**PREDICTION OF SALES: CLASSIFICATION WITH RANDOM FOREST**

**Table of Contents**

[OBJECTIVE](#_7wfdumti1s1z) 3

[METHODOLOGY](#_r8m8nyebm8jt) 3

[END-TO-END PROCESS WITH SOLUTION ARCHITECTURE](#_hy6vwk9lvmbn) 4

[Importing libraries in Python](#_y0w68lvayzyq) 4

[Data exploration](#_3umsl8fq7qow) 5

[Data preprocessing](#_4mv3k18evjsf) 7

[Exploratory data analysis (EDA)](#_s9ymlsncz87y) 9

[Model development](#_7h8nqw7ki6br) 13

[CHALLENGES FACED](#_ny9bcg2144xr) 18

[COMPLEXITY LEVEL](#_e2tazd4o1u0r) 19

# OBJECTIVE

***Problem statement***: A cloth manufacturing company is interested to know about the segment or attributes that cause high sales.

***Objective***: The main purpose of the project is to predict sales categories (High, Moderate or Low) based on attributes like Competitor Price, Income, Advertising, Population, Price, Shelf Location at stores, Age, Education, Urban and U.S., using Random Forest Classifier.

# METHODOLOGY

The description of the dataset has been presented in the below table.

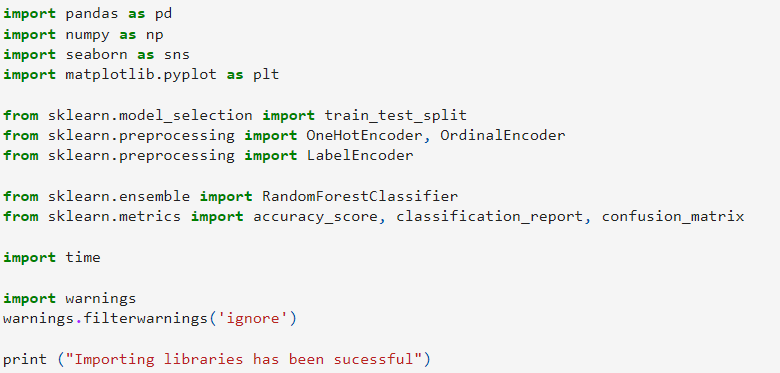
|  |  |
| --- | --- |
| **Variables** | **Description** |
| Sales | Unit sales (in thousands) at each location |
| Competitor Price | Price charged by competitor at each location |
| Income | Community income level (in thousands of dollars) |
| Advertising | Local advertising budget for company at each location (in thousands of dollars) |
| Population | Population size in the region (in thousands) |
| Price | Price company charges for car seats at each site |
| Shelf Location at stores | A factor with levels Bad, Good and Medium indicating the quality of the shelving location for the car seats at each site |
| Age | Average age of the local population |
| Education | Education level at each location |
| Urban | A factor with levels No and Yes to indicate whether the store is in an urban or rural location |
| US | A factor with levels No and Yes to indicate whether the store is in the US or not |

**Table 1: Description of variables**

Target variable in this project is Sales (which was initially a numerical variable), which has been converted into a categorical feature (where sales from 0 to 5 indicate Low, 5 to 10 indicates Moderate and More than 10 indicate High). Thus, due to the categorical nature of the target variable, classification algorithm has been utilised in this project. Random Forest Classification model has been opted for this study, which has been developed, trained and evaluated using Python programming language in Jupyter Notebook IDE.

# END-TO-END PROCESS WITH SOLUTION ARCHITECTURE

## Importing libraries in Python

****

**Figure 1: Importing Libraries in Python**

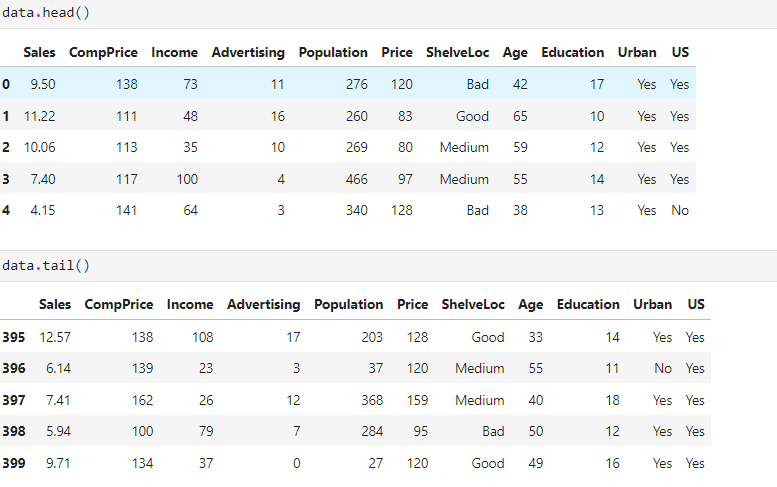
Pandas library has been imported into Python for data loading and manipulation, whereas Seaborn and Matplotlib libraries have been used for data visualisations. In fact, for performing mathematical operations and data aggregation, Numpy library has been used in this project. For developing the random Forest Classification model, the ‘RandomForestClassifier ()’ module has been imported from the ‘sklearn.ensemble’ library. Additionally, for performing data transformation, ‘LabelEncoder’ module has been utilised for label encoding of target variable, whereas, OrdinalEncoder has been imported for encoding ordinal features and OneHotEncoder has been imported for encoding nominal independent variables. All three mentioned modules have been imported from sklearn.preprocessing library from ‘Scikit-learn’ framework. Additionally, essential modules for splitting the dataset (train\_test\_split) and evaluating the model using metrics such as the confusion matrix, accuracy score, and classification report have also been imported from the sklearn.metrics module.

## Data exploration

****

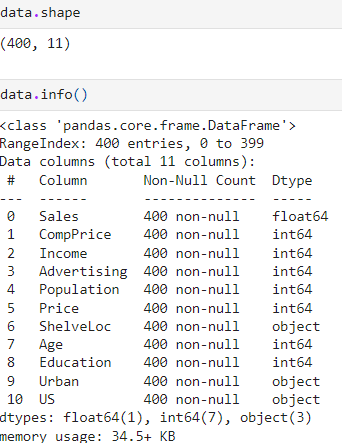
**Figure 2: Data loading**

The dataset has been loaded into Python environment by using ‘read\_csv ()’ function from Pandas library.

****

**Figure 3: Head and tail of the dataset**

Head and tail of the dataset have been shown to check whether the dataset is appropriately loaded or not into the Python environment (**Refer to Figure 3**).

****

**Figure 4: Shape and info of the dataset**

The shape of the dataset is (400,11), which indicates that the dataset contains 400 observations and 11 variables (**Refer to Figure 4**). From **Figure 4**, it can be observed that the dataset contains 3 categorical variables (Shelveloc, Urban and US), whereas 8 variables are categorical.

## Data preprocessing

****

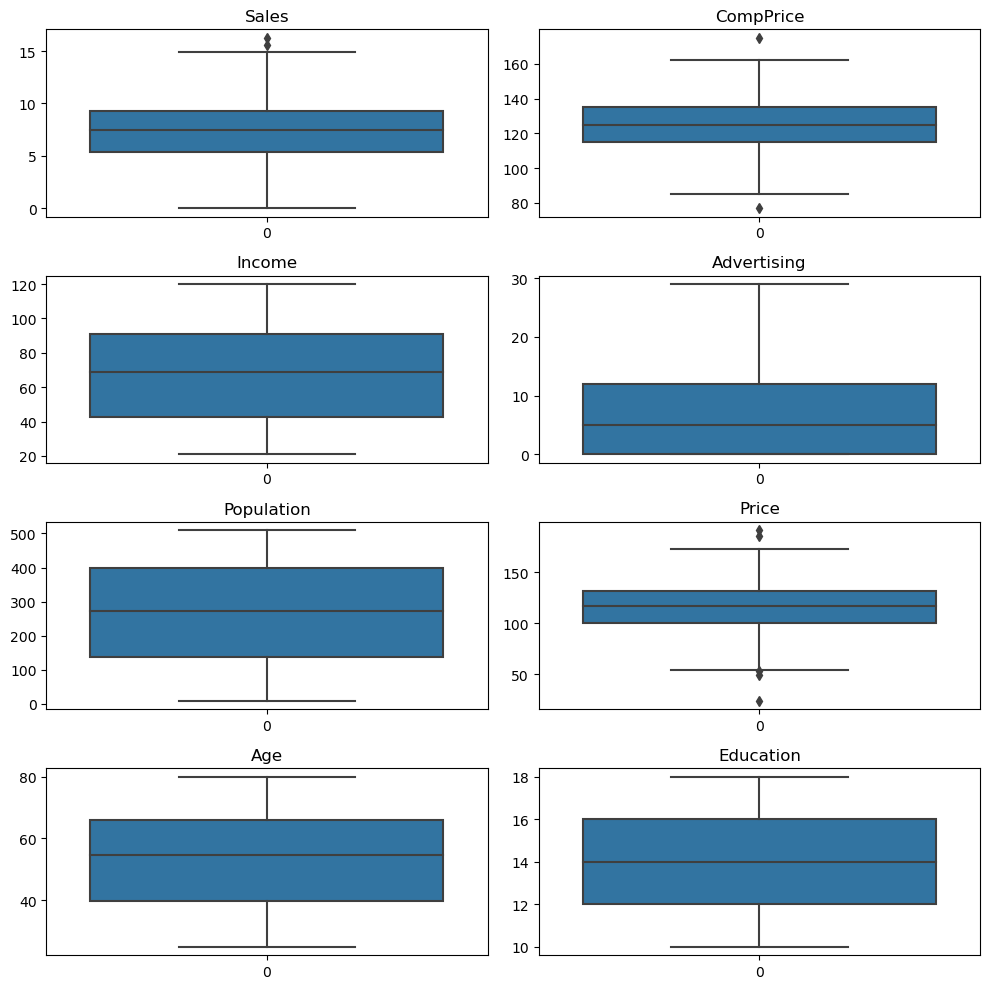
**Figure 5: Null values in the dataset**

Missing values in the dataset have been checked using the ‘isnull (). sum ()’ function, from which the observed null values in the dataset for all variables are 0 (**Refer to Figure 5**).

****

**Figure 6: Duplicate values in the dataset**

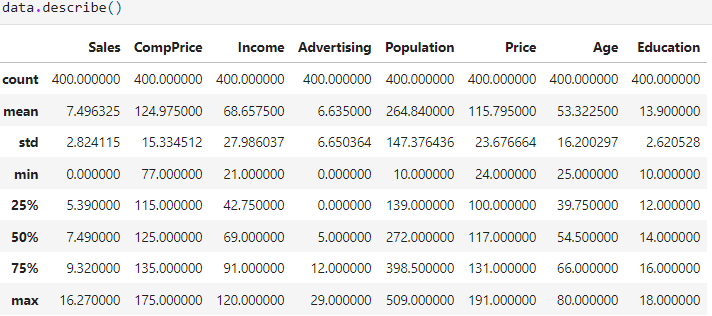
Duplicate values indicate data repetition, which can cause model overfitting, due to this, duplicate values have been checked using ‘duplicated (). sum ()’ from Pandas library (**Refer to Figure 6**). The observed duplicate values are 0, indicating non-existence of duplicate values in the dataset.

****

**Figure 7: Box plots for all numerical variables**

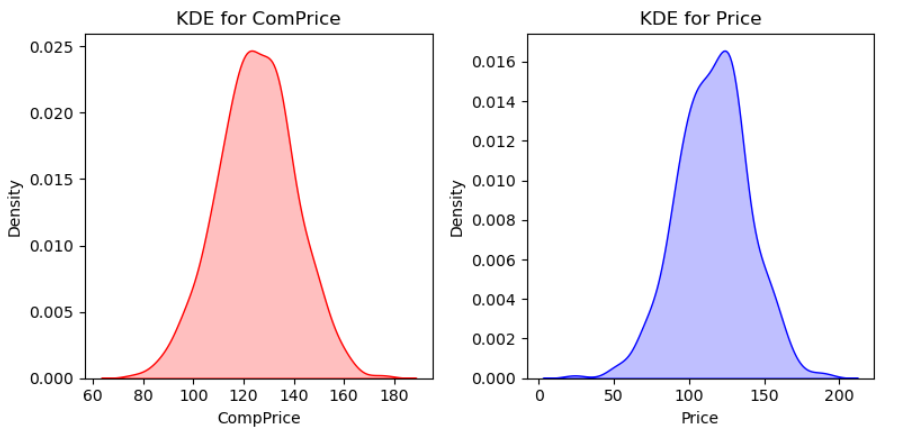
Box plot for all numerical variables (Sales, ComPrice, Income, Advertising, Population, Price, Age and Education) to identify potential outliers. From the box plots, it can be observed that none of the variables contains outliers (**Refer to Figure 7**).

## Exploratory data analysis (EDA)

****

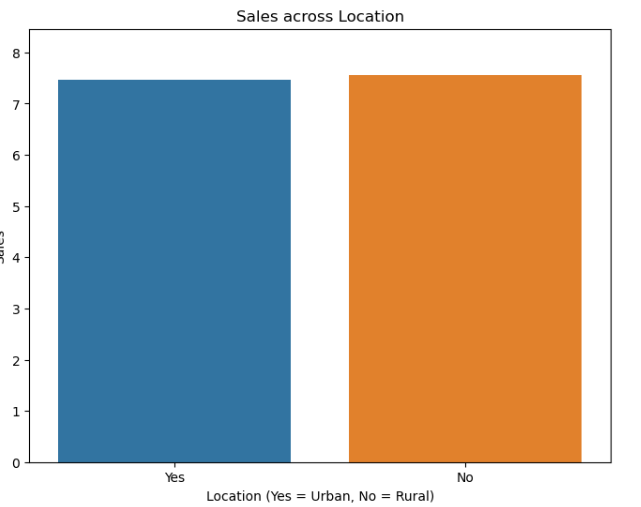
**Table 2: Summary statistics**

**Table 2** demonstrates the summary statistics (central tendency and variability) of numerical features, from which it can be observed that the mean value of competitor price is 124.97 with a standard deviation of 15.33, whereas the mean value of price is 115.79 with a standard deviation of 23.67. This infers that the company sells their products at a comparatively cheaper price point compared to its competitor.

****

**Figure 8: KDE plots for ‘ComPrice’ and ‘Price’**

Kernel Density plots for ‘ComPrice’ and ‘Price’ show a bell-shaped curve, indicating the presence of a normal distribution for both these variables (**Refer to Figure 8**).

****

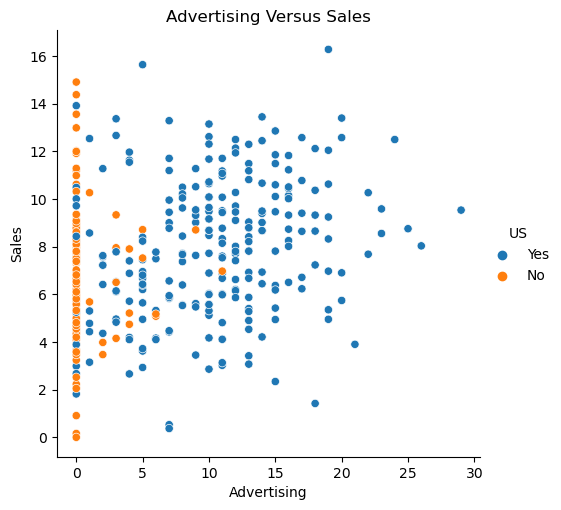
**Figure 9: Distribution of sales across location**

From **Figure 9**, it can be observed that sales are slightly higher in rural areas compared to urban areas, however, the price difference is negligibly small.



**Figure 10: Distribution of Sales and Price by Shelves location**

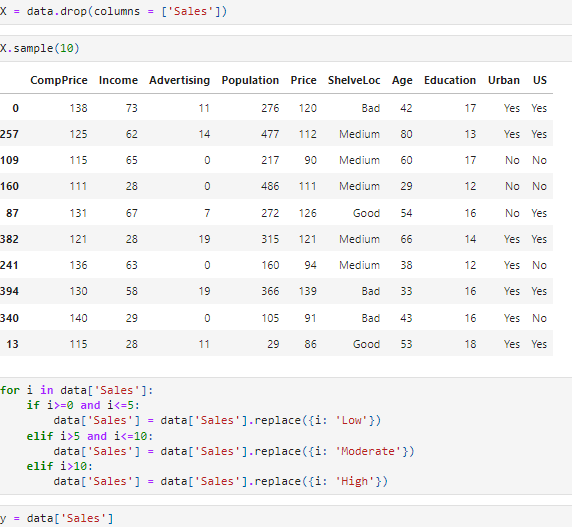
**Figure 10** presents two bar charts analyzing sales and prices based on shelf location (Bad, Good, Medium) and urban areas (Yes or No). The first chart shows higher sales for "Good" shelf locations across both urban and non-urban areas, while the second chart indicates similar prices across different shelf locations and regions.



**Figure 11: Scatter plot between ‘Sales’ and ‘Advertising’**

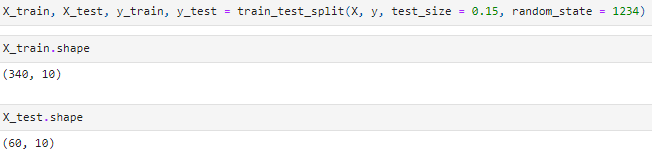
**Figure 11** shows the scatter plot between sales and advertising (segmented by region U.S. or not), from which a linear relationship between sales and advertising has been observed in the U.S. This reflects with the increase in advertising budget, sales of the company have increased.

## Model development



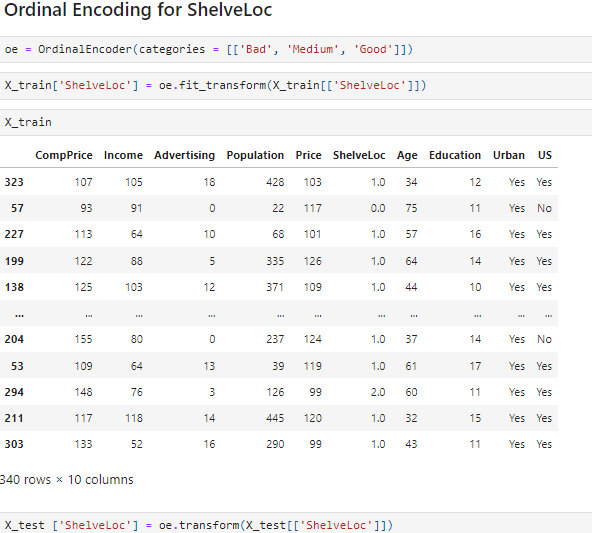
**Figure 12: Feature and target**

**Figure 12** shows the features and the target variable, from which it can be observed that target variable is Sales (which has been categorised into three different categories Low (0 to 5), Moderate (5 to 10) and High (more than 10)).



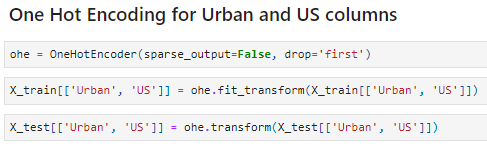
**Figure 13: train-test splitting**

**Figure 13** shows that the train\_test\_split function partitions the dataset into training (340 samples) and test (60 samples) subsets. With test\_size=0.15, 15% of the data is allocated to the test set. random\_state=1234 ensures the split is reproducible by fixing the random seed.



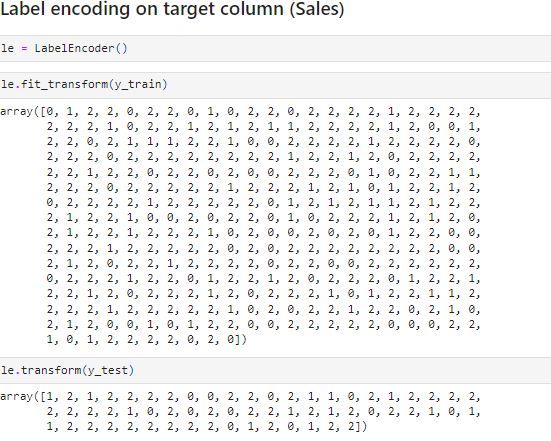
**Figure 14: Ordinal Encoding for ShelveLoc**

Ordinal Encoding assigns numerical values to ordered categories in ‘ShelveLoc’: ‘Bad’ = 0, ‘Medium’ = 1, and ‘Good’ = 2. This transforms ‘ShelveLoc’ into a numeric column, which is then applied to both training and test datasets for consistent feature representation **(Refer to Figure 14)**.



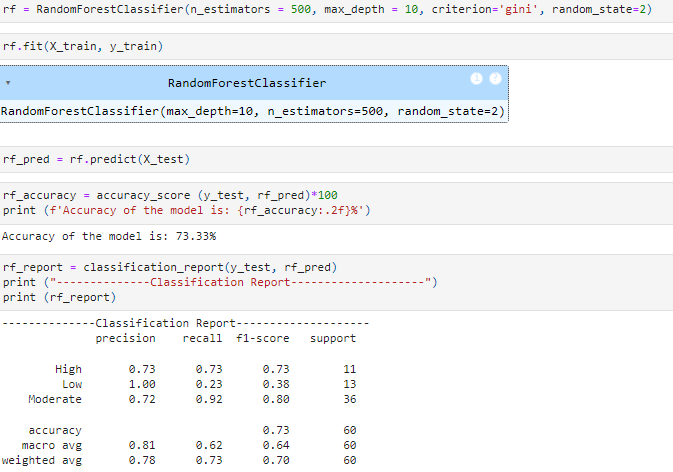
**Figure 15: One Hot Encoding for Urban and US Columns**

**Figure 15** represents that One-Hot Encoding converts categorical variables into binary columns. For the ‘Urban’ and ‘US’ columns, drop=’first’ avoids multicollinearity by excluding the first category, creating binary features for the remaining categories. This is applied to both training and test sets.



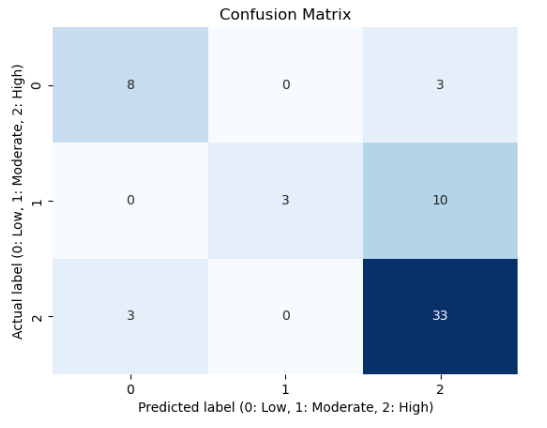
**Figure 16: Label encoding for target variable using LabelEncoder**

Label Encoding transforms categorical target values into numerical form for model processing. In this case, the LabelEncoder maps classes like ‘High’, ‘Low’, and ‘Moderate’ to integers (0, 1, 2). The training and test target values are encoded into these integer representations to facilitate model training and evaluation **(Refer to Figure 16)**.



**Figure 17: Model Architecture and model performance**

The Random Forest model uses 500 decision trees with a maximum depth of 10 to classify data based on the Gini impurity criterion. With an accuracy of 73.33%, it performs well on the ‘High’ and ‘Moderate’ classes but struggles with the ‘Low’ class, as shown by low recall and f1-scores **(Refer to Figure 17)**.



**Figure 18: Confusion Matrix**

The confusion matrix shows prediction results from a classification model. There are 8 true positives for ‘Low’ (label 0), 3 for ‘Moderate’ (label 1), and 33 for ‘High’ (label 2), with some misclassifications **(Refer to Figure 18)**.

# CHALLENGES FACED

The main challenge faced in this project is the treatment of the categorical variables in this project. The challenges have been tackled by selecting appropriate encoding methods (for example, LabelEncoder for the target variable, OneHotEncoder for nominal features and OrdinalEncoder for Ordinal features).

# COMPLEXITY LEVEL

The project can be considered as intermediate level mainly due to the complexity of the dataset.