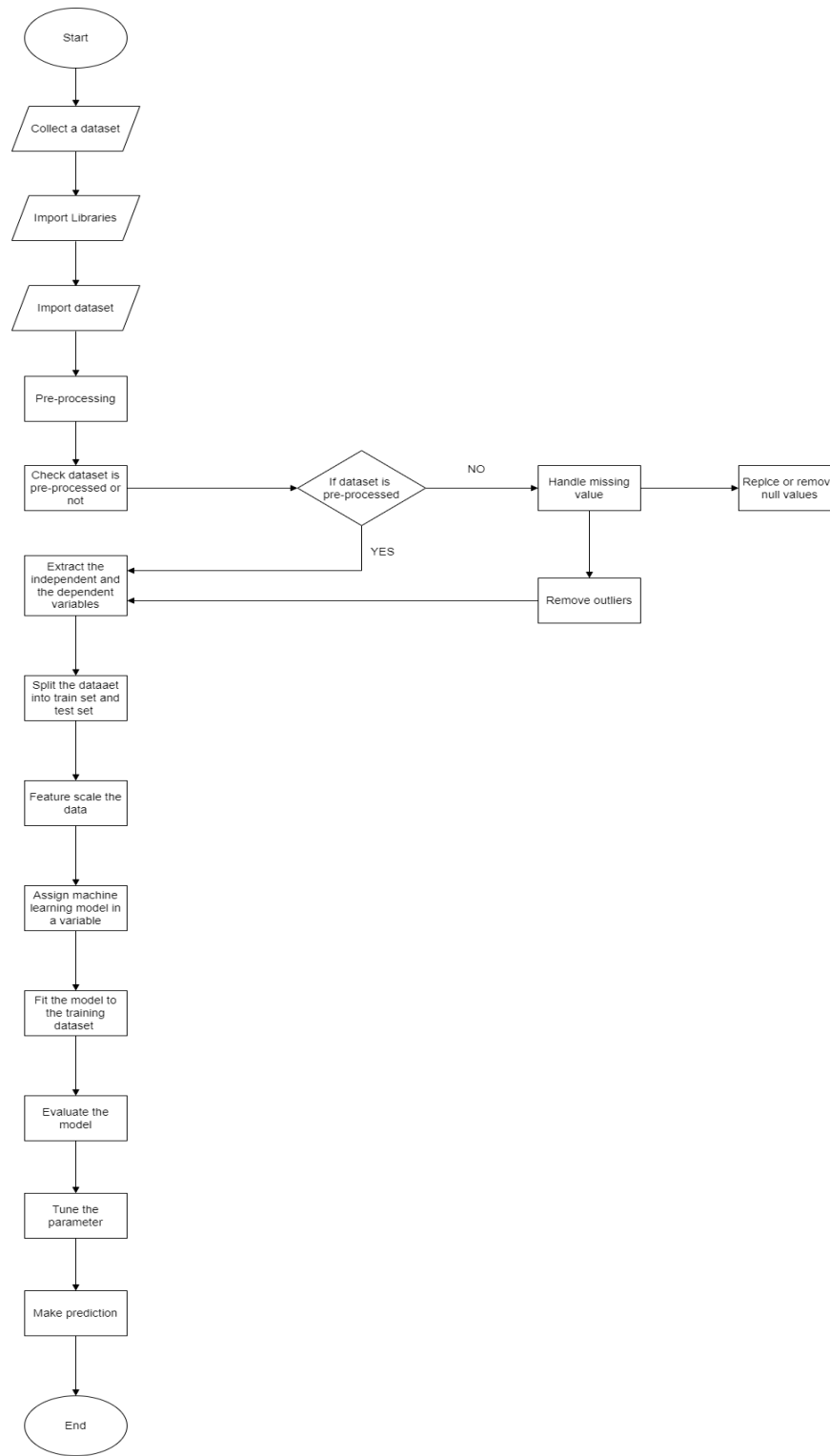


Figure 19 Flowchart

3.5. Tools and libraries:

3.5.1. Programming language:

Python: Python is a general-purpose language, which makes it flexible and appropriate for developing a variety of functionalities. Because it is an interpreted language, Python is a high-level programming language that can abstract away specifics from code and does not require compilation before execution. Even most unskilled programmers can grasp Python's code since it places such a strong focus on abstraction (Wilson, 2022).

3.5.2. Tool or platform:

Jupyter notebook: Jupyter Notebook is a free and open-source web platform that allows one to create and share documents with live code, equations, visualizations, and text. Jupyter Notebook maintenance is within the responsibility of Project Jupyter employees. The IPython project, which formerly had an IPython Notebook project of its own, gave rise to Jupyter Notebooks. The primary programming languages it supports are Julia, Python, and R, hence the name Jupyter. There are presently more than 100 additional kernels available, however, Jupyter comes with the IPython kernel, which enables Python programming (Driscoll, n.d.).

3.5.3. Libraries:

NumPy: The NumPy package contains multidimensional array objects and a selection of algorithms for array processing. NumPy is used along with SciPy and Matplotlib. Computer science uses this combination. NumPy is used to perform mathematical and logical calculations (Sai Nikhil Boyapati, 2020).

Pandas: Python programmers may perform analysis and data manipulation using the Pandas software library. The three-clause BSD license is used to distribute open-source software. It is based on the NumPy package and uses the Data Frame as its main data structure (Sai Nikhil Boyapati, 2020).

Matplotlib: Graphs are produced using the Matplotlib toolkit for Python. Visual representation is an important step in data science. Using a visual representation, it is possible to understand the division of data right away. Even though there are other libraries for expressing data, matplotlib is well known for being user-friendly and facilitating data visualization (Sai Nikhil Boyapati, 2020).

Seaborn: Seaborn is a free and open-source Python program for statistical graphics which in addition provides a data set-oriented API for examining correlations between various variables, it offers tools for selecting colour palettes that appropriately depict the data (Sai Nikhil Boyapati, 2020).

Scikit-learn: Scikit-learn is a free Python library with support vector machines, random forests, DBSCAN, k-means, gradient boosting, and other clusterings, classification, and regression algorithms that were created to operate with the NumPy and SciPy libraries are included (Sai Nikhil Boyapati, 2020).

3.6. Machine learning model as a solution:

- i) Collecting the required dataset.
- ii) Above mentioned algorithms to be implemented in the dataset
- iii) By contrasting parameters like accuracy score, mean absolute error, and maximum error, the output's performance can be improved.
- iv) The result will be able to show which machine learning algorithm was more accurate among those mentioned above to forecast sales

The detailed process flow is shown in the flowchart above and the actual procedure applied is discussed below.

The procedure except fitting the model for both the i) Linear regression and ii) KNN regressor is the same.

3.6.1. Importing necessary libraries:

Sales prediction system using Linear Regression and KNN Regressor

```
In [1]: 1 #importing Libraries
        2 import numpy as np
        3 import pandas as pd
        4 import matplotlib.pyplot as plt
        5 import seaborn as sns
```

Figure 20 Importing necessary libraries

These libraries have been imported at the very first (numpy, pandas, matplotlib.pyplot, seaborn). Some libraries used in this project have been imported at the place of their need.

3.6.2. Importing the dataset:

```
In [2]: 1 #importing dataset
        2 dataframe = pd.read_csv(r'sales_prediction_dataset/train.csv') #raw string, treat everything as it is
```

Figure 21 Importing the dataset

The dataset imported is a raw dataset downloaded from an online source referred to as big mart data set.

3.6.3. Analysing and understanding the dataset:

Analysing and understanding the dataset

```
In [3]: 1 #complete information of the dataset
        2 dataframe.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8523 entries, 0 to 8522
Data columns (total 12 columns):
 #   Column                      Non-Null Count  Dtype
---  -
 0   Item_Identifier              8523 non-null   object
 1   Item_Weight                  7060 non-null   float64
 2   Item_Fat_Content              8523 non-null   object
 3   Item_Visibility              8523 non-null   float64
 4   Item_Type                    8523 non-null   object
 5   Item_MRP                     8523 non-null   float64
 6   Outlet_Identifier            8523 non-null   object
 7   Outlet_Establishment_Year    8523 non-null   int64
 8   Outlet_Size                  6113 non-null   object
 9   Outlet_Location_Type         8523 non-null   object
10   Outlet_Type                  8523 non-null   object
11   Item_Outlet_Sales            8523 non-null   float64
dtypes: float64(4), int64(1), object(7)
memory usage: 799.2+ KB
```

Figure 22 Information of the dataset

This is a raw dataset to some level. Therefore, the dataset needs to be analysed and understood properly and should be cleaned and pre-processed accordingly to its needs.

```
In [4]: 1 #shape of the dataframe
        2 dataframe.shape
```

Out[4]: (8523, 12)

```
In [5]: 1 dataframe.head()
```

Out[5]:

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Identifier	Outlet_Establishment_Year	Outlet_Size	Outlet_Location
0	FDA15	9.30	Low Fat	0.016047	Dairy	249.8092	OUT049	1999	Medium	
1	DRC01	5.92	Regular	0.019278	Soft Drinks	48.2692	OUT018	2009	Medium	
2	FDN15	17.50	Low Fat	0.016760	Meat	141.6180	OUT049	1999	Medium	
3	FDX07	19.20	Regular	0.000000	Fruits and Vegetables	182.0950	OUT010	1998	NaN	
4	NCD19	8.93	Low Fat	0.000000	Household	53.8614	OUT013	1987	High	

Figure 23 Some more information about the dataset

3.6.4. Converting the column heading to the same case letters:

```
In [6]: 1 #converting dataframe columns to same case - Lowercase
        2 dataframe.columns = dataframe.columns.str.lower()
```

```
In [7]: 1 #complete information of the dataset
        2 dataframe.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8523 entries, 0 to 8522
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  -
0   item_identifier        8523 non-null  object
1   item_weight            7060 non-null  float64
2   item_fat_content       8523 non-null  object
3   item_visibility        8523 non-null  float64
4   item_type              8523 non-null  object
5   item_mrp               8523 non-null  float64
6   outlet_identifier       8523 non-null  object
7   outlet_establishment_year 8523 non-null  int64
8   outlet_size            6113 non-null  object
9   outlet_location_type    8523 non-null  object
10  outlet_type            8523 non-null  object
11  item_outlet_sales       8523 non-null  float64
dtypes: float64(4), int64(1), object(7)
memory usage: 799.2+ KB
```

Figure 24 converting the column heading to the same case

3.6.5. Dataframe and its columns:

S.no	Column	Dtype	Description
1.	item_identifier	object	Has item codes.
2.	item_weight	float64	Has item weight.
3.	item_fat_content	object	Tells the fat content of the item
4.	item_visibility	float64	Tells the visibility of the item.
5.	item_type	object	Tells the type of item it is.
6.	item_mrp	float64	Price of the item.
7.	outlet_identifier	object	Has outlet codes.
8.	outlet_establishment_year	int64	Tells the establishment year of the outlet.
9.	outlet_size	object	Tells the size of the outlet.
10.	outlet_location_type	object	Tells the tier of the outlet location.

11.	outlet_type	object	Tells the type of outlet it is.
12.	item_outlet_sales	float64	Tells the sales of items from the outlet

Table 1 Dataframe and its columns

3.6.6. Total unique values in columns:

```
In [8]: 1 #checking unique values
        2 #how many unique values are in column
        3 dataframe.nunique()

Out[8]: item_identifier      1559
        item_weight         415
        item_fat_content      5
        item_visibility      7880
        item_type            16
        item_mrp             5938
        outlet_identifier     10
        outlet_establishment_year 9
        outlet_size           3
        outlet_location_type   3
        outlet_type           4
        item_outlet_sales     3493
        dtype: int64
```

Figure 25 Total unique values in columns

3.6.7. Checking missing/null values in the columns:

```
In [10]: 1 #checking null values
         2 #how many null values are in column and in which columns
         3 dataframe.isna().sum()

Out[10]: item_identifier      0
         item_weight         1463
         item_fat_content      0
         item_visibility      0
         item_type            0
         item_mrp             0
         outlet_identifier     0
         outlet_establishment_year 0
         outlet_size          2410
         outlet_location_type  0
         outlet_type           0
         item_outlet_sales     0
         dtype: int64
```

Figure 26 Checking missing/null values in the columns

item_weight and outlet_size have missing/null values.

3.6.8. Visually checking the missing/null values (visual representation):

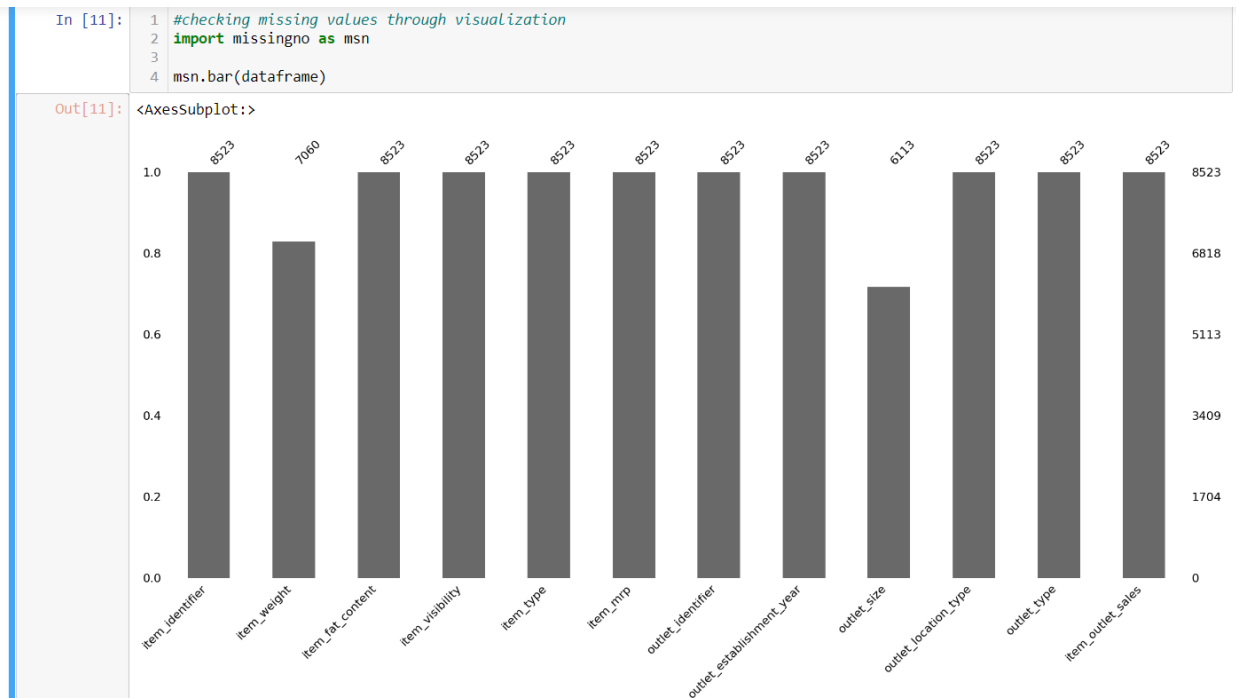


Figure 27 Visually checking the missing/null values (visual representation)

Visually checking the missing/null values from the columns using the 'missingno' library.

The short bar plots have missing/null values.

3.6.9. Handling missing/null values (item_weight):

```
Handling missing values

In [13]: 1 #handling missing values for item_weight
          2 dataframe['item_type'].unique()

Out[13]: array(['Dairy', 'Soft Drinks', 'Meat', 'Fruits and Vegetables',
               'Household', 'Baking Goods', 'Snack Foods', 'Frozen Foods',
               'Breakfast', 'Health and Hygiene', 'Hard Drinks', 'Canned',
               'Breads', 'Starchy Foods', 'Others', 'Seafood'], dtype=object)

In [14]: 1 #number of unique values in the column item_type
          2 dataframe['item_type'].nunique()

Out[14]: 16

In [17]: 1 #Function to calculate total null values in a column according to the unique values in the column
          2 #Best suitable for categorical values
          3
          4 column_name = dataframe['item_type'] #column name
          5
          6 def null_value_count_column_item(dataframe, column):
          7     dataframe_null = {}
          8     uniqueItems = column.unique()
          9     sNo = 1
          10
          11     for item in uniqueItems:
          12
          13         dataframe_null[item] = (dataframe[column_name == item].isna()).sum()
          14         print(f'{sNo}. {item}')
          15         print(dataframe_null[item])
          16         print()
          17         sNo+=1
          18
```

Figure 28 Handling missing/null values (item_weight)

We are filling the missing/null values through the item_type and their mean item_weight respectively for more accurate results.

- i) First, the unique items in the item_type are displayed and checked.
- ii) Then the total number of unique items in the column item_type is checked.
- iii) Then user-defined function was created named null_value_count_column_item, which takes dataframe and column as parameters.

```

18
In [20]: 1 #calling the function
          2 null_value_count_column_item(dataframe, column_name)

1. Dairy
item_identifier      0
item_weight         116
item_fat_content     0
item_visibility      0
item_type            0
item_mrp             0
outlet_identifier    0
outlet_establishment_year  0
outlet_size         186
outlet_location_type  0
outlet_type          0
item_outlet_sales    0
dtype: int64

2. Soft Drinks
item_identifier      0
item_weight         71
item_fat_content     0

```

Figure 29 Calling the function `null_value_count_column_item`

- iv) Then the function is called having arguments as (dataframe, column_name) where dataframe is the name given to the imported dataset and column_name is item_type.
- v) This function returns each unique item based on item_type with their columns and sum of missing/null values respectively.

```

In [21]: 1 uniqueItems = dataframe['item_type'].unique()
          2 uniqueItems

Out[21]: array(['Dairy', 'Soft Drinks', 'Meat', 'Fruits and Vegetables',
               'Household', 'Baking Goods', 'Snack Foods', 'Frozen Foods',
               'Breakfast', 'Health and Hygiene', 'Hard Drinks', 'Canned',
               'Breads', 'Starchy Foods', 'Others', 'Seafood'], dtype=object)

In [22]: 1 dataframe.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8523 entries, 0 to 8522
Data columns (total 12 columns):
#   Column                                Non-Null Count  Dtype
---  ---                                ---
0   item_identifier                       8523 non-null   object
1   item_weight                          7060 non-null   float64
2   item_fat_content                     8523 non-null   object
3   item_visibility                      8523 non-null   float64
4   item_type                            8523 non-null   object
5   item_mrp                            8523 non-null   float64
6   outlet_identifier                    8523 non-null   object
7   outlet_establishment_year            8523 non-null   int64
8   outlet_size                         6113 non-null   object
9   outlet_location_type                 8523 non-null   object
10  outlet_type                          8523 non-null   object
11  item_outlet_sales                    8523 non-null   float64
dtypes: float64(4), int64(1), object(7)
memory usage: 799.2+ KB

```

Figure 30 Assigning values to `uniqueItems` variable

- vi) Assigning each unique item from item_type to uniqueItems named variable

```

In [50]: 1 #filling the missing values of item_weight according to the unique values from item_type respectively
2 uniqueItems = dataframe['item_type'].unique()
3 selectItem = {}
4 print('Mean of items in item_type: (The mean of values are different when the values were null and now when filled ) ')
5 print('This shows the mean after the missing values have been handled')
6 print()
7 for item in uniqueItems:
8
9     selectItem = (dataframe[dataframe['item_type'] == item])
10    fillItemWeight = selectItem['item_weight'].mean()
11    dataframe['item_weight'] = dataframe['item_weight'].fillna(value=fillItemWeight, inplace=False)
12    print(f'{item} : {fillItemWeight}')
13

```

Mean of items in item_type: (The mean of values are different when the values were null and now when filled)
This shows the mean after the missing values have been handled

Dairy : 13.42606890459364
Soft Drinks : 12.09932784769921
Meat : 12.943386032009979
Fruits and Vegetables : 13.259571977823414
Household : 13.3915949501029
Baking Goods : 12.47569400820137
Snack Foods : 13.065293006478209
Frozen Foods : 12.957181669198507
Breakfast : 12.893794972695146
Health and Hygiene : 13.191425387333513
Hard Drinks : 11.693776336646742
Canned : 12.495597194923421
Breads : 11.736255930342235
Starchy Foods : 13.658542164072198
Others : 13.7723917452485
Seafood : 12.730217121245584

Figure 31 Filling in the missing/null values

vii) Filling in missing/null values through the item_type and their mean item_weight respectively for more accurate results using fillna() method.

Note: The mean value of each unique item_type displayed above has been calculated after filling in the missing/null values rather than the mean value that has been used to replace the missing/null value. **The missing/null values have been replaced with the correct mean value while executing. The mean value displayed above shows the values calculated after filling in the missing/null values.**

```
In [51]: 1 #missing value from item_weight has been handled
2 dataframe.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8523 entries, 0 to 8522
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  -
0   item_identifier        8523 non-null   object
1   item_weight            8523 non-null   float64
2   item_fat_content       8523 non-null   object
3   item_visibility        8523 non-null   float64
4   item_type              8523 non-null   object
5   item_mrp               8523 non-null   float64
6   outlet_identifier      8523 non-null   object
7   outlet_establishment_year 8523 non-null   int64
8   outlet_size            6113 non-null   object
9   outlet_location_type    8523 non-null   object
10  outlet_type            8523 non-null   object
11  item_outlet_sales       8523 non-null   float64
dtypes: float64(4), int64(1), object(7)
memory usage: 799.2+ KB
```

```
In [52]: 1 dataframe['item_weight'].isna().sum()

Out[52]: 0
```

Figure 32 Handled missing values of item_weight

viii) Handling missing/null values from column outlet_size

```
In [53]: 1 # Handling the missing values in dataframe['outlet_size']
2
3 dataframe['outlet_size'].fillna(dataframe['outlet_size'].mode()[0], inplace=True) #mode()[0]- get mode of each column
```

```
In [54]: 1 dataframe.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8523 entries, 0 to 8522
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  -
0   item_identifier        8523 non-null   object
1   item_weight            8523 non-null   float64
2   item_fat_content       8523 non-null   object
3   item_visibility        8523 non-null   float64
4   item_type              8523 non-null   object
5   item_mrp               8523 non-null   float64
6   outlet_identifier      8523 non-null   object
7   outlet_establishment_year 8523 non-null   int64
8   outlet_size            8523 non-null   object
9   outlet_location_type    8523 non-null   object
10  outlet_type            8523 non-null   object
11  item_outlet_sales       8523 non-null   float64
dtypes: float64(4), int64(1), object(7)
memory usage: 799.2+ KB
```

```
In [55]: 1 #missing value from outlet_size has been handled
2 dataframe['outlet_size'].isna().sum()

Out[55]: 0
```

Figure 33 Handling missing/null values from column outlet_size

outlet_size is a categorical column so the missing/null values are replaced using the mode of the column.

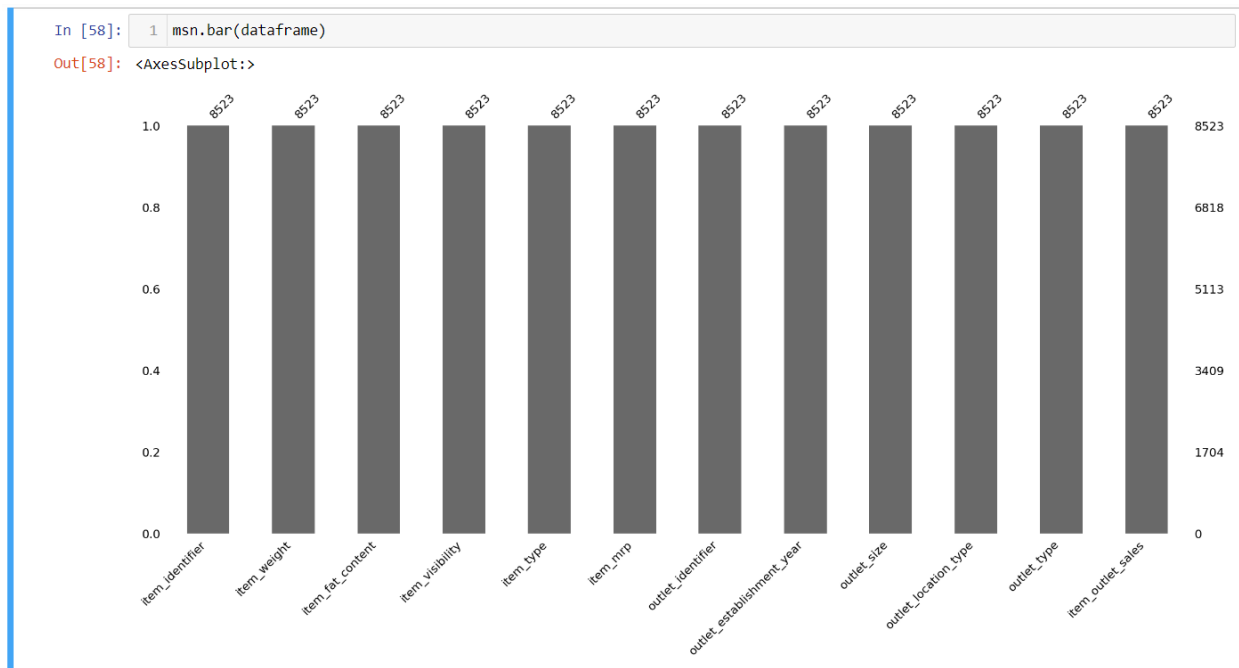


Figure 34 Missing/null values have been handled

The figure shows there aren't any missing values in the dataset now.

3.6.10. Checking dataset for further possible cleaning and cleaning it:

Further cleaning dataset

```
In [61]: 1 dataframe['outlet_identifiser'].unique()
Out[61]: array(['OUT049', 'OUT018', 'OUT010', 'OUT013', 'OUT027', 'OUT045',
               'OUT017', 'OUT046', 'OUT035', 'OUT019'], dtype=object)

In [62]: 1 dataframe['outlet_establishment_year'].unique()
Out[62]: array([1999, 2009, 1998, 1987, 1985, 2002, 2007, 1997, 2004], dtype=int64)

In [63]: 1 dataframe['outlet_size'].unique()
Out[63]: array(['Medium', 'High', 'Small'], dtype=object)

In [64]: 1 dataframe['outlet_location_type'].unique()
Out[64]: array(['Tier 1', 'Tier 3', 'Tier 2'], dtype=object)

In [66]: 1 dataframe['outlet_type'].unique()
Out[66]: array(['Supermarket Type1', 'Supermarket Type2', 'Grocery Store',
               'Supermarket Type3'], dtype=object)

In [67]: 1 dataframe['item_fat_content'].unique()
Out[67]: array(['Low Fat', 'Regular', 'low fat', 'LF', 'reg'], dtype=object)
```

Figure 35 Checking dataset for further possible cleaning

Checking categorical columns and their unique items in them.

```
In [68]: 1 #Creating a dictionary to map all records of dataframe['item_fat_content'] to low or regular only respectively.
2
3 item_fat={
4     'Low Fat': 'low',
5     'LF': 'low',
6     'low fat': 'low',
7     'Regular': 'regular',
8     'reg': 'regular',
9 }
10
11 dataframe['item_fat_content'] = dataframe['item_fat_content'].map(item_fat)
12
13 dataframe['item_fat_content'].unique()
Out[68]: array(['low', 'regular'], dtype=object)
```

Figure 36 Further cleaning (item_fat)

Cleaning the dataset further by reducing the items to 2 items in the item_fat column.

3.6.11. Checking and removing outliers:

Removing Outliers

```
In [70]: 1 # checking for outliers
2 dataframe.plot(kind='box', subplots=True, layout=(1,7), figsize=(20,7))
```

Figure 37 Checking outliers

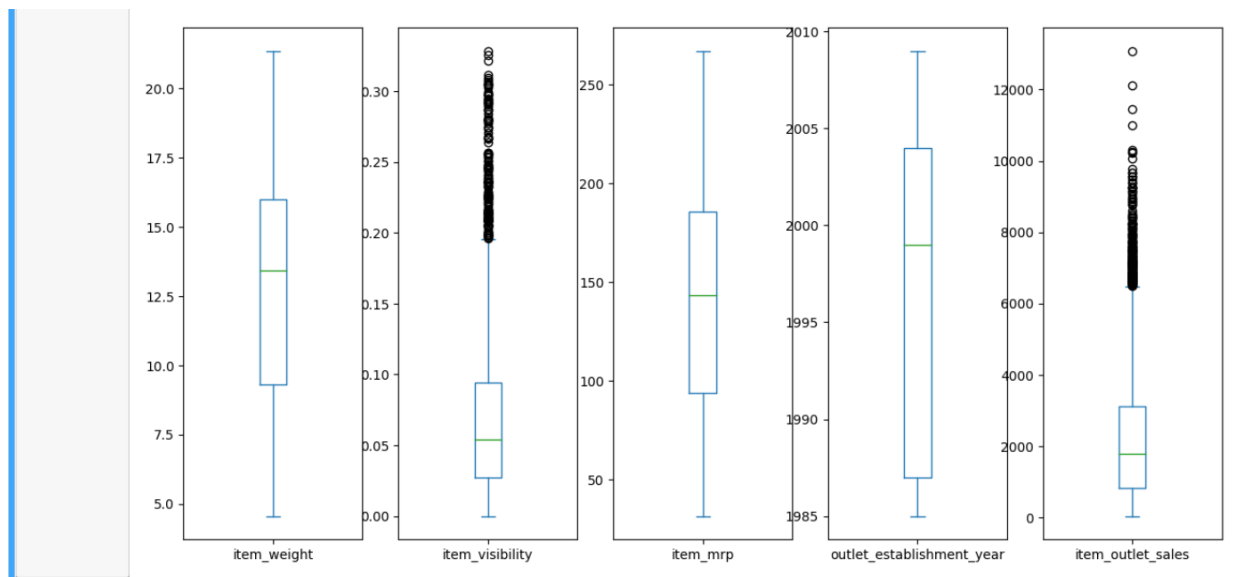


Figure 38 Outliers

Outliers are present in the item_visibility and item_outlet_sales columns.

```
In [72]: 1 # removing outliers
2 dataframe['item_visibility'] = dataframe[dataframe['item_visibility'] < 0.18]['item_visibility']

In [74]: 1 # Nan/null value had been created while removing the outliers in the item_visibility column
2 mean = dataframe['item_visibility'].mean()
3 dataframe['item_visibility'] = dataframe['item_visibility'].fillna(mean)

In [76]: 1 # checking for outliers
2 # Outliers in Item_Outlet_Sales are neglected as sales can sometime go high suddenly in some seasons
3
4 dataframe.plot(kind='box', subplots=True, layout=(1,7), figsize=(20,7))
```

Figure 39 Removing outliers

The outliers have been removed for item_visibility. Nan/null value had been created while removing the outliers in the item_visibility column which have been filled with the mean of the column.

The outliers from item_outlet_sales have been neglected because sales can sometimes go high suddenly in some seasons.

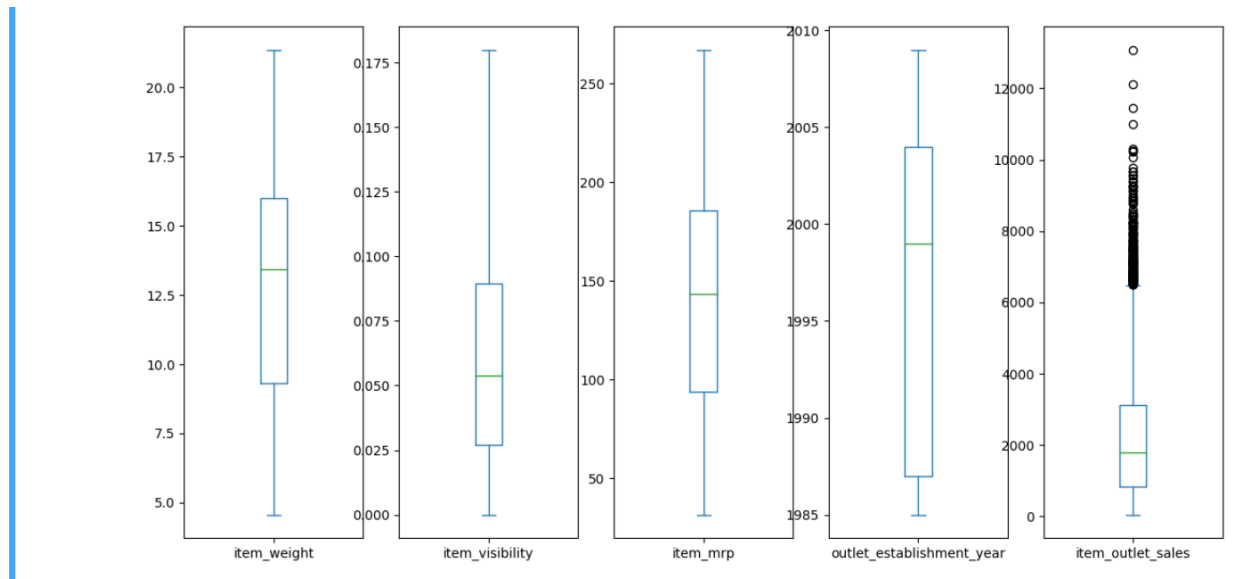


Figure 40 Outliers have been removed

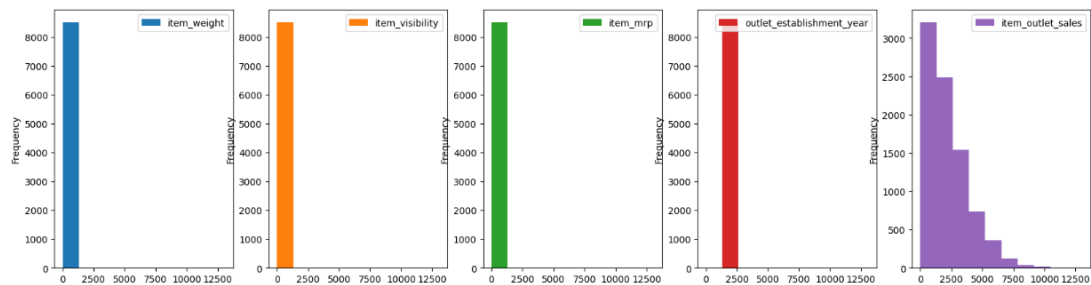
3.6.12. Some important visualizations:

Univariate analysis

Some important visualizations

```
In [78]: 1 #univariate analysis
          2 #finding pattern in data
          3
          4 dataframe.plot(kind='hist', subplots=True, layout=(1,5), figsize=(20,5))
```

```
Out[78]: array([[<AxesSubplot:ylabel='Frequency'>,
                  <AxesSubplot:ylabel='Frequency'>,
                  <AxesSubplot:ylabel='Frequency'>,
                  <AxesSubplot:ylabel='Frequency'>,
                  <AxesSubplot:ylabel='Frequency'>]], dtype=object)
```



Similar data may be found in the dataset for item_weight, item_visibility, item_mrp, and outlet_establishment_year. Even if item_outlet_sales has some really large values, like 10,000, this is possible. Because some festival seasons can see exponential growth in sales.

So, neither will we impute nor will we delete those high number.

Figure 41 Univariate analysis

Similar data may be found in the dataset for item_weight, item_visibility, item_mrp, and outlet_establishment_year.

Even if item_outlet_sales has some really large values, like 10,000, this is possible.

Because some festival seasons can see exponential growth in sales.

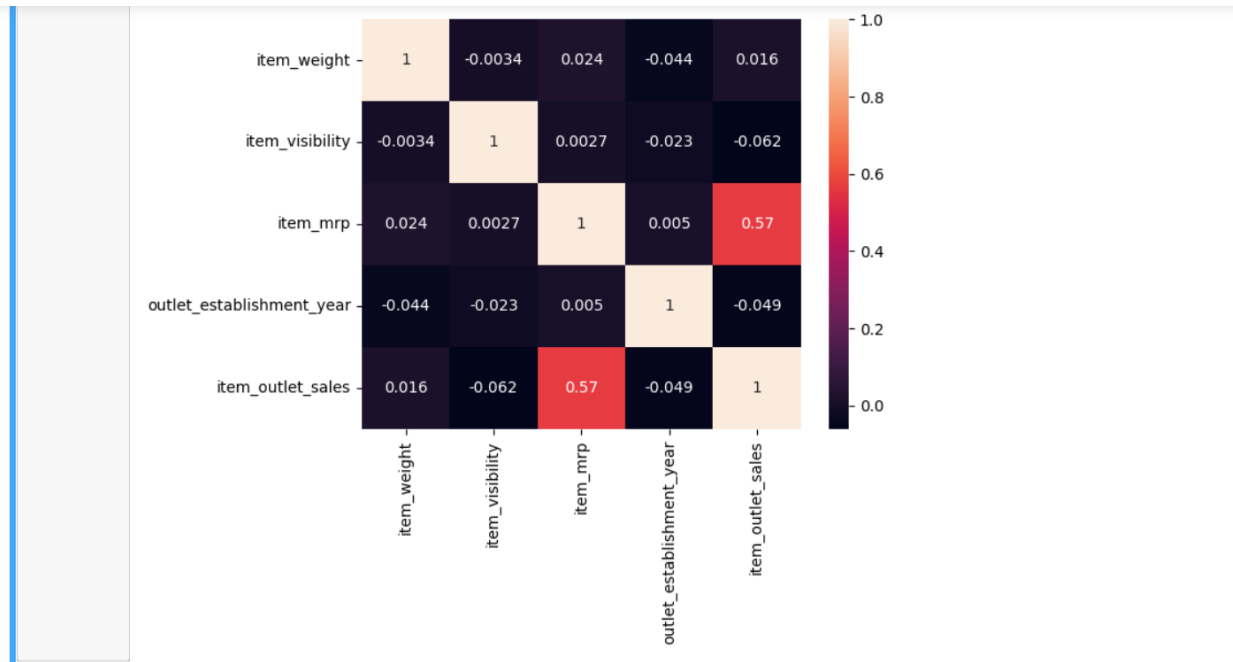
So, neither will we impute, nor will we delete those high numbers.

Bivariate analysis

```
In [80]: 1 #bivariate Analysis
        2 dataframeHeatMap = dataframe.corr() #correlation
        3 sns.heatmap(dataframeHeatMap, annot=True)

Out[80]: <AxesSubplot:>
```

Figure 42 Bivariate analysis



The lighter the color, positively correlated. The darker the color, negatively correlated. Highest correlating value is 0.57.

Figure 43 Heat Map

The lighter the color, positively correlated. The darker the color, the more negatively correlated.

The highest correlating value is 0.57.

Keeping columns

```
In [84]: 1 plt.figure(figsize=(10,10))
          2 sns.barplot(x = dataframe['outlet_identifier'], y = dataframe['item_outlet_sales'])
          3 dataframe['outlet_identifier'].value_counts()

Out[84]: OUT027    935
          OUT013    932
          OUT049    930
          OUT046    930
          OUT035    930
          OUT045    929
          OUT018    928
          OUT017    926
          OUT010    555
          OUT019    528
          Name: outlet_identifier, dtype: int64
```

Figure 44 Keeping column (outlet_identifier)

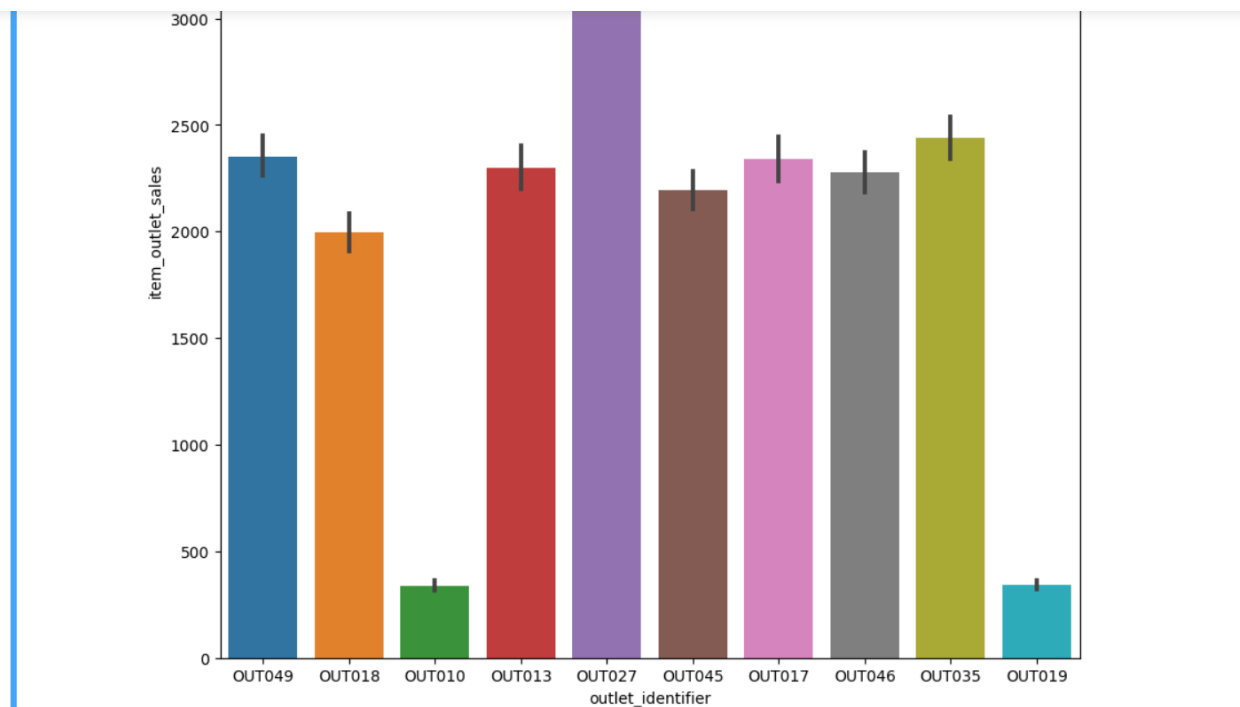


Figure 45 Keeping column (outlet_identifier)2

As a result, we can take into account outlet_identifier for Model creation because there are only a few varieties, and some Outlets have quite high sales.

3.6.13. Dropping useless columns:

```
In [85]: 1 dataframe = dataframe.drop('item_identifier', axis=1)
```

Figure 46 Dropping item_identifier

3.6.14. Encoding:

Encoding

```
In [72]: 1 # to convert categorical text data into model-understandable numerical data, we use the Label Encoder class
2 from sklearn.preprocessing import LabelEncoder
3 label = LabelEncoder()
4 dataframe['item_fat_content'] = label.fit_transform(dataframe['item_fat_content'])
5 dataframe['outlet_size'] = label.fit_transform(dataframe['outlet_size'])
6 dataframe['outlet_location_type'] = label.fit_transform(dataframe['outlet_location_type'])
```

```
In [73]: 1 # to convert categorical text data into model-understandable numerical data
2 # One hot encoding
3 df = pd.get_dummies(dataframe['item_type'])
4 dataframe = pd.concat([dataframe, df], axis=1)
5
6 df = pd.get_dummies(dataframe['outlet_identifier'])
7 dataframe = pd.concat([dataframe, df], axis=1)
8
9 df = pd.get_dummies(dataframe['outlet_type'])
10 dataframe = pd.concat([dataframe, df], axis=1)
11
12 # df = pd.get_dummies(dataframe['item_fat_content'])
13 # dataframe = pd.concat([dataframe, df], axis=1)
14
15 # df = pd.get_dummies(dataframe['outlet_size'])
16 # dataframe = pd.concat([dataframe, df], axis=1)
17
18 # df = pd.get_dummies(dataframe['outlet_location_type'])
19 # dataframe = pd.concat([dataframe, df], axis=1)
```

What one hot encoding does is, it takes a column which has categorical data, which has been label encoded, and then splits the column into multiple columns.

```
In [74]: 1 dataframe.shape
Out[74]: (8523, 41)
```

Figure 47 Encoding

Encoding is done using LabelEncoder() or creating dummies using one hot encoding procedure

What encoding does is convert categorical text data into model-understandable numerical data.

One hot encoding divides a column containing categorical data that has been labelled encoded into numerous columns.


```
In [77]: 1 dataframe = dataframe.drop(['item_type', 'outlet_identfier', 'outlet_type'], axis=1)
         2
```

```
In [78]: 1 dataframe.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8523 entries, 0 to 8522
Data columns (total 38 columns):
#   Column                                Non-Null Count  Dtype
---  -
-----
```

Figure 48 Dropping some columns after encoding

3.6.15. Converting columns to lowercase after encoding:

```
In [80]: 1 #converting dataframe columns to same case - lowercase
        2 dataframe.columns = dataframe.columns.str.lower()

In [81]: 1 dataframe.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8523 entries, 0 to 8522
Data columns (total 38 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   item_weight                           8523 non-null   float64
1   item_fat_content                       8523 non-null   int32
2   item_visibility                        8523 non-null   float64
3   item_mrp                              8523 non-null   float64
4   outlet_establishment_year              8523 non-null   int64
5   outlet_size                           8523 non-null   int32
6   outlet_location_type                   8523 non-null   int32
7   item_outlet_sales                      8523 non-null   float64
8   baking goods                          8523 non-null   uint8
9   breads                                8523 non-null   uint8
10  breakfast                             8523 non-null   uint8
11  canned                                8523 non-null   uint8
12  dairy                                 8523 non-null   uint8
13  frozen foods                          8523 non-null   uint8
14  fruits and vegetables                 8523 non-null   uint8
15  hard drinks                           8523 non-null   uint8
16  health and hygiene                    8523 non-null   uint8
17  household                             8523 non-null   uint8
18  meat                                  8523 non-null   uint8
19  others                                8523 non-null   uint8
20  seafood                               8523 non-null   uint8
21  snack foods                           8523 non-null   uint8
22  soft drinks                           8523 non-null   uint8
23  starchy foods                         8523 non-null   uint8
24  out010                                8523 non-null   uint8
25  out011                                8523 non-null   uint8
```

Figure 49 Converting columns to lowercase after encoding

3.6.17. Feature splitting:

Input and Output Feature Split

```
In [79]: 1 x_orig = dataframe.drop('item_outlet_sales', axis=1) #independent features  
        2 y_orig = pd.DataFrame(dataframe['item_outlet_sales'], columns=['item_outlet_sales']) #dependent features
```

```
In [80]: 1 x_orig.shape
```

```
Out[80]: (8523, 37)
```

```
In [81]: 1 y_orig.shape
```

```
Out[81]: (8523, 1)
```

Figure 50 Feature splitting

x_orig is the independent feature as it has independent variables.

Y_orig is the dependent feature as it has a dependent variable.

3.6.18. Sales prediction using Linear Regression:

Multiple Linear Regression used here as it has multiple independent variables while predicting the result from the model.

```
In [92]: 1 #feature scale
2 #the range of the data is to big
3 #standardizing the data
4 from sklearn.preprocessing import StandardScaler
5
6 scale = StandardScaler()
7 x = scale.fit_transform(x_orig)
8 y = scale.fit_transform(y_orig)
9
10
11 #splitting the dataset into train,test
12 from sklearn.model_selection import train_test_split
13 x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=101)
14
15
16 #fitting the model
17 from sklearn.linear_model import LinearRegression
18 model = LinearRegression()
19 model.fit(x_train, y_train)
20 y_pred = model.predict(x_test)
21
22 y_test = scale.inverse_transform(y_test)
23 y_pred = scale.inverse_transform(y_pred)
24
25
26 #performance metrics
27 from sklearn import metrics
28 print('R2:', metrics.r2_score(y_test, y_pred)) #goodness of fit measure
29 #loss function
30 print('MAE:', metrics.mean_absolute_error(y_test, y_pred))
31 print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
32
R2: 0.5590746152890456
MAE: 822.2765617066194
RMSE: 1096.2127970097497
```

Figure 51 Sales prediction using Linear Regression Model

- i) After feature splitting, feature scaling was performed on the dataset as the range of the data is too big. For standardizing the dataset, feature scaling was done.
- ii) After feature scaling, the dataset was split into train and test.
- iii) After splitting the dataset into train and test, it was fitted into the Linear Regression model.
- iv) Finally, the performance of the model was checked by the performance metrics (R2 score, MAE, RMSE)

3.6.19. Sales prediction using KNN Regressor:

```
In [93]: 1 #feature scale
2 #the range of the data is to big
3 #standardizing the data
4 from sklearn.preprocessing import StandardScaler
5 scale = StandardScaler()
6 x=scale.fit_transform(x_orig)
7 y=scale.fit_transform(y_orig)
8
9
10 #splitting the dataset into train,test
11 from sklearn.model_selection import train_test_split
12 x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=7)
13
14
15 #fitting the model
16 from sklearn.neighbors import KNeighborsRegressor
17 from sklearn import metrics
18 score_max=0
19 rmse_max=0
20 i_max=0
21 for i in range(1,20):
22     model = KNeighborsRegressor(n_neighbors=i)
23     model.fit(x_train, y_train)
24     y_pred = model.predict(x_test)
25
26     y_test = scale.inverse_transform(y_test)
27     y_pred = scale.inverse_transform(y_pred)
28
29     #performance metrics
30     r2 = metrics.r2_score(y_test, y_pred) #goodness of fit measure
31     #Loss function
32     mae = metrics.mean_absolute_error(y_test, y_pred)
33     rmse = np.sqrt(metrics.mean_squared_error(y_test, y_pred))
34     ~
```

Figure 52 Sales prediction using KNN Regressor

```
34
35 #for best neighbors
36 if (rmse > rmse_max):
37     rmse_max=rmse
38     score_max=r2
39     i_max=i
40
41 print('R2:', score_max)
42 print('MAE:', mae)
43 print('RMSE:', rmse_max)
44 print('Neighbor:', i_max)
45
R2: -1.5090299568567342
MAE: 3.2509418928331723e+61
RMSE: 4.19192206387387e+61
Neighbor: 19
```

Figure 53 Sales prediction using KNN Regressor (continued)

- i) After feature splitting, feature scaling was performed on the dataset as the range of the data is too big. For standardizing the dataset, feature scaling was done.
- ii) After feature scaling, the dataset was split into train and test.
- iii) After splitting the dataset into train and test, it was fitted into the KNeighborsRegressor.

iv) Finally, the performance of the model was checked by the performance metrics (R^2 score, MAE, RMSE)

3.7. Comparison of the models used in this project:

Comparison between the models (Linear Regression and KNN Regressor) is done based on the performance metrics.

3.7.1. The performance metrics:

i) R-Squared (R²): R-squared (R²) is a statistical indicator that reveals how much of the variance of a dependent variable in a regression model is explained by one or more independent variables. R-squared, as opposed to correlation, which reflects the strength of the relationship between independent and dependent variables, assesses how well the variation of one variable account for the variance of the second. A model's inputs can therefore explain around half of the observed variation if its R² is 0.50 (Fernando, 2021).

$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2}$$

Figure 54 Formula for R-Squared (R²) (Chugh, 2020)

ii) Mean Absolute Error (MAE): The Mean Absolute Error is a measure of the average absolute difference between the actual and expected values in the dataset. It determines the average residuals for the dataset (Chugh, 2020).

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}|$$

Figure 55 Formula for Mean Absolute Error (MAE) (Chugh, 2020)

iii) Root Mean Square Error (RMSE): The mean squared error is the average of the squared difference between the original and predicted values of the data set. It determines the variance of the residuals. And Root Mean Squared Error is the square root of Mean Squared Error. It determines the standard deviation of the residuals (Chugh, 2020).

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

Figure 56 Formula for Root Mean Square Error (RMSE) (Chugh, 2020)

Where,

\hat{y} – predicted value of y
 \bar{y} – mean value of y

Figure 57 Where (for performance metrics) (Chugh, 2020)

3.7.2. Comparison table:

Algorithms	R-Squared (R2)	Mean Absolute Error (MAE)	Root Mean Square Error (RMSE)
Linear Regression	0.5590746152890456	822.2765617066194	1096.2127970097497
KNN (K-Nearest Neighbor Regressor)	-1.5090299568567342	3.2509418928331723e+61	4.19192206387387e+61

Table 2 Comparison table of the models

Comparing the R-squared (R2) score itself:

Interpretation of R-squared (R2) score

If the R2 score is 0 then the mean line and the regression line is making the same error value which means, the mean line and the regression line are the same.

If the R2 score is 1 then the regression line is not making any error which means the perfect state. (This is not possible)

From the above statement,

The more the model performance moves toward perfection the more the R2 score moves toward 1.

The more the model performance moves away from perfection the more the R2 score moves toward 0.

For negative R2 score,

The regression line is making more errors than the mean line meaning the performance is the least as it can be.

Conclusion of the comparison:

The R2 score of Linear Regression is 0.5590746152890456 which is nearer to 1.

The R2 score of KNN (K-Nearest Neighbor Regressor) is -1.5090299568567342 which is a negative R2 score.

Therefore, **Linear Regression performs better than the KNN (K-Nearest Neighbor Regressor)** for the dataset used in this prediction project.

For the other 2 performance metrics (Mean Absolute Error (MAE), Root Mean Square Error (RMSE)) Linear Regression has better performance scores by far as KNN (K-Nearest Neighbor Regressor) as its performance scores in exponential notation.

Conclusion:

Therefore, Linear Regression is suitable among 2 models (Linear Regression and KNN (K-Nearest Neighbor Regressor)) for this dataset.

4. Conclusion:

4.1. Analysis of the work done:

As traditional methods are not very useful to commercial enterprises in revenue growth, the use of machine learning approaches shows to be a key aspect for building company plans while taking into consideration consumer buying behaviours. Businesses can use sales estimates based on a range of factors, including prior sales, to implement efficient strategies for growing sales and entering the competitive market unafraid. Accurate sales forecasting is necessary for all businesses, but especially for large businesses, it can be difficult due to the many factors that must be considered.

Sales forecasting helps companies manage their cash flow and look for chances to boost profits. Predictive analytics and machine learning algorithms are now essential components of company forecasting. They can be utilized independently or in conjunction with more traditional methods. With both scenarios, the objective is to correctly forecast a company's possible future sales.

4.2. How the solution addresses real-world solution:

Sales forecasting is becoming more and more crucial as e-commerce grows because timely and accurate forecasting can assist e-commerce businesses in resolving all supply and demand-related uncertainty and lowering inventory expenses. Most commercial enterprises heavily rely on data sources and projections of consumer demand for sales trends. The economy is significantly impacted by how accurately sales projections are made. A company can select how to distribute its personnel, cash flow, and resources by using sales forecasting. It is a necessary condition for strategic planning and decision-making in business. It enables companies to effectively plan their corporate strategies. Sales forecasting, then, is the process of estimating future sales based on data that was previously available. Planning for the company's future needs and increasing success chances regardless of external conditions are made possible by

having a thorough awareness of past resources. Businesses that prioritize sales forecasting frequently outperform their competitors. The organization can boost market expansion and revenue generation with accurate forecasting (Akshay Godse, 2019).

The macro and micro planning and management of the retail sector depend on demand forecasting. As the main inputs to multiple decision-making processes in a range of functional areas, including marketing, sales, production/purchasing, as well as finance and accounting, sales forecasts are crucial at the organizational level. Sales forecasts also serve as the foundation for regional and national distribution and replenishment schemes. Accurate retail sales forecasting is crucial for effective inventory management at both the aggregate and disaggregated levels, as well as the relationship between retail stocks and sales at the aggregate level, which has long been recognized (Akshay Godse, 2019).

Sales forecasting influences every department of the company and almost every strategic decision-making. The finance department, for instance, uses it to schedule its quarterly and yearly investments. Product managers use the forecast to predict consumer demand for new products, while HR uses it to align hiring strategies with company growth goals. Because business settings are changing so quickly, being able to accurately predict is more important than ever. The more precise the sales predictions, the quicker business can adapt to changing conditions. Accurate sales projections are crucial for a company to survive. Having a reliable sales forecasting system in place is crucial for the company's future (Mahalingam, n.d.).

4.3. Further work:

Further, the project can be used for the research study of which algorithm model is suited for which dataset. The result showed which machine learning algorithm was more accurate among those mentioned above to forecast sales. More machine learning algorithms can be compared in a manner and the most suitable algorithm can be used to forecast sales accurately. GUI can be implemented for the user's convenience to forecast sales in different sectors.

Bibliography

Akshay Godse, P. P. S. S. S. M. P. K., 2019. INTELLIGENT SALES PREDICTION USING MACHINE LEARNING TECHNIQUES. *INTERNATIONAL JOURNAL OF RESEARCH IN ELECTRONICS AND COMPUTER ENGINEERING*, 7(4).

Alves, S., 2021. *Medium*. [Online]

Available at: <https://medium.com/@sergioalves94/walmart-store-sales-forecasting-4ffebbb650f>

[Accessed 10 January 2023].

Anaplan, n.d. *Anaplan*. [Online]

Available at: <https://www.anaplan.com/blog/sales-forecasting-guide/>

[Accessed 9 December 2022].

Bajaj, P., 2020. SALES PREDICTION USING MACHINE LEARNING ALGORITHMS. *International Research Journal of Engineering and Technology (IRJET)* , 7(6).

Bishop, K., 2020. *Sales Hacker*. [Online]

Available at: <https://www.saleshacker.com/sales-forecasting-101/>

[Accessed 9 December 2022].

Burns, E., 2021. *TechTarget*. [Online]

Available at: <https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML>

[Accessed 7 December 2022].

Burns, E., 2022. *TechTarget*. [Online]

Available at: <https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence>

[Accessed 7 December 2022].

Chugh, A., 2020. *Medium*. [Online]

Available at: <https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e>

[Accessed 10 January 2023].

Coursera, 2022. *Coursera*. [Online]

Available at: <https://www.coursera.org/articles/types-of-machine-learning>

[Accessed 7 December 2022].

Driscoll, M., n.d. *Real Python*. [Online]

Available at: <https://realpython.com/jupyter-notebook-introduction/>

[Accessed 10 January 2023].

Fernando, J., 2021. *Investopedia*. [Online]

Available at: <https://www.investopedia.com/terms/r/r-squared.asp>

[Accessed 10 January 2023].

- Frankenfield, J., 2022. *Investopedia*. [Online]
Available at: <https://www.investopedia.com/terms/a/artificial-intelligence-ai.asp>
[Accessed 7 December 2022].
- Galt, J., n.d. *John Galt*. [Online]
Available at: <https://johngalt.com/learn/blog/3-advantages-disadvantages-of-forecasting>
[Accessed 11 December 2022].
- Hayes, A., 2022. *Investopedia*. [Online]
Available at: <https://www.investopedia.com/terms/m/mlr.asp>
[Accessed 10 January 2023].
- JavaTPoint, n.d. *JavaTPoint*. [Online]
Available at: <https://www.javatpoint.com/regression-analysis-in-machine-learning>
[Accessed 11 December 2022].
- Kanade, V., 2022. *Spiceworks*. [Online]
Available at: <https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-ai/>
[Accessed 7 December 2022].
- Kharwal, A., 2021. *thecleverprogrammer*. [Online]
Available at: <https://thecleverprogrammer.com/2021/05/19/sales-prediction-with-machine-learning/>
[Accessed 10 January 2023].
- Mahalingam, K., n.d. *Chargebee*. [Online]
Available at: <https://www.chargebee.com/blog/importance-of-sales-forecasting/>
[Accessed 13 December 2022].
- Mahalingam, K., n.d. *Chargebee Blog*. [Online]
Available at: <https://www.chargebee.com/blog/importance-of-sales-forecasting/>
[Accessed 11 December 2022].
- Mansi Panjwani, R. R. H. J. K. Z. R. H., 2020. Sales Prediction System Using Machine Learning. *Sales Prediction System Using Machine Learning*, 23 April.
- MathWorks, n.d. *MathWorks*. [Online]
Available at: <https://www.mathworks.com/discovery/reinforcement-learning.html>
[Accessed 7 December 2022].
- Naeem, A., 2022. *Robotics Embedded*. [Online]
Available at: <https://www.embedded-robotics.com/forecast-sales-using-machine-learning/>
[Accessed 10 January 2023].
- Qu F, W. Y.-T. H. W.-H. Z. X.-Y. W. X.-K. L. J.-B. W. J.-Q., 2022. Forecasting of Automobile Sales Based on Support Vector Regression Optimized by the Grey Wolf Optimizer Algorithm. *Mathematics*. 2022, 10(13).

Rijal, S., 2021. *My Republica*. [Online]
 Available at: <https://myrepublica.nagariknetwork.com/mycity/news/is-ai-dangerous-or-beneficial-for-humankind>
 [Accessed 7 December 2022].

saedsayad, n.d. *saedsayad*. [Online]
 Available at: https://www.saedsayad.com/k_nearest_neighbors_req.htm
 [Accessed 11 December 2022].

Sai Nikhil Boyapati, R. M., 2020. [Online]
 Available at: <https://www.diva-portal.org/smash/get/diva2:1455353/FULLTEXT02>
 [Accessed 10 January 2023].

Saint, J., 2020. *Medium*. [Online]
 Available at: <https://medium.com/@jagwithyou/linear-regression-with-sales-prediction-project-8152e7de2cf2>
 [Accessed 11 December 2022].

Schmidt, A., 2022. *MDPI*. [Online]
 Available at: <https://www.mdpi.com/2504-4990/4/1/6>
 [Accessed 10 December 2022].

Selig, J., 2022. *expert.ai*. [Online]
 Available at: <https://www.expert.ai/blog/machine-learning-definition/>
 [Accessed 7 December 2022].

Shelke, M. R. R., Dharaskar, D. R. V. & Thakare, D. V. M., 2017. Data Mining For Supermarket Sale Analysis Using Association Rule. *International Journal of Trend in Scientific Research and Development*, Volume 1(4), ISSN: 2456-6470, Volume 1 .

Smolic, H., 2022. *Graphite*. [Online]
 Available at: <https://graphite-note.com/machine-learning-sales-forecasting>
 [Accessed 10 December 2022].

Soham Patangia, K. S. M. M. R. M. G. K. P. R., 2020. Sales Prediction of Market using Machine Learning. *International Journal of Engineering Research & Technology (IJERT)*, 9(9).

Stuti Raizada, J. R. S., 2021. Comparative Analysis of Supervised Machine Learning Techniques for Sales Forecasting. *(IJACSA) International Journal of Advanced Computer Science and Applications*, 12(11).

Team, G. L., 2022. *Great Learning Team*. [Online]
 Available at: <https://www.mygreatlearning.com/blog/how-machine-learning-is-used-in-sales-forecasting/>
 [Accessed 10 December 2022].

Terra, J., 2022. *Simple Learn*. [Online]
Available at: https://www.simplilearn.com/regression-vs-classification-in-machine-learning-article#the_difference_between_regression_vs_classification
[Accessed 11 December 2022].

Wilson, J., 2022. *technopedia*. [Online]
Available at: <https://www.techopedia.com/definition/3533/python>
[Accessed 10 January 2023].

Wong, L., n.d. *Apruve*. [Online]
Available at: <https://blog.apruve.com/how-to-forecast-sales-for-your-new-product-manual-predictions-vs-machine-based-analysis#:~:text=Manual%20product%20forecast%20comes%20with,Time%20is%20money!>
[Accessed 10 December 2022].