Project Title: E-Commerce Object Classification Model

Name: Muhammad Razi Ur Rehman

Contents

Project Title: E-Commerce Object Classification Model

Data Preparation:

In this project, we have the openness to use any kind of dataset. We can create our own dummy data. But I prefer to get some kind of realistic kind of dataset. I found this amazing dataset on data.world. Here is the complete link to the dataset.
https://data.world/jegazhu/amazon-products-data
I am only using the first table "Amazon Sales Report.csv". This dataset is of a clothing brand. The Style is the specific name of the product. It can be any kind of pattern. SKU is like a stock-keeping unit and it is like a complete description but in short form. Then, we have an amount which is the price of the product. We also have some kind of currency.

Here, are the following steps that I have done during this step.

1- Loading the CSV file
2- Deleting the unnecessary columns to reduce dimensions.
3- Renaming the columns for better understanding

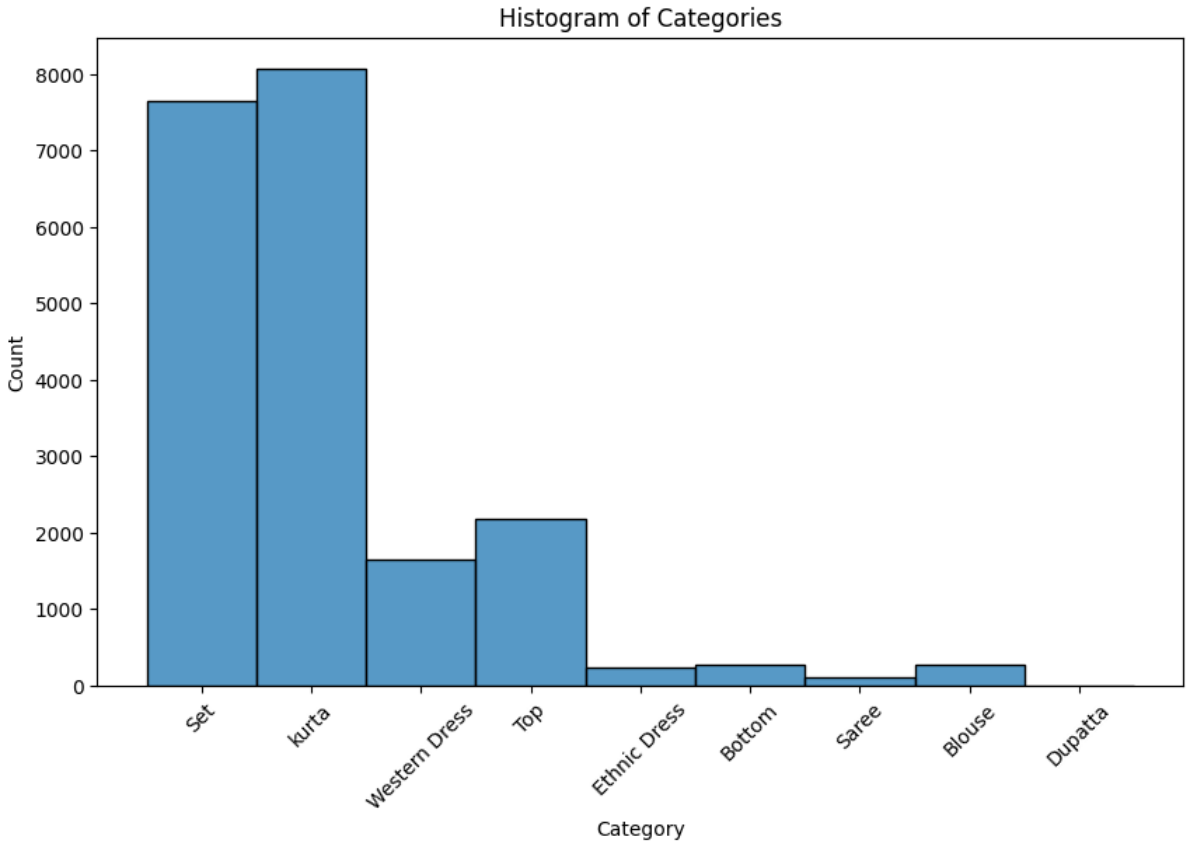
Data Preprocessing:
The following steps are taken along with their reasons:

1- **Dealing with null data:** There are many techniques to deal with null data. Like using mean, mode, medium, and deleting them. As we have a large dataset, we can just delete the rows having null values
2- **Dealing with unique values:** A single unique value of a column does not change anything but increases the computation and sometimes complexity. So, it is better to remove it beforehand.
3- **Dealing with Duplicates:** We remove the duplicates so that we do not overfit our model.
4- **Changing the type:** We change the type of columns into their specific type for better working and dealing with them later on.
5- **Save CSV file:** I want to save the preprocessed CSV file for later usage.

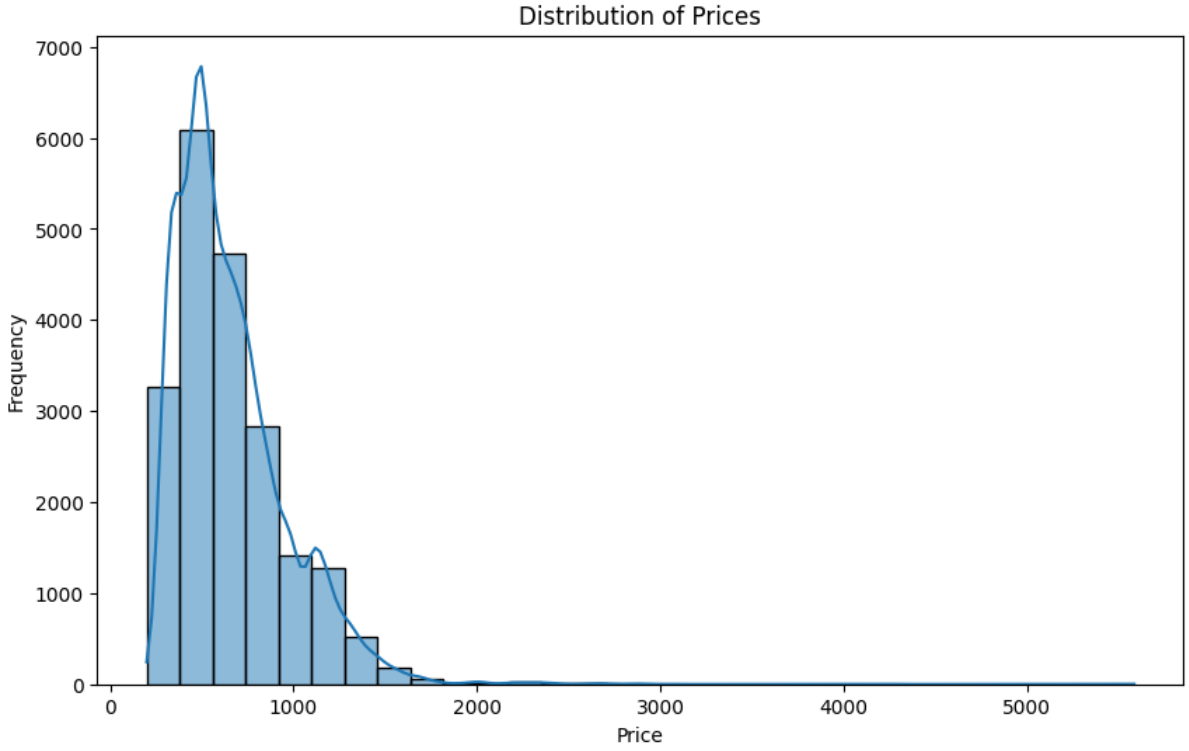
Data Visualization:

Data Visualization was not specifically mentioned in the requirement, but it is a necessary step before creating the model. It helps us to understand data better. The following steps are done in this step:

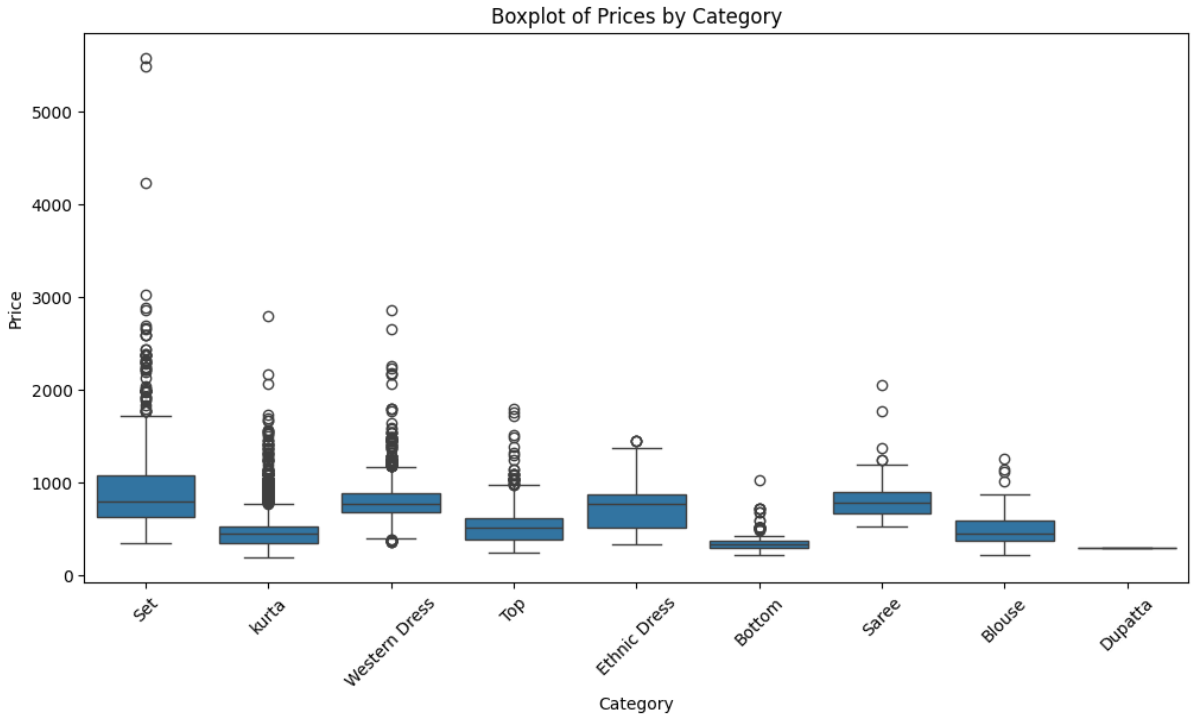
1- **Histogram of Category:** This step visualizes the categories count with other categories. This is a necessary step to see which category has the highest count or if the dataset is biased or not. After visualization of this, we can clearly say the dataset is biased towards the set and kurta.



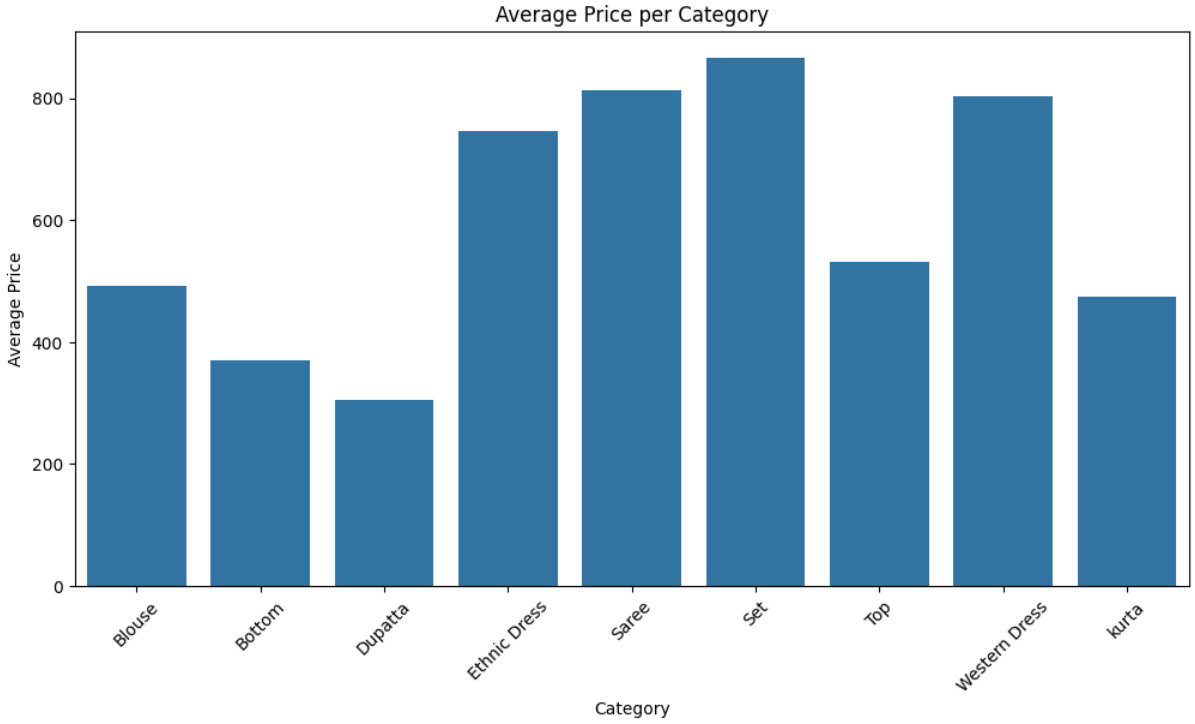
2- **Distribution of Prices:** Then, we are visualizing the price distribution.



3- **Boxplots:** One reason for visualizing boxplots of prices of all categories, is to identify the outliers in the dataset. Outliers can sometimes make the model complex and not easy to learn.



4- **Average Price per Category:** This bar plot is used to show the average price of each category so that we can decide which category is more diverse in terms of prices than others.



5- **Conclusion steps:** After carefully viewing the graphs, we can see the Dupatta is too much below the line, so we need to remove it from the category or need more data on Dupatta. Since we do not have more data on Dupatta, we remove it from our dataset. Another step is after viewing the boxplot, we should remove the outliers of the set as they are way above the line.

Data Conversion and Encoding:

In this section, we shall be encoding and scaling the dataset. The following steps are performed:

1- **Category Encoding:** We have performed category encoding using the simple mapping function
2- **Price Scaling:** We have performed price scaling using a custom standard scaler. The reason why we used the custom method instead of using Sklearn Standard Scaler is that now we do not need to save the scaler file.
3- **Text Tokenization:** We have performed text tokenization using the TensorFlow library. We also saved the layer of text tokenizer after converting it into the model.

Model Creation:

We have created a simple TensorFlow model for the classification of 8 categories. We first, define the input layer of name, description, and price. Convert input integers into dense vectors of a fixed length of 16 size using an embedding layer for both name and description. Then, Apply Global average pooling to them. Concatenate all three inputs and apply a dense layer having 32 neurons and Relu as an activation layer. Then, apply another fully connected ANN having 8 neurons and Softmax as an activation layer. This is the model summary:

```
Model: "model_2"
__________________________________________________________________________________________________
 Layer (type)                Output Shape              Param #   Connected to
==================================================================================================
 name_input (InputLayer)     [(None, 1)]               0         []

 desc_input (InputLayer)     [(None, 5)]               0         []

 embedding_4 (Embedding)     (None, 1, 16)             16000     ['name_input[0][0]']

 embedding_5 (Embedding)     (None, 5, 16)             16000     ['desc_input[0][0]']

 global_average_pooling1d_4  (None, 16)                0         ['embedding_4[0][0]']
  (GlobalAveragePooling1D)

 global_average_pooling1d_5  (None, 16)                0         ['embedding_5[0][0]']
  (GlobalAveragePooling1D)

 price_input (InputLayer)    [(None, 1)]               0         []

 concatenate_2 (Concatenate  (None, 33)                0         ['global_average_pooling1d_4[0
 )                                                               ][0]',
                                                                  'global_average_pooling1d_5[0
                                                                 ][0]',
                                                                  'price_input[0][0]']

 dense_4 (Dense)             (None, 32)                1088      ['concatenate_2[0][0]']

 dense_5 (Dense)             (None, 8)                 264       ['dense_4[0][0]']

==================================================================================================
Total params: 33352 (130.28 KB)
Trainable params: 33352 (130.28 KB)
Non-trainable params: 0 (0.00 Byte)
__________________________________________________________________________________________________
```

Model Training:

In this section, we train the model using the preprocessed and encoded data. In training, we set the batch size to 32 and the epoch to 10.

Model Evaluation:

In the evaluation step, the following is the result:

```
Loss: 0.010428916662931442, Accuracy: 0.996673047542572
```

Saving the model:

We save the model in the h5 format using the TensorFlow library. The model file name is “Object_Classification_Model.h5”.

Complete Model Testing:

In this section, we have loaded the main model, text tokenizers model (then separated the layers), and Scaling Price function. We have created a format input function to format the input according to the model input requirement. The run model function is used to run the model. The result is shown below:

```
1/1 [==============================] - 0s 39ms/step
'kurta'
```

Note:

I have done this work on Google Colab. You can also view my work using this link.

https://colab.research.google.com/drive/1F_DhIaHeZDqb8hA22e2HNX8HT8eRj9uT?usp=sharing