BAN 620 Data Mining Project Report

Bestselling Product Prediction using KNN & Classification Trees

Submitted by :

Group-6

Sarath Kumar Vatyam- ss2405

Abhi Ram Sai - bd2500

Neeharika Vijjada – as1195

INTRODUCTION:

In the dynamic landscape of e-commerce, predicting bestselling products has become important for businesses seeking a competitive edge. In this project titled "Best Selling product prediction using KNN & Classification trees" we have developed machine learning models for identifying products likely to be best sellers. Using K-nearest neighbors (KNN) and Classification trees. Our objective is to identify patterns in the dataset and explore the features like pricing, rating, reviews, units sold etc.

Identifying whether a product will become a bestseller is crucial for businesses to optimize their inventory, marketing, and sales strategies. Through this project we aim to provide sellers/businesses with insights about predictions to optimize inventory management and enhance marketing strategies. This ultimately helps them in succeeding in this ever-evolving market.

DATA COLLECTION:

Our first priority in this project is to establish a reliable foundation for data gathering. To achieve this, we have evaluated and selected a specific data source that aligns with our project's objectives and requirements. The chosen data source for our project on product price prediction is from KAGGLE.

Link to the data source: https://www.kaggle.com/datasets/asaniczka/usa-optimal-product-price-prediction

TABLE OF CONTENTS of DATASET:

S.No	columns	Description
1	**uid**	Unique identifier for each record or product in dataset
2	**asin**	Unique standard identification number assigned to products on Amazon
3	**title**	Titles or names of products
4	**stars**	Star ratings given to products
5	**reviews**	Number of reviews each product received
6	**price**	Price of products
7	**category**	Category or type of products
8	**isBestSeller**	Binary values (0,1) indicating whether a product is bestseller
9	**boughtInLastMonth**	Products sold in last month

Limitations: Although the chosen data set has less columns (i.e., 9), it has huge number of records to analyze close to (~1.7 million)

DATA PREPROCESSING:

Data preprocessing is a crucial step in data analysis. It involves cleaning and transforming raw data into format suitable for analysis and modeling. Here are some key things that we have done as a part of data preprocessing:

- We ensured that there are no missing values or duplicates in the columns. We have removed uid, aisin, title columns as they are product lookup references ,however not useful for analysis part. We have created dummies for categorical variables in -the dataset for 'Category' and 'iBestSeller' Columns. Checked datatypes of all the columns. We have removed the columns for which prices are 0- Approximately ~37k Records.

```
 #   Column            Non-Null Count    Dtype
---  ------            --------------    -----
 0   uid               1443512 non-null  int64
 1   asin              1443512 non-null  object
 2   title             1443512 non-null  object
 3   stars             1443512 non-null  float64
 4   reviews           1443512 non-null  float64
 5   price             1443512 non-null  float64
 6   category          1443512 non-null  int64
 7   isBestSeller      1443512 non-null  int64
 8   boughtInLastMonth 1443512 non-null  float64
dtypes: float64(4), int64(3), object(2)
```

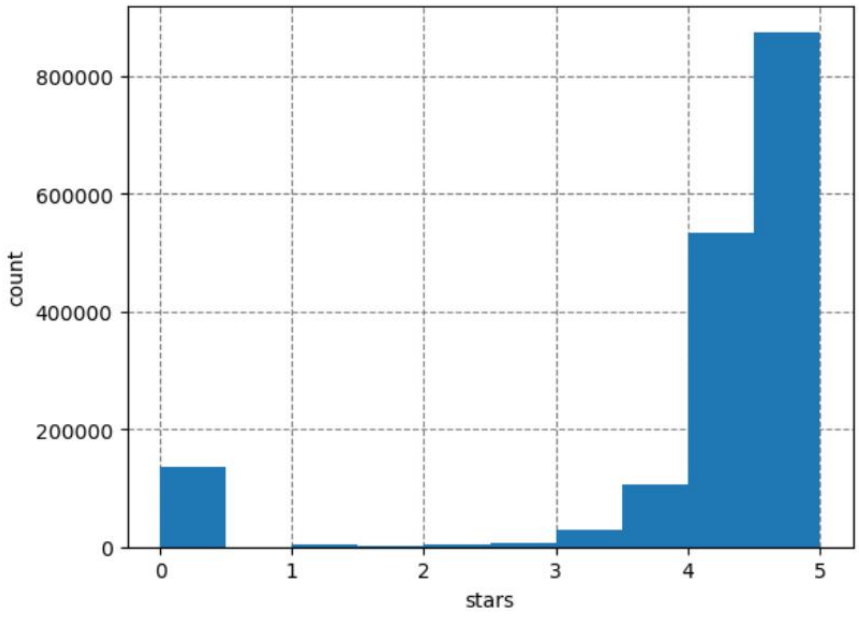
Analyzing the bar chart reveals a significantly higher number of isBestSeller=False records when compared to True records.

Categorized Variables: Identified and categorized variables into numerical and non-numerical types. Numerical variables include integers and floats, while non-numerical variables are represented as strings or objects.

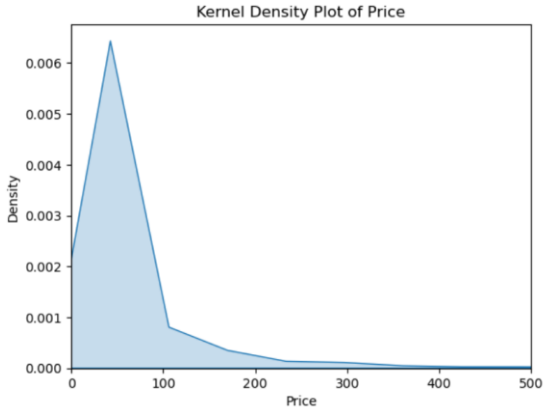
- Label Encoding: Coded category names to numerical values for efficient analysis. For instance, the category "Accessories & Supplies" is transformed into numerical value of 1

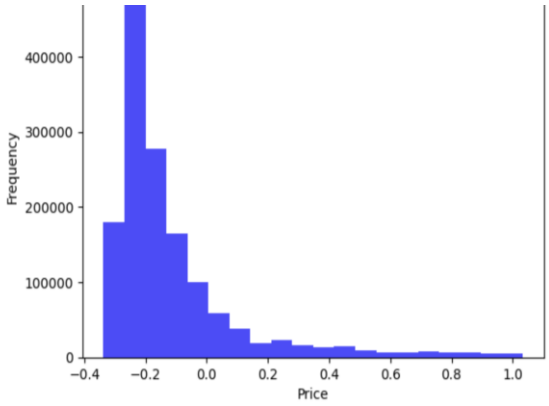
```
from sklearn.preprocessing import LabelEncoder

# Assuming 'X' is your DataFrame and 'category' is the column you label encoded
label_encoder = LabelEncoder()
balanced_data_shuffled['category'] = label_encoder.fit_transform(balanced_data_shuffled['category'])
```

Another observation is that most of the products have 5-star ratings and there are some items with 0-star rating that have high sales.



We have plotted a kernel density function for pricing to see the pricing demography. It is noticed that major products fall under the pricing category of < 100$.We have also observed some outliers that are priced above 15000$ and we have removed those as a part of data cleaning. Also, the frequency of products specific to price ranges is also plotted.





METHODOLOGY:

KNN MODEL:

K-Nearest Neighbors (KNN) is a straightforward and versatile algorithm for classification and regression tasks. The methodology involves understanding the problem, exploring dataset, preprocessing data to handle missing values, outliers and convert features into suitable format. Critical step is choosing the value of K, determining the number of neighbors considered during prediction. Model performance is evaluated using appropriate metrics and iteration. The KNN model can then be deployed for predictions on new data if it meets the desired performance criteria.

CLASSIFICATION TREES:

Classification trees are a popular machine learning algorithm, follows a systematic methodology for solving classification problems. The tree structure is built during training phase with nodes representing decision points based on specific features and splitting criteria. The algorithm recursively determines the optimal feature to split on each node, constructing a hierarchical set of decision rules. The resulting tree is pruned to avoid over fitting, ensuring generalization to new data. The model is then evaluated using metrics such as accuracy or Gini impurity. Adjustments are made to parameters for optimization.

KNN-MODEL TRAINING AND EVALUATION:

After pre-processing and data exploration, we have split the data into train data and validation data to run the model. We have normalized the predictor columns and used the z-score values for analysis and prediction. We have trained the model accordingly on the train dataset.

	k	accuracy
0	1	0.789082
1	2	0.800972
2	3	0.813172
3	4	0.819996
4	5	0.827337
5	6	0.828060
6	7	0.829094
7	8	0.831266
8	9	0.832610
9	10	0.834781
10	11	0.832920
11	12	0.834057
12	13	0.835918
13	14	0.834677
14	15	0.835401
15	16	0.834057
16	17	0.834160
17	18	0.834574
18	19	0.833850

```
[61]: #At k=13 we have 83.59 accuarcy
```

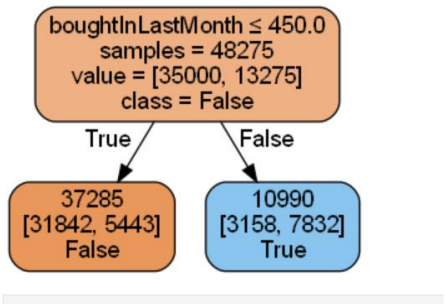
We have run the model on train dataset and figured the highest accuracy of K on training data set at K = 13 I.e., 83.59% of accuracy. Hence, we have decided to use K= 13 and run the model on that validation dataset. After running K= 13, we have got 83.36% of accuracy in prediction with respect to validation dataset**.**

```
Confusion Matrix (Accuracy 0.8336)

       Prediction
Actual    0    1
     0 6514  525
     1 1082 1534
```

CLASSIFICATION TREE MODEL TRAINING AND EVALUATION:

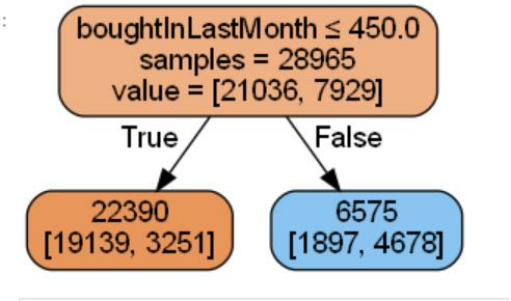
A classification model was developed using decision tree classifiers and was fed with the same data as KNN**.** Trees with depths=1,3 have been developed to check the accuracy on validation data.

[93]:



Depth=1

Accuracy on validation data=82.18



Depth=3

Accuracy on validation data=82.77

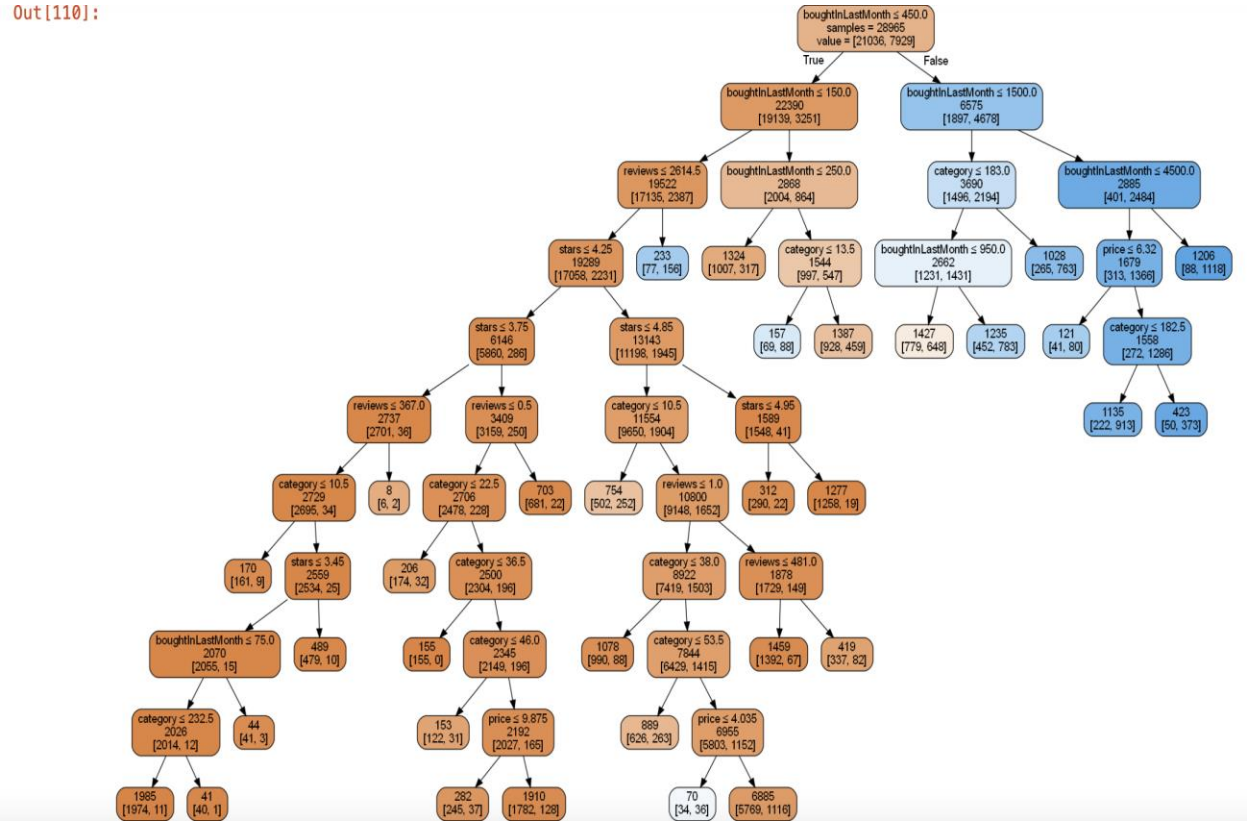
Later using grid search the parameters for the best tree was identified and used in model. Accuracy from validation data is 82.7% and this model was improved by random forest method which then gave an accuracy of 83.3%

```
classificationSummary(valid_y, rf.predict(valid_X))
```

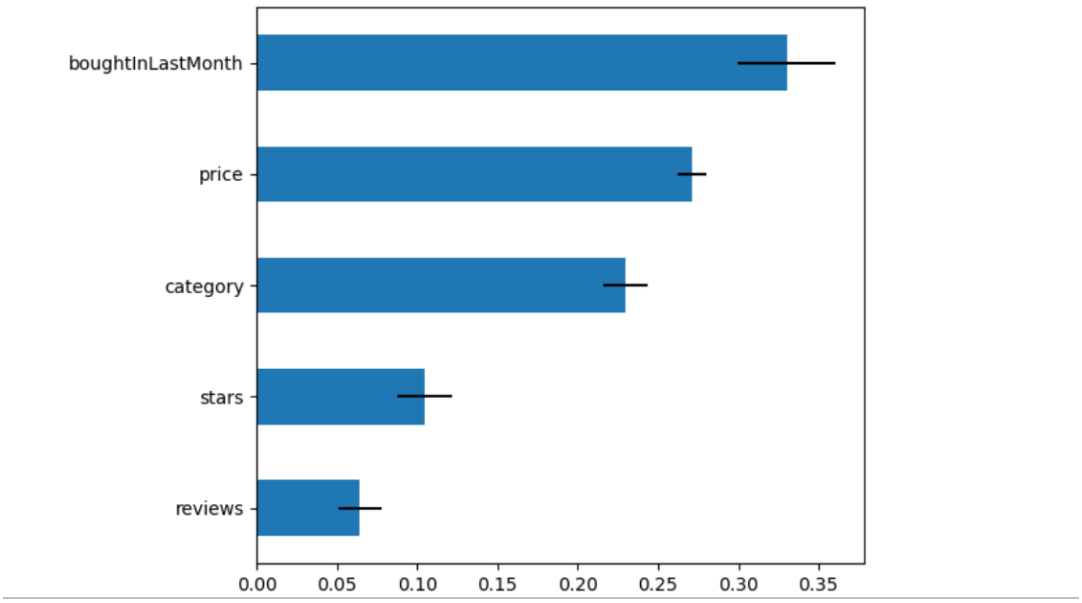
```
Confusion Matrix (Accuracy 0.8338)

       Prediction
Actual     0     1
     0 12635  1329
     1  1880  3466
```

We have set a depth of 10 and split count=1500 and obtained the following decision tree. We can derive multiple rules from this tree. For Example, If boughtinlastmonth >4500, then surely that product 'isBestSeller' Product as a rule.

Out[110]:



Additionally, identified and utilized key features to construct a parsimonious model. Important features are **'boughtInLastMonth', price', 'category', 'stars', 'reviews.**



Used **grid search method** to enhance accuracy even further.

```
param_grid = {
    'max_depth': [10, 20, 30, 40],
    'min_samples_split': [20, 40, 60, 80, 100],
    'min_impurity_decrease': [0, 0.0005, 0.001, 0.005, 0.01],
}
gridSearch = GridSearchCV(DecisionTreeClassifier(), param_grid, cv=5, n_jobs=-1) #n_jobs=-1 means
        #that the availalbe computer memory (CPU) will be used to make calculations faster.
gridSearch.fit(train_X, train_y)
print('Initial score: ', gridSearch.best_score_)
print('Initial parameters: ', gridSearch.best_params_)
```

```
Initial score:  0.840704298291041
```

RANDOM FOREST TREES:

Random forest is a powerful machine learning method that boosts the accuracy of predictions by using a combination of multiple classifiers or prediction algorithms. It takes advantage of the wisdom of the crowd by aggregating the outputs of several decision trees, each built using a different subset of the data and features. Although the results of a Random Forest cannot be displayed as a single tree-like diagram, the collective predictions of the individual trees provide highly accurate and reliable predictions for classification or regression problems.

Achieved a validation accuracy of 83.35%

```
rf2 = RandomForestClassifier(n_estimators=1000, random_state=1)
rf2.fit(train_X, train_y)
classificationSummary(valid_y, rf2.predict(valid_X))
```

```
Confusion Matrix (Accuracy 0.8335)

       Prediction
Actual     0     1
     0 12626  1338
     1  1877  3469
```

COMPARISION AND ANALYSIS:

Technically, Classification tree (83.38%) has more accuracy compared with KNN (83.36). However, Classification tree is more complex though we have achieved utmost accuracy which is hard for interpretation. As we see, both the models are similarly accurate, hence we can choose KNN being a simplest model comparatively. The choice between KNN and decision trees should be based on factors

such as the nature of the data, the interpretability requirements, and the computational resources available. We are trading off 0.02% of accuracy by choosing simpler model KNN for ’isBestseller’ prediction.

CONCLUSION:

If the boughtinlastmonth is greater than 4500, then there would be highest probability of being a best seller. Using these prediction models, E-commerce websites can gain a competitive advantage that they can quickly respond to market trends and consumer preferences. Efficient Inventory management and reducing risk of stock-outs can be achieved by predicting using these models. Using these bestselling products, we can make association rules with complementary products and promote them with ease.