Міністерство освіти і науки України
Національний університет «Одеська політехніка»
Інститут комп’ютерних систем
Кафедра інформаційних систем

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на тему:

«Аналіз продажів. Прогнозування знижок на товари. Аналіз тональності коментарів. Побудова системи рекоменацій»

Виконав студент
групи АІ221
Попазов П.І.

Перевірили:
Антощук С.Г.
Смик С.Ю.

Одеса-2024

CONTENT

INTRODUCTION

Analysis of sales, including the construction of graphs, forecasting discounts, and analysis of comments, is critically important for running a successful business, as these tools allow for a deep understanding of customer behavior, optimize sales strategies, and make well-founded decisions regarding product policy.

Firstly, analyzing comments left under a product allows you to understand what customers like or dislike about the products. Analyzing negative reviews allows you to identify problems with the product or service and improve the product based on them. Determining the tone of comments helps to get a general picture of customer satisfaction and determine whether actions are needed to improve the experience. Managers can track or filter negative comments to connect with the buyer and find out the cause of dissatisfaction, which allows to improve the company's reputation.

On the other hand, sales analysis allows you to understand which categories of goods are most popular among customers, which allows you to create a product assortment that better meets the needs of the market and individual customers, as well as identify goods that have low demand. Analysis of purchase history helps to create personalized offers, which increases customer loyalty and encourages repeat purchases, allowing better satisfaction of consumer needs and increasing sales volume.

Thirdly, discount forecasting allows you to determine the reduced price of goods based on factors such as actual price, product rating, and number of reviews. This allows businesses to predict which discount should be applied to the product to maximize sales or improve competitiveness.

It allows to personalize discounts by considering the popularity and reviews of goods. Helps in planning promotions, sales, and other marketing campaigns.

1 Theoretical information

1.1 Task to perform

A defined task to accomplish for an individual task:

1. Pre -processing and preparation of data. Cleaning data from missing values, anomalies and duplicate. Normalization and standardization of variables to ensure the correctness of further analysis.

2. Data Visualization. Creating graphs such as histograms, box plot, scatter plot, and heatmap to identify patterns, correlations, and trends. Visualization of dependencies between product categories, prices, ratings, and sales volumes. Definition of the number of items by categories. Identification of top 5 items with the largest discount. Analysis of review length.

3. Comment analysis. Determining the mood (positive, negative, neutral) using text processing tools. Preprocessing of text: removal of irrelevant parts (URL addresses, special symbols), tokenization, lemmatization. Construction of a histogram.

4. Prediction of reduced prices for goods. Using linear regression with such data X as price, rating, number of reviews, and Y - reduced price - build a model for predicting reduced prices. Conduct model evaluation (MAE, MSE, RMSE, R2). Construction of graphs with residual function and "Residuals Distribution".

5. Building a personalized recommendation model. The recommend_products function recommends products to the user based on the similarity of descriptions, using the history of his purchases, calculates the similarity between products by calculating the cosine similarity of their descriptions (represented by TF-IDF vectors). The function returns the 5 best products with the highest similarity scores that the user has not yet purchased. If the user has no purchase history, it will return a message about this. The get_tfidf_matrix function calculates and returns the TF-IDF matrix of product descriptions, which is used to measure the similarity between products.

1.2Amazon Sales Dataset

Amazon Sales Dataset, available on Kaggle, contains data of 1K+ ratings and reviews of products on the platform according to their details specified on the official Amazon website, contains detailed information about various products sold through Amazon, including such attributes as product pricing, ratings, number of reviews, sales volumes, and product descriptions that can be used for various analytical purposes, such as sales forecasting, product trend analysis, or customer behavior analysis.

One of the main goals of using this dataset is to gain an understanding of how various factors, such as price, ratings, and product category, affect sales effectiveness. Possible analysis of comments for classification into negative, neutral, and positive. The presence of user and purchase data allows for the formation of a recommendation model based on previous purchases.

Columns in the dataset:

- product_id: a unique identifier for each product in the dataset;
- product_name: the name of the product as presented on Amazon;
- category: category to which the product is classified (e.g., electronics, books, fashion);
- discounted_price: product price after any discounts are applied;
- actual_price: original product price without any discounts;
- discount_percentage: discount percentage applied to the original product price;
- rating: evaluation from 1 to 5 stars;
- rating_count: number of users who have rated the product;
- about_product: brief description of the product, usually provided by the seller;
- user_id: unique identifier of the user who wrote the review about the product;
- user_name: username that wrote the feedback;
- REVIEW_ID: A unique ID for each review sent by the user;
- Review_title: a short headline or inspection resume;

– REVIEW_CONENT: Detailed Content of the Response Written by the User;

– IMG_LINK: URL product image;

– Product_Link: URL of the official product page on Amazon.

1.3 Libraries are involved

Libraries involved together support data processing, text analysis, mood classification and machine learning tasks, such as regression and development of a system of recommendations:

1. re (regular expressions): provides functions for working with regular expressions, allowing for matching text patterns and manipulating them.

2. nltk (Natural Language Toolkit): a library for text processing and natural language processing (NLP) tasks such as tokenization (word_tokenize), stopword removal, lemmatization (WordNetLemmatizer), and sentiment analysis (SentimentIntensityAnalyzer).

3. pandas: a powerful library for data processing and analysis, especially for handling tabular data (DataFrames)

4. numpy: provides support for large multi-dimensional arrays and matrices.

5. Seaborn: Matplotlib -based data library that provides a high -level interface to draw statistical graphics.

6. Matplotlib: A widely used library to create static, animated and interactive visualizations in Python.

7. Sklearn (Scikit-LEARN): A powerful library for machine training, including tools for pre-processing, classification, regression, clustering, etc.:

– LabelnCoder: used to convert categorical labels into numerical values that are required for machine training models;

– Standardscaler: Data scaling tool that standardizes data by deleting the average value and scaling to a single variance;

– Train_test_Split: Utility for separation of data sets into learning and testing sets for model evaluation;

– Linear Regression: implements linear regression to forecast numerical values based on input functions;

– mean_absolute_error, mean_squared_error, R2_score: metrics for evaluating regression models;

– TfidfVectorizer: converts a collection of text documents into a TF-IDF feature matrix, which is used for text-based machine learning tasks;

– cosine_similarity: measures the cosine of the angle between two vectors, which is commonly used to calculate similarity between text documents;

1.4 Used Technologies

1.4.1 Linear Regression

Linear regression is one of the simplest and most widely used methods in statistics and machine learning for modeling the relationship between variables. It allows for predicting the value of one variable (dependent) based on the values of other variables (independent), assuming a linear relationship between them.

The basic idea of linear regression is to build a line that minimizes the sum of squared deviations between actual values and predicted values. This is achieved through the method of least squares (OLS - Ordinary Least Squares).

Linear regression is used for predicting numerical values such as prices, costs, sales, or other indicators that may have a linear relationship with other variables.

However, linear regression also has its limitations, for example, it assumes a linear relationship between variables, as well as requires that variables be independent and not have strong correlation.

1.4.2 Neural Network

Neural Network is a mathematical model that imitates the human brain's work for solving various machine learning tasks, such as classification, regression, image processing, and natural language processing. It consists of a large number of simple computational elements, called neurons, which are connected to each other and work together to analyze input data and provide output results. Typically, it consists of an input, hidden, and output layers.

During training, the network receives data and adjusts the weights (coefficients) of the connections between neurons to minimize error. Methods such as backpropagation and gradient descent are used. The trained model can predict new data using the obtained weights.

Compared to linear regression, it can be deduced that the NN is capable of detecting hidden relationships between data, but requires much more training data to effectively solve regression tasks than linear regression.

1.4.3 Natural Language Processing

Natural Language Processing (NLP) is a field of artificial intelligence (AI) that focuses on enabling machines to understand, interpret, and generate human language in a meaningful and useful way. NLP combines linguistics and informatics to process and analyze large amounts of natural language data. This is a key technology underlying many modern AI applications, such as chatbots, sentiment analysis, and language translation systems. Main components and tasks of NLP:

Preprocessing text: this step involves cleaning and preparing unprocessed text for analysis, including tasks such as tokenization (splitting text into words or sentences), stop-word removal (common but unimportant words), and stemming or lemmatization (shortening words to their root forms).

Mood Analysis: Used to determine the mood or emotional tone behind the text. It is often used for customer reviews, social media monitoring, and brand sentiment analysis.

Named Entity Recognition (NER): This technique identifies and classifies objects in the text into predefined categories, such as names of people, organizations, locations, dates, etc.

Part-of-Speech Tagging (POS): This involves tagging words with their respective parts of speech, such as nouns, verbs, adjectives, etc., to understand the sentence structure.

Text Classification: This includes tasks such as spam detection, topic categorization, and document classification.

In recent years, progress in deep learning and neural networks, such as transformer models (e.g., BERT, GPT), has significantly improved the accuracy and capabilities of NLP systems. These models can better understand context and nuances, leading to more complex language tasks such as answering questions, summarizing, and creating dialogue.

1.4.4 TfidfVectorizer

The TF-IDF (Term Frequency-Inverse Document Frequency) vectorizer is a technique used in text processing to convert text data into a numeric representation that reflects the importance of words in a document relative to the corpus of documents. It combines two concepts:

Term Frequency (TF): the number of times a term (word) appears in a document. This measures the visibility of the term in a specific document.

Inverse Document Frequency (IDF): a metric indicating how important a term is in the entire corpus of documents. It assigns higher values to terms that occur in fewer documents, thus highlighting terms that are more characteristic.

The combination of these two metrics gives a weight for each word in the document, ensuring better representation of the content. The TF-IDF Vectorizer in scikit-learn is a tool that computes these values and transforms text into vectors (a numerical form), which can then be used for various machine learning tasks, such as text classification, clustering, or similarity analysis.

1.4.5Cosine Similarity

Cosine similarity is a measure used to calculate the similarity between two non-zero vectors in the inner product space. It is used for measuring the similarity between text documents since it compares the angle between two vectors rather than their magnitude, making it robust to differences in the length of documents.

$$CS = \frac{A \cdot B}{\|A\| * ||B||}\text{too,}$$

- A and B are vectors (for example, TF-IDF vectors).
- *denotes the scalar product of two vectors.
- $\|A\|$ і $\|B\|$– величини (евклідова норма) векторів.

Cosine similarity yields a value from -1 to 1:

- 1 indicates identical vectors;
- 0 means orthogonality (no similarity);
- -1 indicates completely dissimilar vectors.

Practically, cosine similarity is often used to calculate similarity between documents, for example, for recommendation systems or information search.

2 IMPLEMENTATION OF THE TASK

2.1 Preprocessing

Any task solved in the field of data intelligence analysis begins with cleaning the dataset data, checking for existing columns and data types (Fig. 2.1).

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1465 entries, 0 to 1464
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   product_id           1465 non-null   object
 1   product_name         1465 non-null   object
 2   category             1465 non-null   object
 3   discounted_price     1465 non-null   object
 4   actual_price         1465 non-null   object
 5   discount_percentage  1465 non-null   object
 6   rating               1465 non-null   object
 7   rating_count         1463 non-null   object
 8   about_product        1465 non-null   object
 9   user_id              1465 non-null   object
 10  user_name            1465 non-null   object
 11  review_id            1465 non-null   object
 12  review_title         1465 non-null   object
 13  review_content       1465 non-null   object
 14  img_link             1465 non-null   object
 15  product_link         1465 non-null   object
dtypes: object(16)
```

Figure 2.1 The contents of the dataset

The following will be removed "" and coma from Discounted_price and Actual_price columns to obtain numerical values without textual characters, and convert this data into a numerical format (Float64). Similarly, you are cleansing the discount_Percentage column, removing the characters "%" and turning it into particles, dividing the value by 100.

Processing of rating columns. Initially, all unique values in the Rating column are checked, and then converted into numerical format (Float64), setting incorrect values (such as text characters) in NAN. You then remove all the lines where the Rating value is empty (NAN).

For the Rating_count column, comma is deleted to get net numerical values and again turn into Float64 type. Lines with empty values are also deleted.

The daset has a column with categories. The first category (Main_CateGory) provides a general overview

Finally, checking whether dataset has duplicates to make sure all the records are unique. No duplicate was detected.

Summary of numerical columns in the dasette (Fig. 2.2).

	discounted_price	actual_price	discount_percentage	rating	rating_count
count	1462.000000	1462.000000	1462.000000	1462.000000	1462.000000
mean	3129.981826	5453.087743	0.476724	4.096717	18307.376881
std	6950.548042	10884.467444	0.216139	0.289497	42766.096572
min	39.000000	39.000000	0.000000	2.000000	2.000000
25%	325.000000	800.000000	0.320000	4.000000	1191.500000
50%	799.000000	1670.000000	0.500000	4.100000	5179.000000
75%	1999.000000	4321.250000	0.630000	4.300000	17342.250000
max	77990.000000	139900.000000	0.940000	5.000000	426973.000000

Figure 2.2 Datesset statistics

2.2 Data visualization

Distribution of products by basic categories (Fig. 2.3). The most popular categories are electronics, computers and accessories and homes and low demand cuisine, such as office goods, musical instruments, home improvements, toys and games, cars and motorcycles, health and personal hygiene.

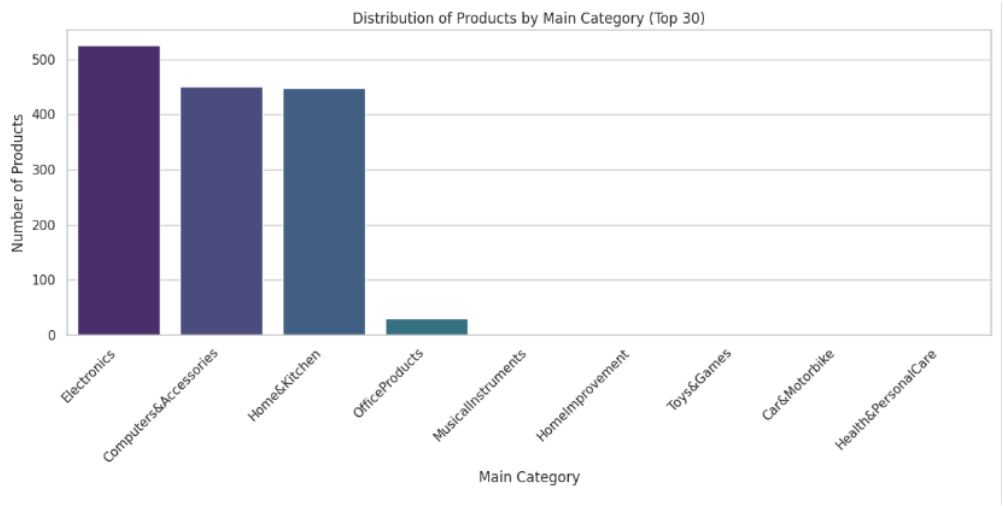

Figure 2.3 Distribution of products by basic categories

The distribution of products into the latest categories (Fig. 2.4) The most popular categories are USB cables, smart watches, smartphones, smart TVs, headphones and remote controls.

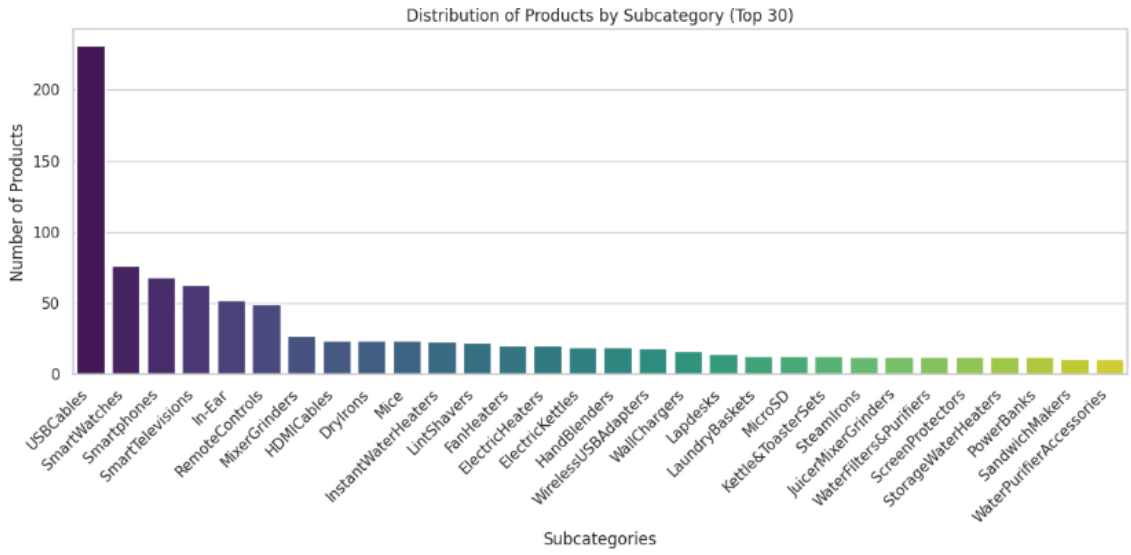

Figure 2.4 - Distribution of Products by Latest Categories

Price Distribution (fig. 2.6) If the histogram is shifted to the right (most products have a lower price, while some are more expensive), this means that most products are affordable, but there may be a few premium products.

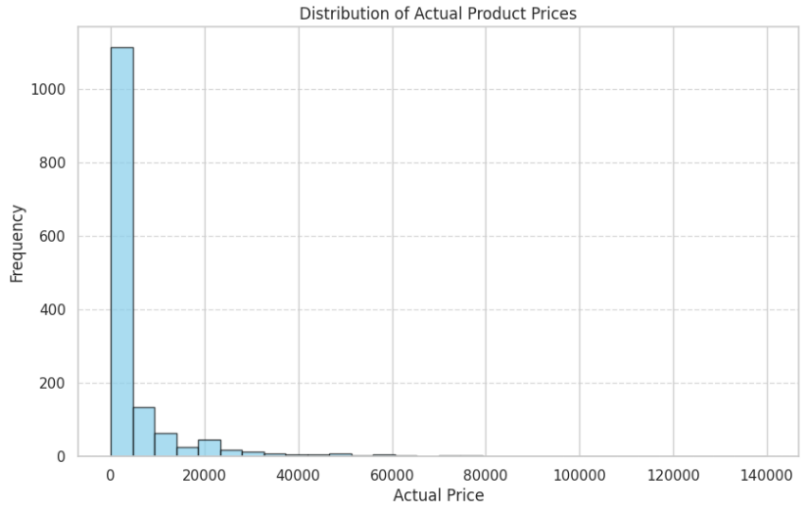

Figure 2.6 – Price Distribution

Rating Distribution (fig. 2.7). Most ratings cluster around higher values (for example, 4 and 4.5 stars), indicating that customers are generally satisfied with the products.

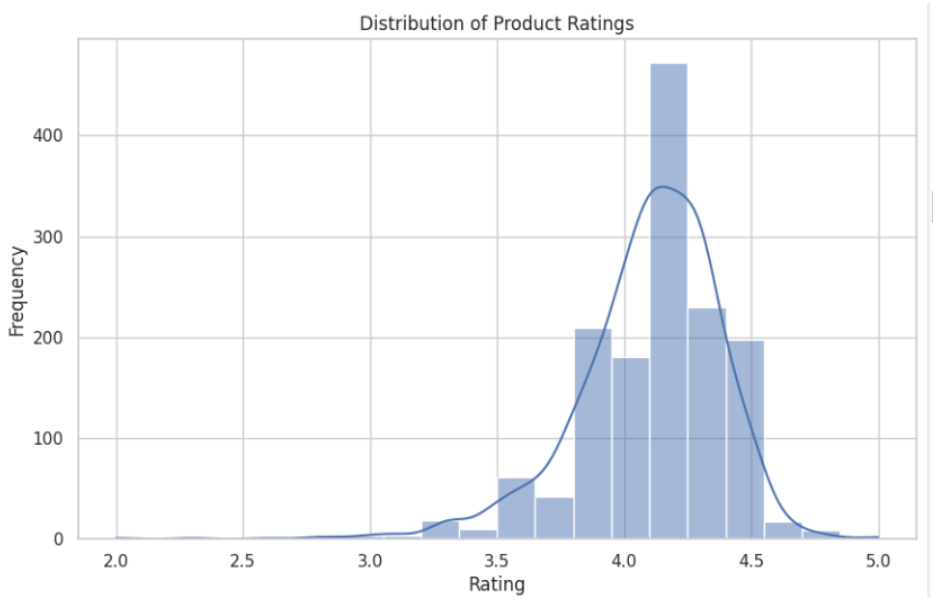

Figure 2.7 – Rating Distribution

Rating Analysis (fig. 2.8). Do longer reviews usually correspond to higher or lower ratings? Average.

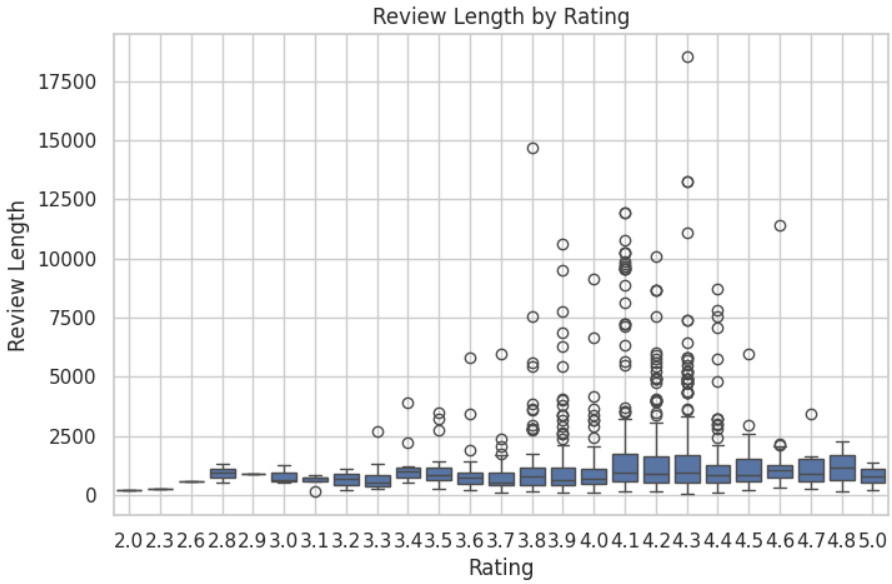

Figure 2.8 – Rating Analysis

2.3 Review Analysis

Proposed software implementation of text review analysis for users, specifically their cleaning, lemmatization, and sentiment evaluation, using natural language processing (NLP) methods. Steps taken:

1. Text preprocessing. First, the code identifies a set of English stop words and creates a lemmatizer (a technique used in natural language processing (NLP) models to break down a word to its root value to determine similarity) based on the WordNetLemmatizer library. Transformation to lowercase, removal of URLs, removal of non-alphanumeric symbols. Tokenization, lemmatization, removal of stop words are performed. The cleaned text is saved in a new column cleaned_review .

2. Tone Analysis using VADER. The SentimentIntensityAnalyzer library is used to determine the emotional rating of cleaned reviews. The function get_sentiment_vader calculates the compound tone index, which can range from -1 (very negative) to +1 (very positive). Results are stored in the sentiment_score column.

3. Tone Classification. The classify_sentiment_vader function classifies the review rating based on the compound index:

– positive (Positive) if the index > 0;

– negative (Negative) if the index < 0;

– Neutral (Neutral), if the indicator is equal to 0.

4. Visualization of tone distribution. Finally, the code calculates the number of reviews for each category (Positive, Negative, Neutral) and builds a column chart for visualizing the results (fig. 2.9). Verification of the correctness of the results (fig. 2.10). Items with negative reviews received sentiment_score -0.56 and -0.74, indicating the correctness of the program code.

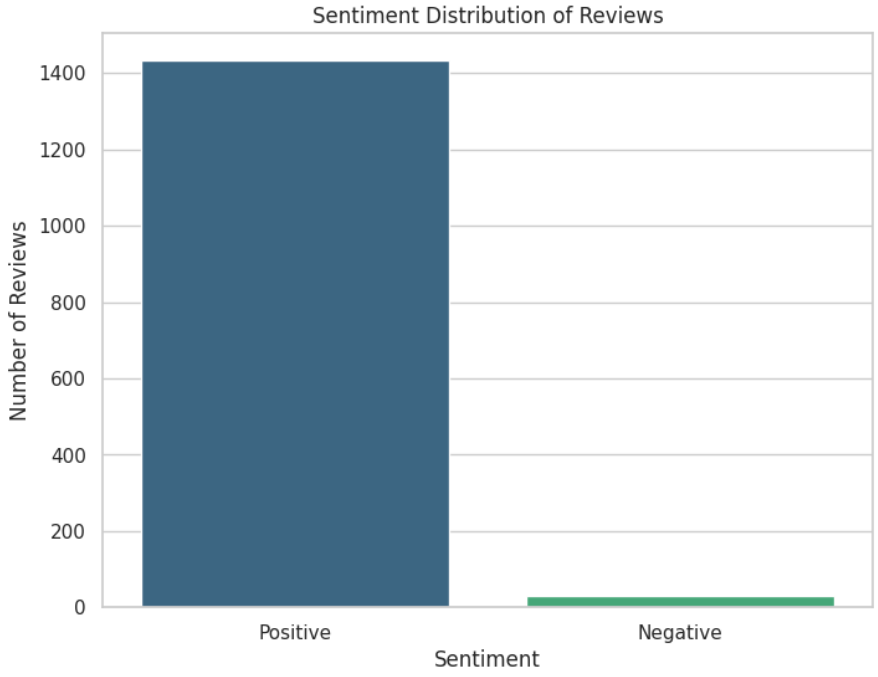

Figure 2.9 – Review classification

152	Samsung 80 cm (32 inches) Wondertainment Serie...	Good,Sound is very low another brand comparing...	-0.5574	Negative
155	7SEVEN® Compatible for Tata Sky Remote Origina...	do not buy	-0.7425	Negative

Figure 2.10 – Verification of the correctness of the results

2.4 Discount proces forecast

2.4.1 Discount proces forecast – Linear Regression

An analysis of data is performed for forecasting the reduced price of a product using machine learning methods, specifically using Linear Regression.

After data cleaning, features and the target variable are selected that will be used to build the model. The features (X) are: actual price of the product, rating, number of reviews, and the target variable (y) is the reduced price of the product.

For improving the model quality, data normalization is performed using StandardScaler, which scales the features so that the mean equals 0 and the standard deviation is 1. This improves the efficiency of the machine learning algorithm, especially when the features have different value ranges.

Next, the data is split into training and test samples in a ratio of 80% to 20%. This allows the model to learn from the main part of the data and then evaluate its accuracy on new, previously unseen data. After preparing the data, a linear regression model is created and trained on the training sample.

After the model is trained, predictions are made on the test sample. The following metrics are used to evaluate the model's quality (fig. 2.11):

– Mean Absolute Error (MAE) — the average absolute deviation of the predicted values from the actual;

– Mean Squared Error (MSE) — the average squared difference, emphasizing larger errors;

– Root Mean Squared Error (RMSE) — the square root of MSE, indicating the average error in the same units as the target variable;

– R-squared (R^2) — the coefficient of determination, showing what proportion of the variability of the target variable the model explains. A value close to 1 suggests that the model performs well in its task.

```
Mean Absolute Error (MAE): 684.12
Mean Squared Error (MSE): 1913777.78
Root Mean Squared Error (RMSE): 1383.39
R-squared (R²): 0.93
```

Figure 2.11 – Model Accuracy Assessment

Create a scatter plot (Fig. 2.12) showing the comparison between actual (y_test) and predicted (y_pred) values of the discounted price of the product. This allows for assessing how accurate the model is in its forecasts.

Figure 2.12 – Actual and predicted prices with discount and regression line

Create a histogram of residuals (Fig. 2.13) (the difference between actual and predicted values), which helps to determine how well the model performs its task. We observe a normal distribution, indicating that the model is well-suited for forecasting.

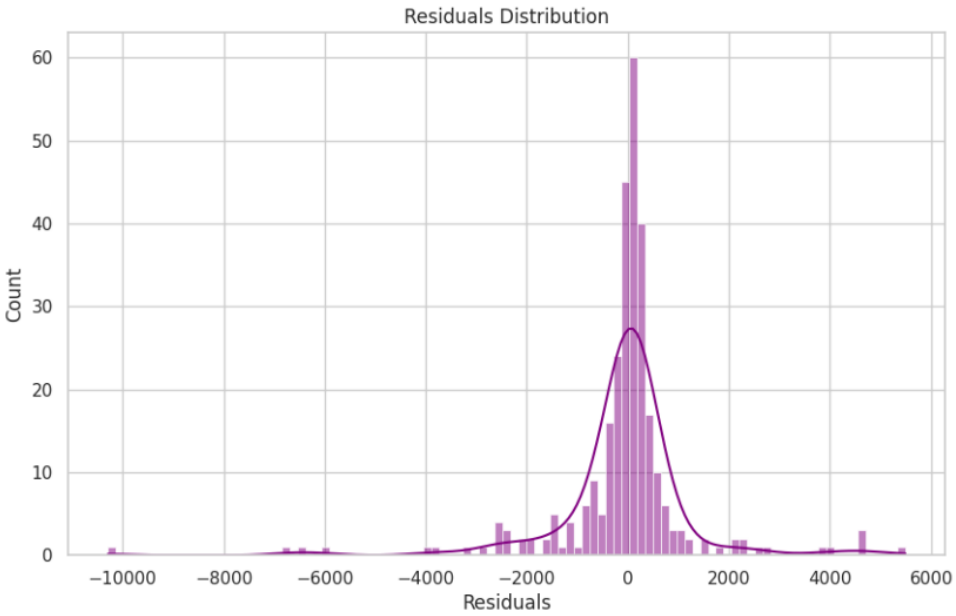

Figure 2.13 – Histogram of residuals

2.4.2 Discount prices forecast – Neural Network

The same task is realized with the help of NM. First, the required columns from the data set: Actual_price, Rating and Rating_count to use them as incoming variables (factors) for the model. The target variable, that is, what we want to predict is Discounted_price.

Standardscaler is used at the data preparation stage to normalize input. Further, the data is divided into a training set (80%) and a test set (20%) to check how well the model works on unknown data.

The architecture of the neural network consists of three hidden layers containing 64, 32 and 16 neurons, respectively, each uses the function of Leaky Relu activation, helps to avoid "fading gradient" problems, which can improve the model's ability to learn. The last layer is the original and has only one neuron, which returns the value of the reduced price.

Model is compiled using an optimizer. Mean squared error (MSE) is used as the loss function, which evaluates the mean squared error between actual and predicted values, as well as mean absolute error (MAE) is calculated for additional analysis.

The model is trained for 100 epochs with a batch size of 32. During training, 20% of the training set is allocated for validation to monitor the quality of training and prevent overfitting. After completing the training process, the model is tested (fig. 2.14) on the delayed test set, where its performance is evaluated using MSE and MAE metrics.

	Actual	Predicted
894	1199.0	1146.860229
1108	1099.0	953.694580
415	599.0	1074.925049
524	99.0	239.272308
1038	1290.0	1086.210693

Figure 2.14 – Data comparison

Metrics are calculated (fig. 2.15). RMSE (root mean squared error) to get an idea of the mean error in the same units as the original data. Also, the coefficient of determination R^2 is calculated, which shows how well the model explains the variation in the target variable: an R^2 value close to 1 indicates a high accuracy of the model.

```
Test Loss (MSE): 1870265.50
Test Mean Absolute Error (MAE): 653.66
Root Mean Squared Error (RMSE): 1367.58
R² Score: 0.94
```

Figure 2.15 – Metrics of the Neural Model

The histogram of residuals also shows a normal distribution, indicating sufficient accuracy of the model.

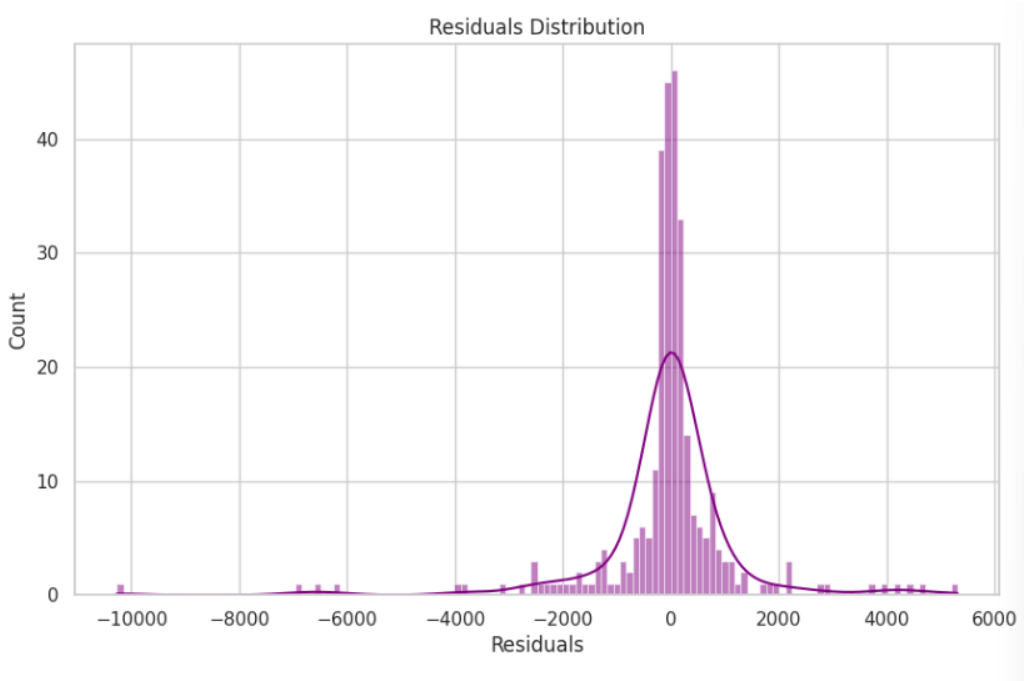

Figure 2.16 – Histogram of Residuals

Summary on the comparison of Linear Regression and Neural Network for solving the same task: both models showed comparable results. Linear regression is a better choice for simple, linear tasks with lower computational requirements, whereas neural networks are suitable for more complex, non-linear problems with large datasets.

2.5 Recommendation System

Implementation of a system of recommendations based on the similarity of content using TF-IDF for descriptions of products and cosine similarity to compare these descriptions. Steps:

1. Magazine coding for users: Labelncoder () Sklearn.preprocessing is used to convert user IDs into a numeric format (user_id_encoded) to facilitate the calculation in the model. This coding is stored in the new User_id_encoded column.

2. Frequency Table: Code generates a frequency table for encoded users by providing information on how often each user appears in a data set (Fig. 2.17), although this is first and foremost for analysis, not for the logic of recommendations.

	User ID	Frequency
0	1045	10
1	622	8
2	673	8
3	253	7
4	87	7

Figure 2.17 – Table of purchase frequencies of users

3. TF-IDF matrix: TfidfVectorizer is used to convert product descriptions (column about_product) into numeric vectors. These vectors reflect the importance of each word in the context of the entire product description corpus, with common words having less weight. Missing values in the about_product column are processed by replacing them with empty lines to prevent errors during transformation.

4. Calculation of cosine similarity: using cosine_similarity from sklearn.metrics.pairwise, the code calculates similarity metrics between products. This is done by comparing the TF-IDF vectors of the products with which the user interacted (bought) with the entire product catalog.

5. Recommended products: for a specific user, the recommendation system filters products with which the user has already interacted and then ranks other products based on their cosine similarity to the user's previous purchases. Then it returns the 5 most similar products as recommendations, excluding already purchased items.

6. Exit: Results are stored in DataFrame that includes a user -coded identifier, recommended products and evaluations of their similarity.

The recommend_products feature is designed to provide product recommendations based on their previous interactions using content -based filtration

approach. This method compares the content of the products that the user interacted with, such as descriptions, and recommends such products.

Input parameters

– DF: DataFrame containing product information, including product descriptions and user actions;

– user_id_encoded: numeric identifier of the user used to filter products with which the user has previously interacted;

– tfidf_matrix: an optional precomputed matrix of TF-IDF values representing the product description. If not provided, the function computes it internally.

Handling missing data: the function checks if the TF-IDF matrix has already been provided. If not, it computes the matrix using the TfidfVectorizer from the scikit-learn library, which converts product descriptions into numeric vectors based on the document frequency-inverse document frequency (TF-IDF). This step helps represent textual data in a way that a machine learning model can understand.

The function then filters the dataset to obtain products with which the specified user has interacted based on the user_id_encoded parameter. If no purchase history is found for the user, the function prints a message and returns None.

The essence of the recommendation system lies in calculating the cosine similarity between the user's products with past interactions and all other products in the catalog. Cosine similarity measures how close two vectors are in terms of their angle, which in this case means how similar the descriptions of two products are. The higher the cosine similarity score, the more similar the products. This function uses the cosine_similarity function from scikit-learn to calculate similarity ratings.

After calculating similarity scores, the function repeats all products except those with which the user has already interacted. It sorts the products by similarity scores in descending order, ensuring that the most similar products have priority. From this sorted list, the function selects the 5 best recommended products.

The feature returns Dataframe containing the following columns:

– user_id_encoded: User ID;

– Recommended product: name of the recommended product;

– Ball: The cosine similarity score indicates how much the product corresponds to the user's past.

Function call will return recommend_products(df, 253, matrix) which will return a list of products most recommended for purchase for the user number 253 (fig. 2.18).

	user_id_encoded	recommended_product	score
0	253	boAt Deuce USB 300 2 in 1 Type-C & Micro USB S...	1.0
1	253	boAt Deuce USB 300 2 in 1 Type-C & Micro USB S...	1.0
2	253	boAt Rugged v3 Extra Tough Unbreakable Braided...	1.0
3	253	boAt Deuce USB 300 2 in 1 Type-C & Micro USB S...	1.0
4	253	boAt Deuce USB 300 2 in 1 Type-C & Micro USB S...	1.0

Figure 2.18 – Recommended products for purchase

Check the purchase history of the same user (fig. 2.19).

	user_id_encoded	product_name	sub_category
3	253	boAt Deuce USB 300 2 in 1 Type-C & Micro USB S...	USBCables
11	253	boAt Rugged v3 Extra Tough Unbreakable Braided...	USBCables
92	253	boAt Deuce USB 300 2 in 1 Type-C & Micro USB S...	USBCables
258	253	boAt Rugged V3 Braided Micro USB Cable (Pearl ...	USBCables
392	253	boAt Deuce USB 300 2 in 1 Type-C & Micro USB S...	USBCables
443	253	boAt Rugged v3 Extra Tough Unbreakable Braided...	USBCables
628	253	boAt Deuce USB 300 2 in 1 Type-C & Micro USB S...	USBCables

Figure 2.19 – Check the purchase history of the same user

CONCLUSION

The work covers the most important methods in machine learning, from pre - processing of data to evaluation of forecast models and the generation of personalized recommendations on products. The analysis of moods makes it clear to customer feedback, and regression models and neural networks provide valuable forecasts for products for products. The recommendation system guarantees that users will receive individual proposals based on the similarity of the content, using advanced methods such as TF-IDF and cosine similarity. Common methodologies are the basis of data processing work processes for e-commerce platforms, where understanding user behavior, creating accurate forecasts and providing personalized recommendations are crucial for improving client interaction and achieving business results.