Business forecasting using machine learning

END SEMESTER Report
Submitted in fulfillment of the requirements of
CS F376 Design Project
By

Siddharth Choudhury
IDNO: 2020A7PS0028U

Under the supervision of
Dr. Siddhaling Urolagin
Professor

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

DUBAI CAMPUS, DUBAI UAE

DECEMBER - 2022

ACKNOWLEDGEMENTS

I would like to express my deepest sense of gratitude, first and foremost, to my Supervisor **Dr. Siddhaling Urolagin**, Professor, Computer Science Department, BITS Pilani, Dubai campus, United Arab Emirates, for her valuable guidance and encouragement during the course of this Project. I am extremely grateful to him for his able guidance, valuable technical inputs and useful suggestions.

I express my sincere thanks and gratitude to our Director, BITS Pilani, Dubai Campus, **Prof. Dr. Srinivasan Madapusi,** for their motivation, encouragement and support to pursue my Project.

I am grateful to examiners **Dr. Siddhaling Urolagin** for their valuable suggestions.

Above all, I thank the Lord for giving me the strength to carry out this work to the best of my abilities.

Name : Siddharth Choudhury
ID No. : 2020A7PS0028U

CERTIFICATE

This is to certify that the Mid Semester Project Report entitled, in partial fulfillment **Business forecasting using machine learning** of the requirement of CS F376 Design Project embodies the work done by him under my supervision.

Date:

Signature of the Supervisor

Name: Dr Siddhaling Urolagin
Designation: Professor (CS)

BITS Pilani, Dubai Campus

FIRST Semester 2022-2023

Project Course Code and Course Name:
Semester: First Semester 2022-2023
Duration: 10.09.2022-10.01.2023
Date of Start: 10.09.2022
Date of Submission: 17.12.2022
Title of the Report: Business Forecasting Using machine learning

ID No. / Name of the student: **2020A7PS0028U / SIDDHARTH CHOUDHURY**

Discipline of Student: B.E (Hons.) Computer Science

Name of the Project Supervisor: Dr SIDDHALING UROLAGIN

Key Words: **Machine Learning, Heat maps, Categorial data plots, Correlation plots, Data preprocessing**

Project Area: **Machine Learning**

Abstract: **Business forecasting is critical for retailers since it is necessary for a variety of operational choices. Forecasting demand on special days, when demand patterns are very different from those on typical days, is one of the biggest challenges. We discuss the issue of predicting the daily demand for various product categories at the shop level using the example of a supermarket chain. These projections serve as a guide for purchasing and manufacturing decisions. We address the forecasting issue using machine learning. We describe and talk about the potential of creating a classification problem rather than a regression problem in specific. Machine learning techniques outperform traditional methods empirically, whereas classification-based approaches outperform regression-based approaches. We also discovered that machine learning techniques are better suited for use in a sizeable demand forecasting scenario that frequently happens in the retail sector, in addition to offering more accurate forecasts.**

Signature of Student **Signature of Supervisor**

Date: 17/12/2022 **Date: 17/12/2022**

TABLE OF CONTENTS

BUSINESS FORECASTING USING MACHINE LEARNING

INTRODUCTION

Sales forecasting has always been a very important topic to focus on. All vendors must now anticipate well and optimally in order to maintain the effectiveness of marketing groups.
Manually performing this work would be time-consuming, which is undesirable in today's fast-paced environment and could result in grave mistakes that would result in bad management of the firm. A significant portion of the world economy is dependent on the business sectors, which are literally expected to generate enough goods in the right amounts to satisfy demand.

The primary objective of business sectors is market audience targeting. It is crucial that the business has been successful in achieving this goal by utilizing a forecasting system. Forecasting requires examining data from a variety of sources, including market trends, customer behavior, and other elements. The companies would benefit from this analysis by having better financial resource management. The forecasting method can be used for a variety of things, such as estimating future demand for the product or service or estimating how much of the product will be sold in a specific time frame.

Here, machine learning has a lot of potential for use. In the field of machine learning, computers are able to execute some jobs better than people. They are employed to carry out specific tasks in a methodical manner and produce improved outcomes for the advancement of the modern civilization.
The foundation of machine learning is mathematics, which may be used to design various paradigms that are close to the ideal output. Machine learning has been shown to be beneficial in the instance of sales forecasting. It aids in more precise forecasting of upcoming sales.

In this project report, we suggested machine learning techniques for data gathered from a grocery store's prior sales. Based on a few key characteristics identified from the available raw data, the goal is to predict the sales pattern and the quantities of the products to be sold. To fully understand the data, analysis and study of the acquired data have also been done. At each crucial stage of the marketing strategy, analysis would assist business organizations in arriving at a probable decision.

DATA VISUALIZATION

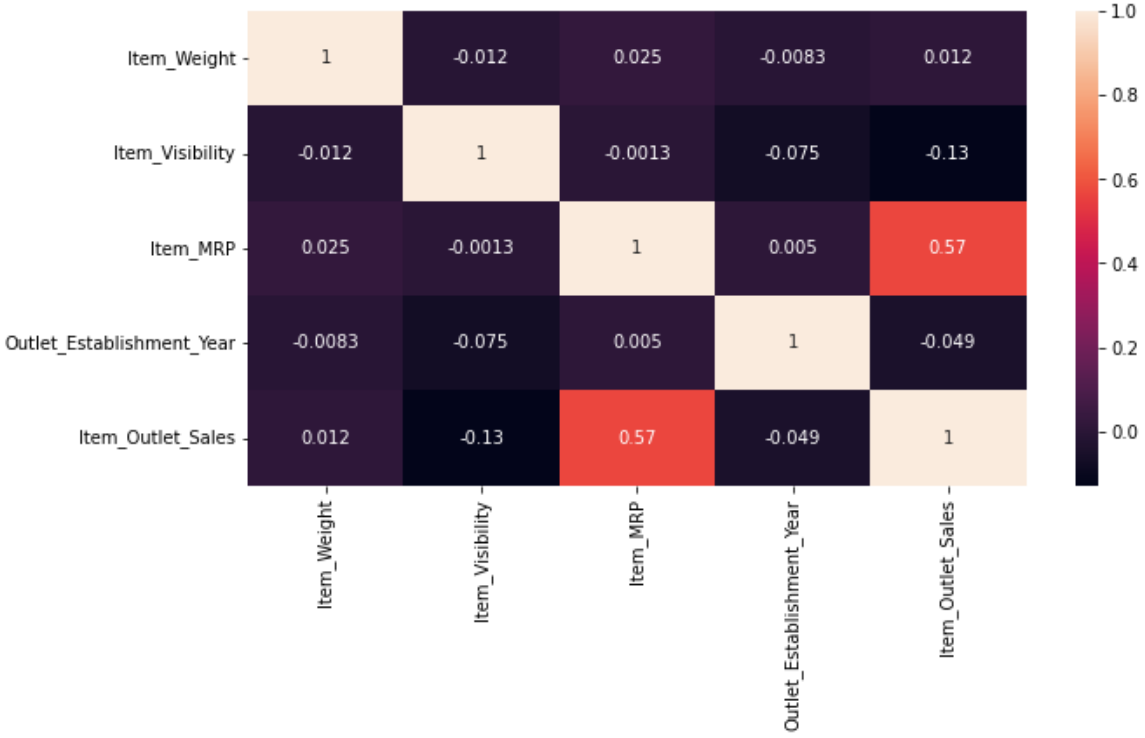
HEAT MAP

Heat map for figuring out the relationships between the dataset's characteristics.

Here, the correlation between the target variable and the other qualities is shown using a heat map, a color-coded matrix from the Seaborn data visualization toolkit.

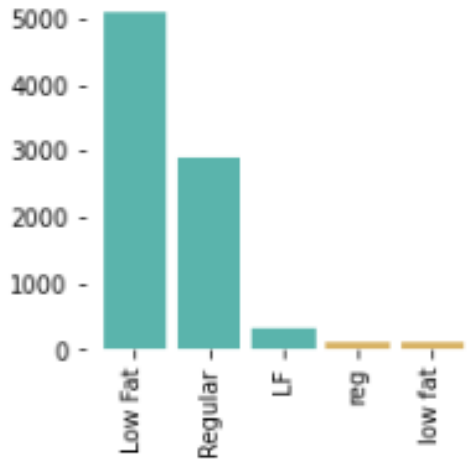
The target variable's dependence on an attribute decreases as the color intensity of the attribute's relative to the target variable increases.

The goal variable, Item_Outlet_Sales, is seen to be most dependent on item MRP and least dependent on Item_Visibility. Therefore, higher the MRP of an item, lower will be the Item_Outlet_Sales.

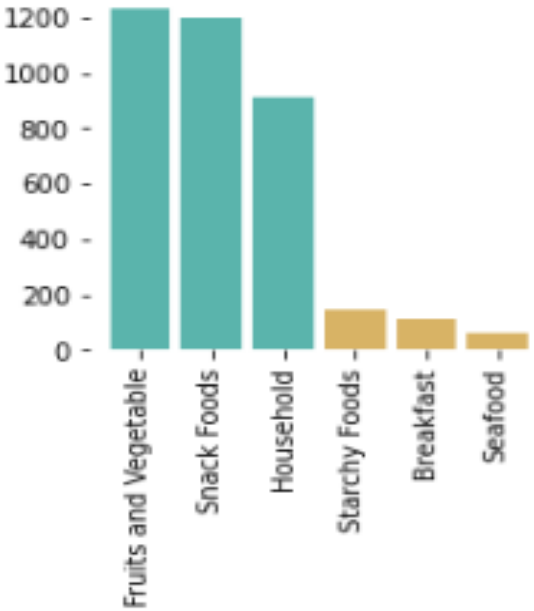


CATEGORIAL DATA PLOTS

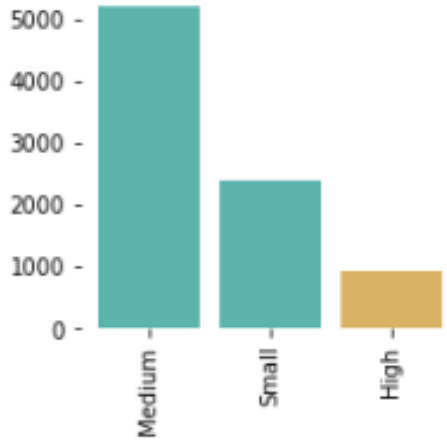
The distribution of various Item fat content i.e., Low Fat and Regular Fat are written in distinct ways in the categorial data plots. It is observed that maximum items have low fat content.



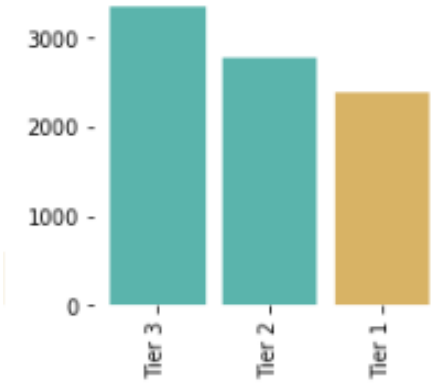
The following figure shows how each item kind is distributed. Fruits and vegetables make up the majority of the goods, followed by snack foods. Seafood, in comparison, is least in number.



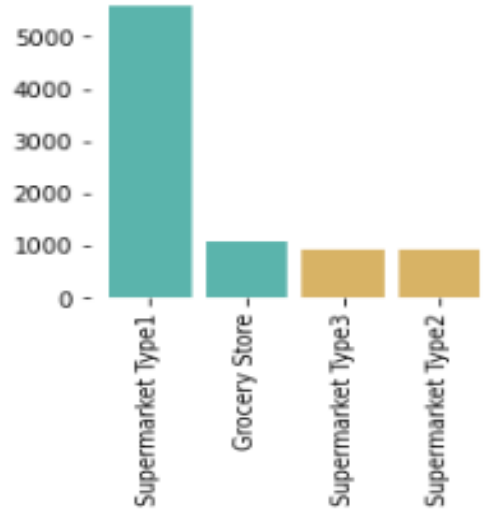
Very few of the outlets are high or large in size, whereas the majority are medium in size.



According to the statistics, there are three types of outlet locations: Tier1, Tier2, and Tier3. The Tier3 location type has the most outlets.

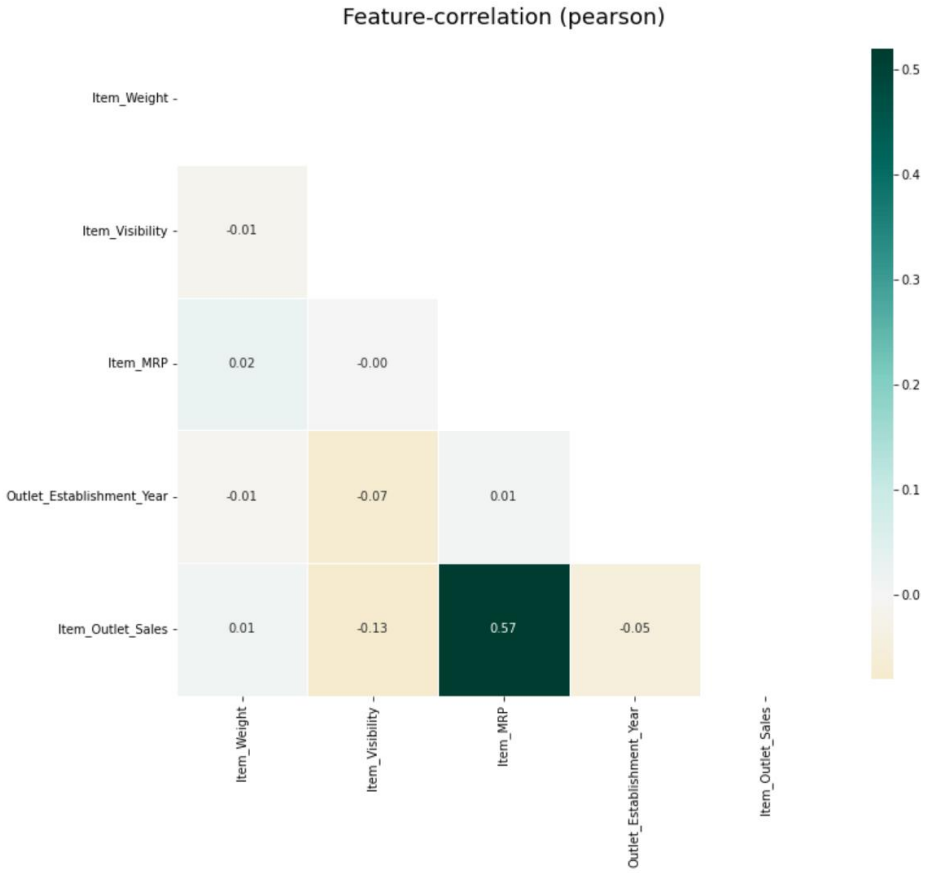


Plotted is the distribution of the several outlet types, such as Supermarket Type1, Supermarket Type2, Grocery Store, and Supermarket Type3. It has been noted that Supermarket Type 1 outlets are in majority



FEATURE-CORRELATION PLOTS

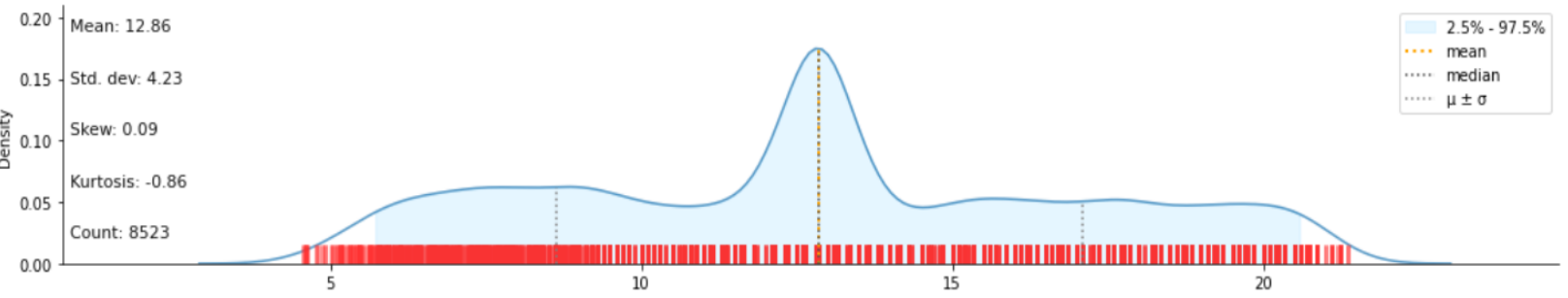
Correlation plots are used to understand which variables are connected to each other and the strength of this relationship. A correlation plot often has a number of numerical variables, with each variable represented by a column. The relationships between each pair of variables are shown by the rows. Positive values indicate a positive association, while negative values indicate a negative relationship. The values in the cells represent the strength of the relationship. You may use correlation heatmaps to identify possible links between variables and to gauge how strong these associations are.

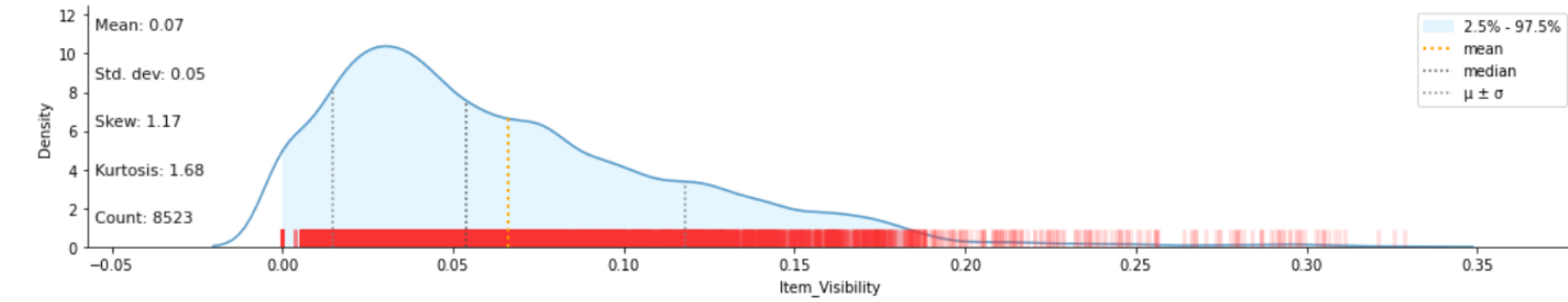


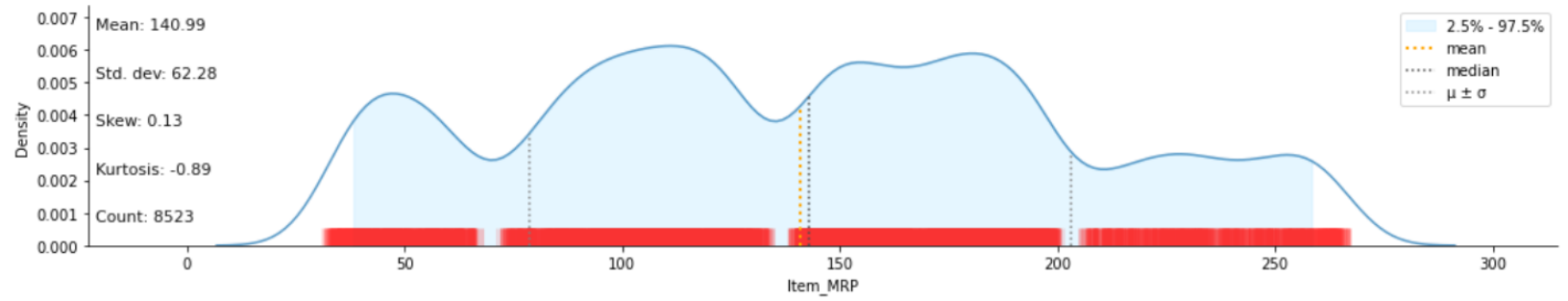
DISTRIBUTION PLOTS

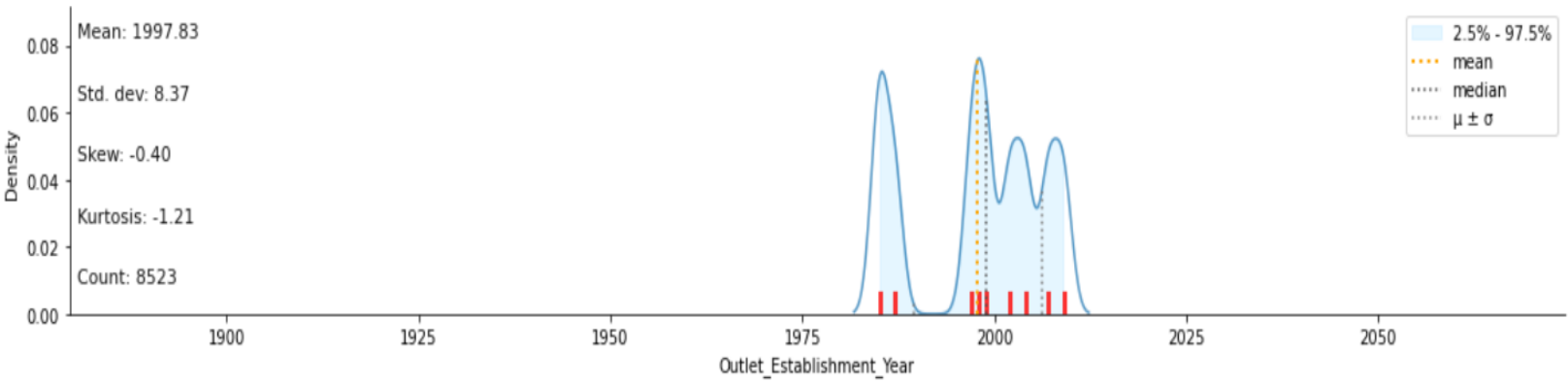
Distribution plots visually analyze the distribution of sample data by comparing the data's actual distribution to the theoretical values anticipated from a certain distribution. To ascertain if

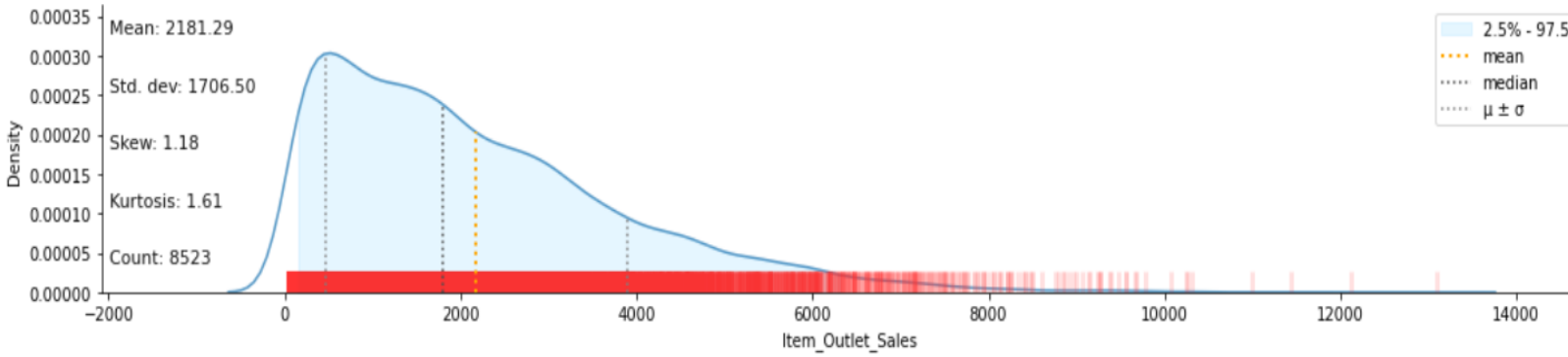
the sample data belongs to a certain distribution, use distribution plots in addition to more formal hypothesis testing. The mean, median, standard deviation, and other statistics are also provided.








Mean: 1997.83
Std. dev: 8.37
Skew: -0.40
Kurtosis: -1.21
Count: 8523
2.5% - 97.5%
mean
median
μ ± σ
Density
0.00
0.02
0.04
0.06
0.08
1900
1925
1950
1975
2000
2025
2050
Outlet_Establishment_Year


Mean: 2181.29
Std. dev: 1706.50
Skew: 1.18
Kurtosis: 1.61
Count: 8523
2.5% - 97.5
mean
median
μ ± σ
Density
0.00000
0.00005
0.00010
0.00015
0.00020
0.00025
0.00030
0.00035
−2000
0
2000
4000
6000
8000
10000
12000
14000
Item_Outlet_Sales

DATA PREPROCESSING

In machine learning algorithm, data can't be used in its normal form as it is the as the way it is obtained, so the data needs to be devised before employing it in machine learning models. This technique is used to solve problems that are not yet known by the knowledge extractor. This is called preprocessing work. The goal of preprocessing is to find out what kind of information the algorithm needs before making any decisions about whether to use it or not.

Preprocessing requires clean, well formatted data. The following tasks are included in data preprocessing:

1) Importing the dataset: To check the potential sales or demand of an item outlet, we used the dataset gathered from a grocery store in our study. It has the following characteristics:

   Item_identifier, Item_weight, Item_fat_content, Item_visibility, Item_type, Item_MRP, Outlet_identifier, Outlet_establishment_year, Outlet_size, Outlet_location_type, Outlet_type, Item_Outlet_sales

   The dataset file is saved as a CSV file before being imported.

2) As a part of data cleaning, it is necessary to delete some columns that don't help the algorithm reach its conclusions. Item_identifier and Outlet_identifier are removed in this case.

3) Handling missing values: Data gaps are something that must be changed to ensure that there is no disparity in the data that will be used to feed the model. It's here a few values in the fields Item_weight and Outlet_size were absent. Outlet size is an example of where the entire row has been dropped together with missing value cases and in the event of the mean of all the missing spaces for item weight is used. The other columns' entries.

4) Data Integration: One of the data preparation procedures called data integration is used to combine data from several sources into a single, bigger data storage, such as a data warehouse.

5) Data Transformation: Once the data has been cleared, we must use data transformation techniques to change the value, structure, or format of the data in order to combine the quality data into other forms.

6) Generalization: We used idea hierarchies to translate low-level or granular data to high-level information. We can transform the primitive data in the address like the city to higher-level information like the country.

7) Normalization: It is the most significant and extensively used data transformation method. Depending on the range, the numerical attributes are scaled up or down. In this method, we limit our data attribute to a certain container to create a correlation between several data points.

CLASSIFIERS

RANDOM FOREST

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is controlled with the max_samples parameter if bootstrap=True (default), otherwise the whole dataset is used to build each tree.

REGRESSION

Linear regression is a simple and common type of predictive analysis. Linear regression attempts to model the relationship between two (or more) variables by fitting a straight line to the data. Put simply, linear regression attempts to predict the value of one variable, based on the value of another (or multiple other variables).

LINEAR REGRESSION

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables, they are considering and the number of independent variables being used. This article is going to demonstrate how to use the various Python libraries to implement linear regression on a given dataset.

BACKEND

KLIB

Klib is a Python library that provides amazing functionality for exploring your data in just a few lines of code. If you find that data exploration takes a lot of time, you can use this library as it gives you all the functions that will help you to explore, clean and prepare your data.

Most data scientists go through the same process while exploring the data they use to gain insight. Some of the common steps used by all data scientists while exploring a dataset are:

1. check whether there are missing values or not
2. understand the distribution of all the features
3. understanding the categorical features
4. understanding the correlation between the features of the data

After these steps, you may need to change the way you explore your datasets depending on the type of problem you are working on and the type of results you are looking for. But to get to this point, you need to explore your data to understand the type of data you are using. Sometimes it takes a long time to explore your dataset, this is where the Klib library in Python comes in. It helps you in exploring your data in just a few lines of code. In the section below, I'll show you a tutorial on the Klib library in Python to explore your data.

- **klib.missingval_plot(df_train)**

  returns a figure containing information about missing values.

- **klib.data_cleaning(df_train)**

  performs data cleaning (drop duplicates & empty rows/columns, adjust data types and much more.

- **klib.clean_column_names(df_train)**

  cleans and standardizes column names, also called inside data_cleaning()

- **df_train.info()**

  Pandas dataframe.info() function is used to get a concise summary of the dataframe. It comes really handy when doing exploratory analysis of the data.

- **klib.convert_datatypes(df_train)**

  converts existing to more efficient dtypes, also called inside data_cleaning()

- **klib.mv_col_handling(df_train)**

  drops features with high ratio of missing values based on informational content

scikit-learn

scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

- **from sklearn.preprocessing import LabelEncoder**

  LabelEncoder is a utility class to help normalize labels such that they contain only values between 0 and n_classes-1. This is sometimes useful for writing efficient Cython routines. It can also be used to transform non-numerical labels to numerical labels.

- **le.fit_transform(df_train['item_type']**

  fit_transform() is used on the training data so that we can scale the training data and also learn the scaling parameters of that data.

- **from sklearn.model_selection import train_test_split**

  Using train_test_split() from the data science library scikit-learn, you can split your dataset into subsets that minimize the potential for bias in your evaluation and validation process.

- **from sklearn.preprocessing import StandardScaler**

  The StandardScaler function of sklearn is based on the theory that the dataset's variables whose values lie in different ranges do not have an equal contribution to the model's fit parameters and training function and may even lead to bias in the predictions made with that model. Therefore, before including the features in the machine learning model, we must normalize the data ($\mu = 0$, $\sigma = 1$). Standardization in feature engineering is commonly employed to address this potential issue.

- **from sklearn.linear_model import LinearRegression**

  LinearRegression fits a linear model with coefficients w = (w1, ..., wp) to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation. A linear regression model minimizes the mean squared error on the training set. This means that the parameters obtained after the fit (i.e. `coef_` and `intercept_`) are the optimal parameters that minimizes the mean squared error. In other words, any other choice of parameters will yield a model with a higher mean squared error on the training set.

- **lr.predict(X_train_std)**

  Python predict() function enables us to predict the labels of the data values on the basis of the trained model. The predict() function accepts only a single argument which is usually the data to be tested.

- **from sklearn.metrics import r2_score**

  The R2 score is a very important metric that is used to evaluate the performance of a regression-based machine learning model. It is pronounced as R squared and is also known as the coefficient of determination. It works by measuring the amount of variance in the predictions explained by the dataset. Simply put, it is the difference between the samples in the dataset and the predictions made by the model.

- **from sklearn.metrics import mean_absolute_error**

  Mean absolute error refers to the magnitude of difference between the prediction of an observation and the true value of that observation. It takes the average of absolute errors for a group of predictions and observations as a measurement of the magnitude of errors for the entire group.

- **sklearn.metrics import mean_squared_error**

  The Mean Squared Error measures how close a regression line is to a set of data points. It is a risk function corresponding to the expected value of the squared error loss. Mean square error is calculated by taking the average, specifically the mean, of errors squared from data as it relates to a function.

- **from sklearn.ensemble import RandomForestRegressor**

  A random forest is an ensemble model that consists of many decision trees. Predictions are made by averaging the predictions of each decision tree. Or, to extend the analogy—much like a forest is a collection of trees, the random forest model is also a collection of decision tree models. This makes random forests a strong modeling technique that's much more powerful than a single decision tree.

- **from sklearn.model_selection import RepeatedStratifiedKFold**

  RepeatedStratifiedKFold allows improving the estimated performance of a machine learning model, by simply repeating the cross-validation procedure multiple times (according to the n_repeats value), and reporting the mean result across all folds from all runs.

- **from sklearn.model_selection import GridSearchCV**

  GridSearchCV is the process of performing hyperparameter tuning in order to determine the optimal values for a given model. As mentioned above, the performance of a model significantly depends on the value of hyperparameters. Note that there is no way to know in advance the best values for hyperparameters so ideally, we need to try all possible values to know the optimal values. Doing this manually could take a considerable amount of time and resources and thus we use GridSearchCV to automate the tuning of hyperparameters.

FRONTEND

FLASK

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

HTML

The HyperText Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

Header Tags

The <header> element represents a container for introductory content or a set of navigational links.

A <header> element typically contains:

- one or more heading elements (<h1> - <h6>)
- logo or icon
- authorship information

Body Tags

The <body> tag defines a web page content (text, images, links, etc.). It is placed inside the <html> element, after the <head> element. In an HTML document, we can use only one <body> tag.

Commonly, a list of content-specific CSS classes is placed within the <body> element allowing JavaScript developers and designers to target pages easily. Even if these classes are not used, they won't cause any problems.

Paragraph Element

The HTML element represents a paragraph. Paragraphs are usually represented in visual media as blocks of text separated from adjacent blocks by blank lines and/or first-line indentation, but HTML paragraphs can be any structural grouping of related content, such as images or form fields.

Paragraphs are block-level elements, and notably will automatically close if another block-level element is parsed before the closing tag.

div Tag

The HTML element is the generic container for flow content. It has no effect on the content or layout until styled in some way using CSS (e.g. styling is directly applied to it, or some kind of layout model like Flexbox is applied to its parent element).

As a "pure" container, the element does not inherently represent anything. Instead, it's used to group content so it can be easily styled using the class or id attributes, marking a section of a document as being written in a different language (using the lang attribute), and so on.

Main Section

The <main> HTML element represents the dominant content of the <body> of a document. The main content area consists of content that is directly related to or expands upon the central topic of a document, or the central functionality of an application.

A document mustn't have more than one <main> element that doesn't have the hidden attribute specified.

Footer Section

The <footer> HTML element represents a footer for its nearest ancestor sectioning content or sectioning root element. A <footer> typically contains information about the author of the section, copyright data or links to related documents.

Form Element

HTML form elements are used to capture user input. There are many different types of form elements such as the text box, check box, drop down, submit button, and much more.

Button Element

The <button> HTML element is an interactive element activated by a user with a mouse, keyboard, finger, voice command, or other assistive technology. Once activated, it then performs an action, such as submitting a form or opening a dialog.

By default, HTML buttons are presented in a style resembling the platform the user agent runs on, but you can change buttons' appearance with CSS.

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of content and presentation, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

Google Fonts

Google Fonts is a computer font and web font service owned by Google. This includes free and open-source font families, an interactive web directory for browsing the library, and APIs for using the fonts via CSS and Android.

Integrated Development Environment

VS Code

Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

JUPYTER NOTEBOOK

Jupyter Notebook (formerly IPython Notebook) is a web-based interactive computational environment for creating notebook documents. Jupyter Notebook is built using several open-source libraries, including IPython, ZeroMQ, Tornado, jQuery, Bootstrap, and MathJax. A Jupyter Notebook application is a browser-based REPL containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media.

Jupyter Notebook is similar to the notebook interface of other programs such as Maple, Mathematica, and SageMath, a computational interface style that originated with Mathematica in the 1980s. Jupyter interest overtook the popularity of the Mathematica notebook interface in early 2018.

JupyterLab is a newer user interface for Project Jupyter, offering a flexible user interface and more features than the classic notebook UI. The first stable release was announced on February 20, 2018. In 2015, a joint $6 million grant from The Leona M. and Harry B. Helmsley Charitable Trust, The Gordon and Betty Moore Foundation, and The Alfred P. Sloan Foundation funded work that led to expanded capabilities of the core Jupyter tools, as well as to the creation of JupyterLab.

Result And Analysis

The performance of the classification algorithms is mostly focused on Classification accuracy, Accuracy in each class and confusion matrix which shows the number of predictions of each class which can be compared to the instances of each class. Root Mean Square Error, Mean Square Error, Absolute error are calculated and average of the error is shown as the Error Rate. This measure helps to identity whether the given prediction is wrong on average.

CONCLUSION

Demand forecasting is one of the major challenges faced by supply chains in the retail sector when trying to maximize stock levels, save costs, and boost revenue, profits, and customer loyalty. The solution to this problem is to examine and understand complex relationships and patterns from historical data using techniques like time series analysis and machine learning. In other words, it is critical to have the capability to determine what customers will buy, when they'll need it, and how much they will demand from a specific retailer or store. In order to mitigate the risks of forecasting errors, it is critical for supply chains to have a solid understanding of their end customers and how this understanding will allow them to manage uncertainty when forecasting demand from consumers. This is typically a three-part process in which data scientists first analyze and understand historical demand patterns, then use mathematical modeling to determine the impacts of various factors on the forecasted demand, and finally evaluate the accuracy of their forecasts using historical data and live data.
In this study we have used machine learning techniques to predict the sales of a particular item in a store. It will highly beneficial for increasing profits and sales of the store.

REFERENCES

[1] Huber, Jakob, and Heiner Stuckenschmidt. "Daily retail demand forecasting using machine learning with emphasis on calendric special days." *International Journal of Forecasting* 36.4 (2020): 1420-1438.

[2] Zhu, Xiaodan, et al. "Demand forecasting with supply-chain information and machine learning: Evidence in the pharmaceutical industry." *Production and Operations Management* 30.9 (2021): 3231-3252.

[3] Kilimci, Zeynep Hilal, et al. "An improved demand forecasting model using deep learning approach and proposed decision integration strategy for supply chain." *Complexity* 2019 (2019).

[4] Böse, Joos-Hendrik, et al. "Probabilistic demand forecasting at scale." *Proceedings of the VLDB Endowment* 10.12 (2017): 1694-1705.