BigMart (1)

[1]: *# Importing all necessary libraries*

[2]: train_data = pd.read_csv('bigmart_train.csv') test_data = pd.read_csv('bigmart_test.csv')

(8523, 12) (5681, 11)

[3]: train_data.head()

1

Item Identifier - Product ID's, Not a relevant variable
Item Weight - Floating point type, Continous data
Item Fat Content - Categorical column, ordered data
Item Visibility - Continous data
Item Type - Not a relevant variable
Item MRP - Continous data
Outlet Identifier - Not a relevant variable
Outlet Establishment Year - Modify this column by subtracting current year with the establishment year
Outlet Size - Categorical columns, ordered
Outlet Location Type - Categorical, ordered
Outlet Type - Categorical, Not ordered
Item_Outlet_Sales - Target variable, continuous variable

[4]: train_data.drop(['Item_Identifier','Item_Type','Outlet_Identifier'], axis = 1,␣ ↪inplace = **True**)
test_data.drop(['Item_Identifier','Item_Type','Outlet_Identifier'], axis = 1,␣ ↪inplace = **True**)
train_data.dtypes

2

[6]: test_data.isna().sum()

1. Univariate Analysis
Plot the histogram of each column
Plot a correlation graph of each column with target variable 2. Bivariate analysis
Plot the scatter plot of all the variables

[7]: *# Replacing the missing values*

3

[10]: train_data['Item_Fat_Content'] = train_data['Item_Fat_Content'].replace(['low␣

↪fat','LF'],'Low Fat')
train_data['Item_Fat_Content'] = train_data['Item_Fat_Content'].

```
    →replace(['reg'],'Regular')
    train_data['Item_Fat_Content'].unique()
```

[10]: array(['Low Fat', 'Regular'], dtype=object)

[11]:
```
test_data['Item_Fat_Content'] = test_data['Item_Fat_Content'].replace(['low␣
    →fat','LF'],'Low Fat')
test_data['Item_Fat_Content'] = test_data['Item_Fat_Content'].
    →replace(['reg'],'Regular')
test_data['Item_Fat_Content'].unique()
```

[11]: array(['Low Fat', 'Regular'], dtype=object)

4

[14]: train_data['Outlet_Size'].unique()

5

[14]: array(['Medium', 'High', 'Small'], dtype=object)

---

[15]: train_data['Outlet_Location_Type'].unique()

[15]: array(['Tier 1', 'Tier 3', 'Tier 2'], dtype=object)

---

[16]: train_data['Outlet_Type'].unique()

[16]: array(['Supermarket Type1', 'Supermarket Type2', 'Grocery Store', 'Supermarket Type3'], dtype=object)

---

[18]: *#Creating label encoder for ordered data*

[21]: train_data.head()

6

7

8

[26]: new_test_data.isna().sum()

9

[27]: *# Splitting the data into train and test*

```
(7244, 11)
(1279, 11)
(7244, 1)
(1279, 1)
```

[28]: my_model = LinearRegression()

[29]: my_model.predict(new_test_data)

10

[30]: pd.concat([new_test_data,pd.DataFrame(my_model.predict(new_test_data))], axis =␣
1).to_csv('final_val.csv', index = **False**)

0.0.2 Ridge Regression

[35]: my_model = Ridge()

[36]: my_model = Lasso()

[37]: my_model = ElasticNet()

11

[ ]:

12