In [1]:
```
#display image using python
from IPython.display import Image
url = 'https://img.etimg.com/thumb/msid-71806721,width-650,imgsize-807917,,resiz
Image(url,height=300,width=400)
```

Out[1]:

In [3]:
```
#importing libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
import warnings
warnings.filterwarnings('ignore')
#importing the dataset
data = pd.read_csv(r'C:\Users\DELL\Desktop\FSDS\ML\29th- REGRESSION PROJECT\RESU
# Check the data
data.info()
```

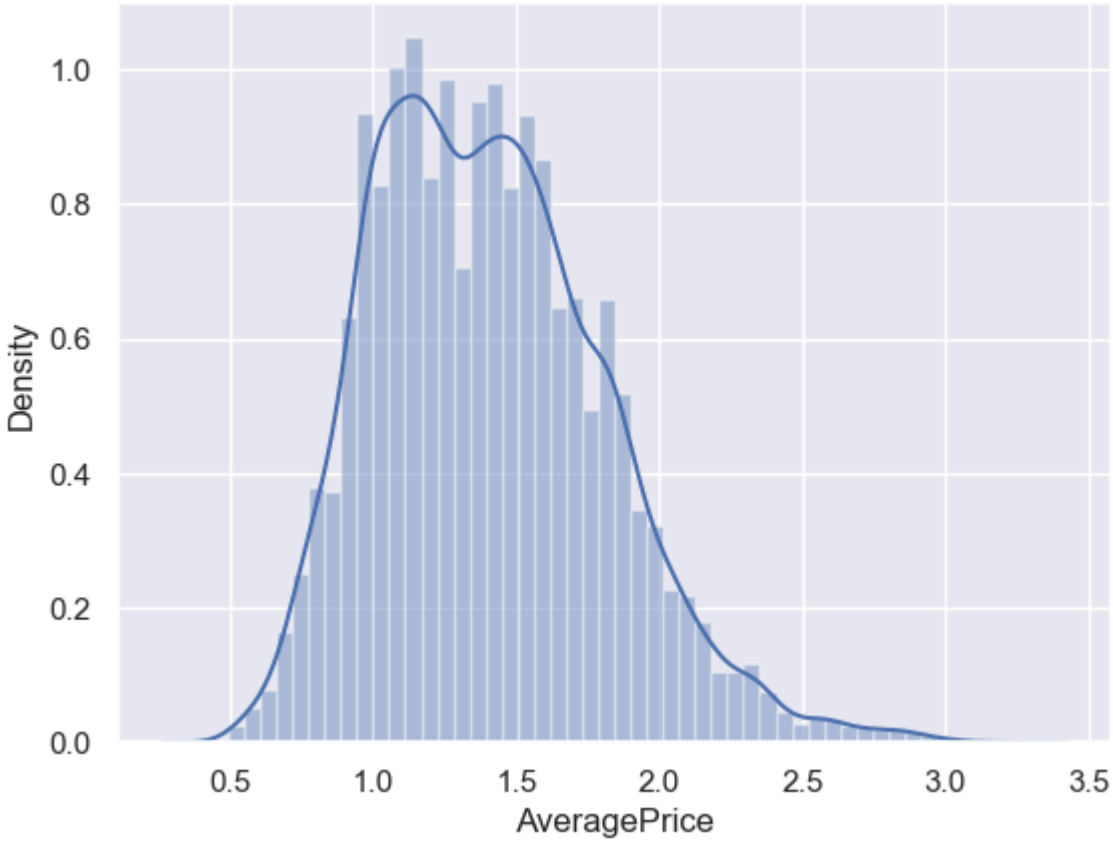
```
<class 'pandas.core.frame.DataFrame'>
Index: 18249 entries, 0 to 11
Data columns (total 13 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Date          18249 non-null  object
 1   AveragePrice  18249 non-null  float64
 2   Total Volume  18249 non-null  float64
 3   4046          18249 non-null  float64
 4   4225          18249 non-null  float64
 5   4770          18249 non-null  float64
 6   Total Bags    18249 non-null  float64
 7   Small Bags    18249 non-null  float64
 8   Large Bags    18249 non-null  float64
 9   XLarge Bags   18249 non-null  float64
 10  type          18249 non-null  object
 11  year          18249 non-null  int64
 12  region        18249 non-null  object
dtypes: float64(9), int64(1), object(3)
memory usage: 1.9+ MB
```

In [4]:
```
data.head(3)
```

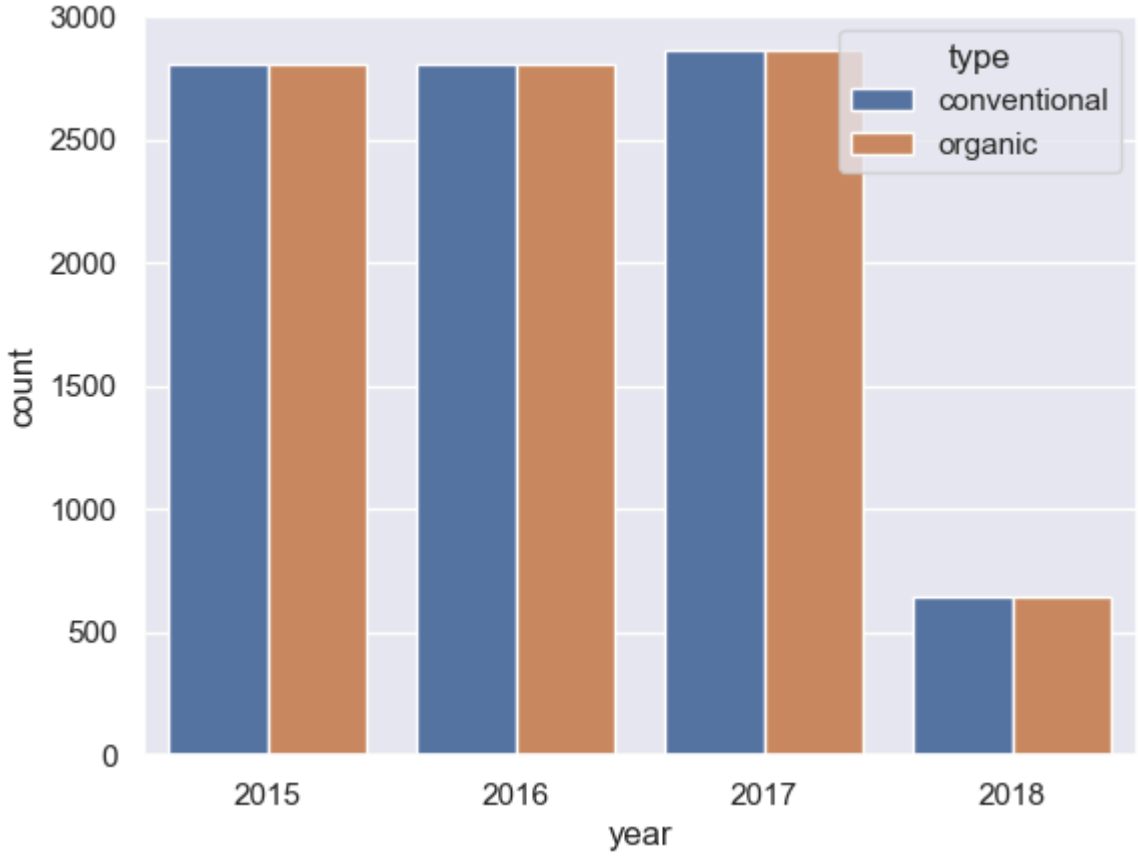
Out[4]:

	Date	AveragePrice	Total Volume	4046	4225	4770	Total Bags	Small Bags	Large Bags
0	2015-12-27	1.33	64236.62	1036.74	54454.85	48.16	8696.87	8603.62	93.25
1	2015-12-20	1.35	54876.98	674.28	44638.81	58.33	9505.56	9408.07	97.49
2	2015-12-13	0.93	118220.22	794.70	109149.67	130.50	8145.35	8042.21	103.14

In [5]:
```
sns.distplot(data['AveragePrice']);
```



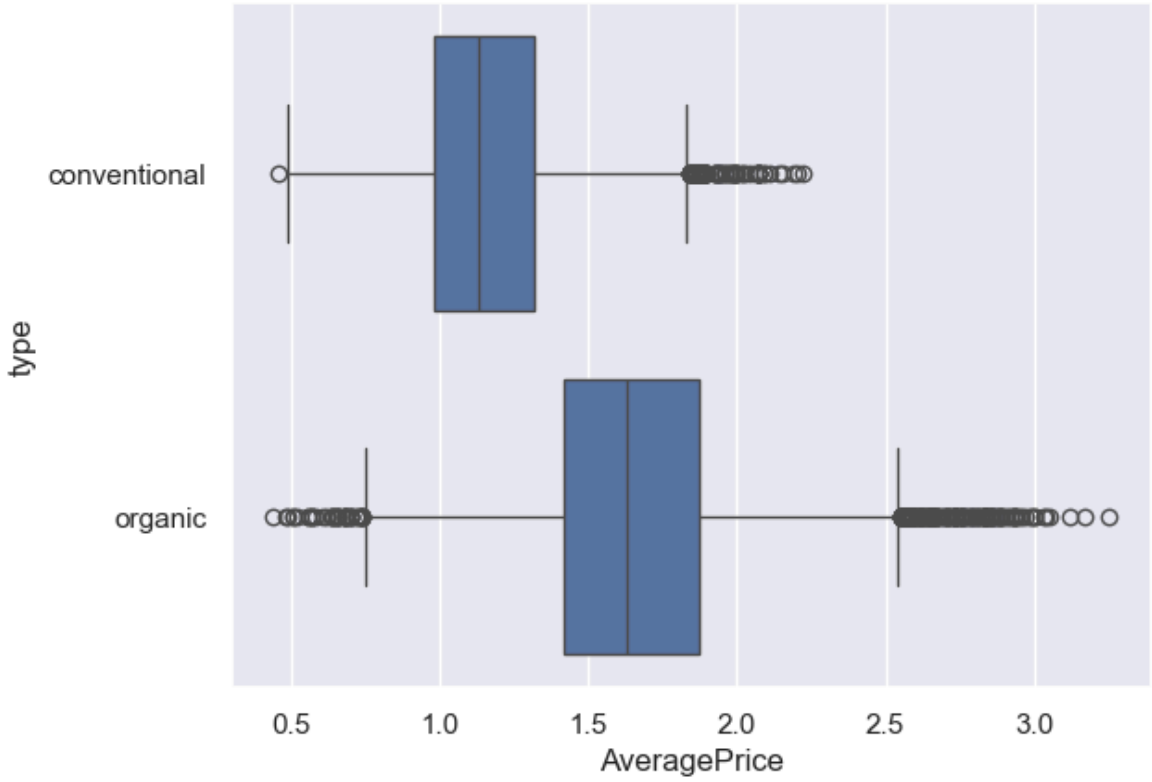
In [6]:
```
sns.countplot(x='year',data=data,hue='type');
```



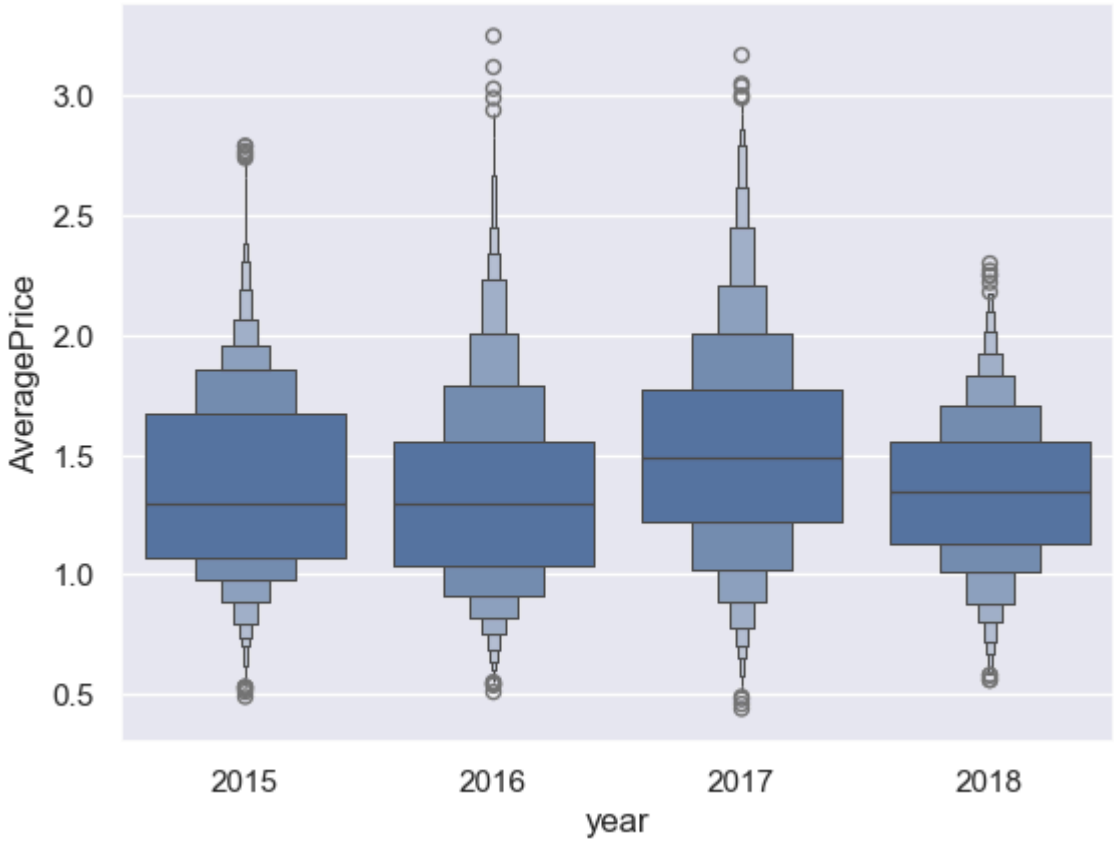
In [7]:
```
data.year.value_counts()
```

Out[7]:
```
year
2017    5722
2016    5616
2015    5615
2018    1296
Name: count, dtype: int64
```

In [8]:
```
sns.boxplot(y="type", x="AveragePrice", data=data);
```



In [9]:
```
data.year=data.year.apply(str)
sns.boxenplot(x="year", y="AveragePrice", data=data);
```

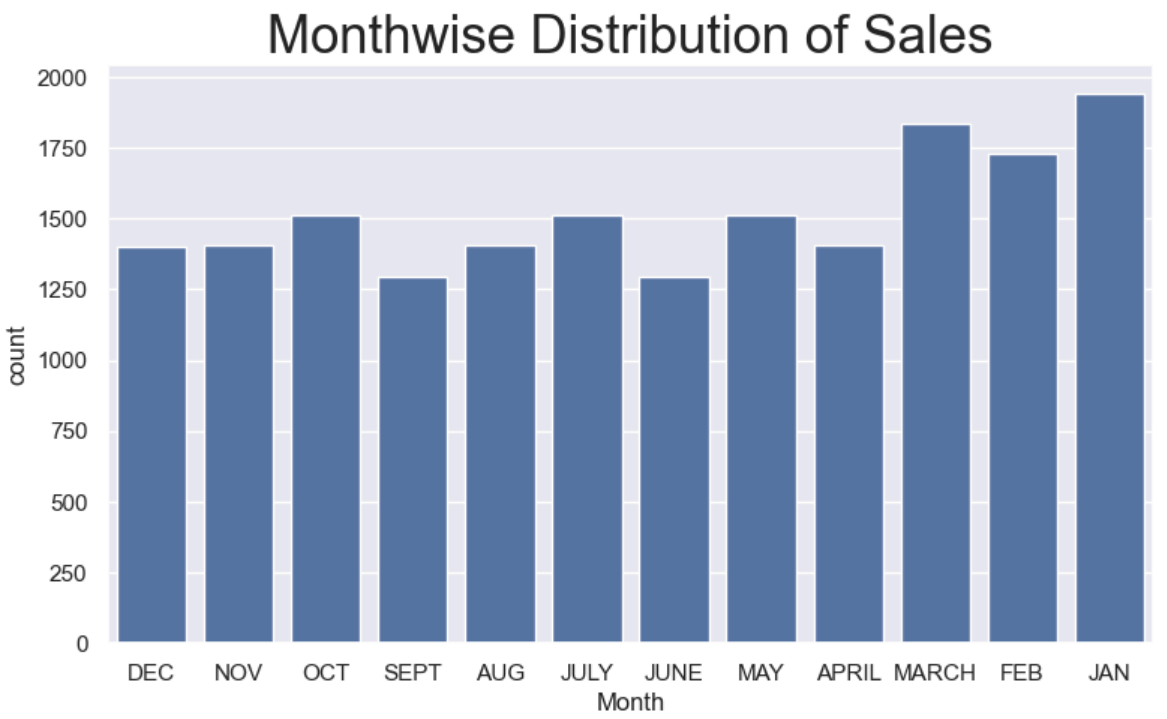


Dealing with categorical features.

In [10]:
```
data['type']= data['typeplt.figure(figsize=(9,5))
sns.countplot(data['Month'])
plt.title('Monthwise Distribution of Sales',fontdict={'fontsize':25});'].map({'c
```

```
# Extracting month from date column.
data.Date = data.Date.apply(pd.to_datetime)
data['Month']=data['Date'].apply(lambda x:x.month)
data.drop('Date',axis=1,inplace=True)
data.Month = data.Month.map({1:'JAN',2:'FEB',3:'MARCH',4:'APRIL',5:'MAY',6:'JUNE
```

In [12]:
```
plt.figure(figsize=(9,5))
sns.countplot(x=data['Month'])
plt.title('Monthwise Distribution of Sales',fontdict={'fontsize':25});
```



Preparing data for ML models

In [13]:
```
# Creating dummy variables
dummies = pd.get_dummies(data[['year','region','Month']],drop_first=True)
df_dummies = pd.concat([data[['Total Volume', '4046', '4225', '4770', 'Total Bag
       'Small Bags', 'Large Bags', 'XLarge Bags', 'type']],dummies],axis=1)
target = data['AveragePrice']

# Splitting data into training and test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(df_dummies,target,test_size=

# Standardizing the data
cols_to_std = ['Total Volume', '4046', '4225', '4770', 'Total Bags', 'Small Bags
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
scaler.fit(X_train[cols_to_std])
X_train[cols_to_std] = scaler.transform(X_train[cols_to_std])
X_test[cols_to_std] = scaler.transform(X_test[cols_to_std])
```

In [14]:
```
#importing ML models from scikit-learn
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
from sklearn.neighbors import KNeighborsRegressor
```

```
from xgboost import XGBRegressor
from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score
```

In [15]:
```
#to save time all models can be applied once using for loop
regressors = {
    'Linear Regression' : LinearRegression(),
    'Decision Tree' : DecisionTreeRegressor(),
    'Random Forest' : RandomForestRegressor(),
    'Support Vector Machines' : SVR(gamma=1),
    'K-nearest Neighbors' : KNeighborsRegressor(n_neighbors=1),
    'XGBoost' : XGBRegressor()
}
results=pd.DataFrame(columns=['MAE','MSE','R2-score'])
for method,func in regressors.items():
    model = func.fit(X_train,y_train)
    pred = model.predict(X_test)
    results.loc[method]= [np.round(mean_absolute_error(y_test,pred),3),
                          np.round(mean_squared_error(y_test,pred),3),
                          np.round(r2_score(y_test,pred),3)
                         ]
```

Deep Neural Network

In [20]:
```
!pip install tensorflow
```

```
Requirement already satisfied: tensorflow in c:\users\dell\anaconda3\lib\site-pac
kages (2.20.0)
Requirement already satisfied: absl-py>=1.0.0 in c:\users\dell\anaconda3\lib\site
-packages (from tensorflow) (2.3.1)
Requirement already satisfied: astunparse>=1.6.0 in c:\users\dell\anaconda3\lib\s
ite-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=24.3.25 in c:\users\dell\anaconda3\li
b\site-packages (from tensorflow) (25.2.10)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in c:\users\de
ll\anaconda3\lib\site-packages (from tensorflow) (0.6.0)
Requirement already satisfied: google_pasta>=0.1.1 in c:\users\dell\anaconda3\lib
\site-packages (from tensorflow) (0.2.0)
Requirement already satisfied: libclang>=13.0.0 in c:\users\dell\anaconda3\lib\si
te-packages (from tensorflow) (18.1.1)
Requirement already satisfied: opt_einsum>=2.3.2 in c:\users\dell\anaconda3\lib\s
ite-packages (from tensorflow) (3.4.0)
Requirement already satisfied: packaging in c:\users\dell\anaconda3\lib\site-pack
ages (from tensorflow) (24.1)
Requirement already satisfied: protobuf>=5.28.0 in c:\users\dell\anaconda3\lib\si
te-packages (from tensorflow) (6.32.0)
Requirement already satisfied: requests<3,>=2.21.0 in c:\users\dell\anaconda3\lib
\site-packages (from tensorflow) (2.32.3)
Requirement already satisfied: setuptools in c:\users\dell\anaconda3\lib\site-pac
kages (from tensorflow) (75.1.0)
Requirement already satisfied: six>=1.12.0 in c:\users\dell\anaconda3\lib\site-pa
ckages (from tensorflow) (1.16.0)
Requirement already satisfied: termcolor>=1.1.0 in c:\users\dell\anaconda3\lib\si
te-packages (from tensorflow) (3.1.0)
Requirement already satisfied: typing_extensions>=3.6.6 in c:\users\dell\anaconda
3\lib\site-packages (from tensorflow) (4.11.0)
Requirement already satisfied: wrapt>=1.11.0 in c:\users\dell\anaconda3\lib\site-
packages (from tensorflow) (1.14.1)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in c:\users\dell\anaconda3\lib
\site-packages (from tensorflow) (1.74.0)
Requirement already satisfied: tensorboard~=2.20.0 in c:\users\dell\anaconda3\lib
\site-packages (from tensorflow) (2.20.0)
Requirement already satisfied: keras>=3.10.0 in c:\users\dell\anaconda3\lib\site-
packages (from tensorflow) (3.11.3)
Requirement already satisfied: numpy>=1.26.0 in c:\users\dell\anaconda3\lib\site-
packages (from tensorflow) (1.26.4)
Requirement already satisfied: h5py>=3.11.0 in c:\users\dell\anaconda3\lib\site-p
ackages (from tensorflow) (3.11.0)
Requirement already satisfied: ml_dtypes<1.0.0,>=0.5.1 in c:\users\dell\anaconda3
\lib\site-packages (from tensorflow) (0.5.3)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\dell\anaconda
3\lib\site-packages (from requests<3,>=2.21.0->tensorflow) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in c:\users\dell\anaconda3\lib\site-p
ackages (from requests<3,>=2.21.0->tensorflow) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\dell\anaconda3\lib
\site-packages (from requests<3,>=2.21.0->tensorflow) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\dell\anaconda3\lib
\site-packages (from requests<3,>=2.21.0->tensorflow) (2025.8.3)
Requirement already satisfied: markdown>=2.6.8 in c:\users\dell\anaconda3\lib\sit
e-packages (from tensorboard~=2.20.0->tensorflow) (3.4.1)
Requirement already satisfied: pillow in c:\users\dell\anaconda3\lib\site-package
s (from tensorboard~=2.20.0->tensorflow) (10.4.0)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in c:\users
\dell\anaconda3\lib\site-packages (from tensorboard~=2.20.0->tensorflow) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in c:\users\dell\anaconda3\lib\sit
e-packages (from tensorboard~=2.20.0->tensorflow) (3.0.3)
```

```
Requirement already satisfied: wheel<1.0,>=0.23.0 in c:\users\dell\anaconda3\lib
\site-packages (from astunparse>=1.6.0->tensorflow) (0.44.0)
Requirement already satisfied: rich in c:\users\dell\anaconda3\lib\site-packages
(from keras>=3.10.0->tensorflow) (13.7.1)
Requirement already satisfied: namex in c:\users\dell\anaconda3\lib\site-packages
(from keras>=3.10.0->tensorflow) (0.1.0)
Requirement already satisfied: optree in c:\users\dell\anaconda3\lib\site-package
s (from keras>=3.10.0->tensorflow) (0.17.0)
Requirement already satisfied: MarkupSafe>=2.1.1 in c:\users\dell\anaconda3\lib\s
ite-packages (from werkzeug>=1.0.1->tensorboard~=2.20.0->tensorflow) (2.1.3)
Requirement already satisfied: markdown-it-py>=2.2.0 in c:\users\dell\anaconda3\l
ib\site-packages (from rich->keras>=3.10.0->tensorflow) (2.2.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in c:\users\dell\anaconda3
\lib\site-packages (from rich->keras>=3.10.0->tensorflow) (2.15.1)
Requirement already satisfied: mdurl~=0.1 in c:\users\dell\anaconda3\lib\site-pac
kages (from markdown-it-py>=2.2.0->rich->keras>=3.10.0->tensorflow) (0.1.0)
```

In [25]:
```
# Splitting train set into training and validation sets.
X_train, X_val, y_train, y_val = train_test_split(X_train,y_train,test_size=0.20

#importing tensorflow libraries
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Activation,Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping

#creating model
model = Sequential()
model.add(Dense(76,activation='relu',kernel_initializer=tf.random_uniform_initia
    bias_initializer=tf.random_uniform_initializer(minval=-0.1, maxval=0.1)))
model.add(Dense(200,activation='relu',kernel_initializer=tf.random_uniform_initi
    bias_initializer=tf.random_uniform_initializer(minval=-0.1, maxval=0.1)))
model.add(Dropout(0.5))
model.add(Dense(200,activation='relu',kernel_initializer=tf.random_uniform_initi
    bias_initializer=tf.random_uniform_initializer(minval=-0.1, maxval=0.1)))
model.add(Dropout(0.5))
model.add(Dense(200,activation='relu',kernel_initializer=tf.random_uniform_initi
    bias_initializer=tf.random_uniform_initializer(minval=-0.1, maxval=0.1)))
model.add(Dropout(0.5))
model.add(Dense(1))

model.compile(optimizer='Adam', loss='mean_squared_error')
early_stop = EarlyStopping(monitor='val_loss', mode='min', verbose=0, patience=1
```

In [26]:
```
print(X_train.dtypes)
print(y_train.dtypes)
```

```
Total Volume    float64
4046            float64
4225            float64
4770            float64
Total Bags      float64
                 ...
Month_MARCH       bool
Month_MAY         bool
Month_NOV         bool
Month_OCT         bool
Month_SEPT        bool
Length: 76, dtype: object
float64
```

In [27]:
```
X_train = pd.get_dummies(X_train)
X_val   = pd.get_dummies(X_val)

# Align columns in case train/val differ
X_train, X_val = X_train.align(X_val, join='left', axis=1, fill_value=0)
```

In [28]:
```
X_train = X_train.astype('float32')
X_val   = X_val.astype('float32')
y_train = y_train.astype('float32')
y_val   = y_val.astype('float32')
```

In [29]:
```
model.fit(
    x=X_train.values,
    y=y_train.values,
    validation_data=(X_val.values, y_val.values),
    batch_size=100,
    epochs=150,
    callbacks=[early_stop]
)
```

```
Epoch 1/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 5s 14ms/step - loss: 0.3503 - val_loss: 0.0875
Epoch 2/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 1s 9ms/step - loss: 0.1307 - val_loss: 0.0707
Epoch 3/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.1091 - val_loss: 0.0681
Epoch 4/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0939 - val_loss: 0.0681
Epoch 5/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0921 - val_loss: 0.0518
Epoch 6/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0815 - val_loss: 0.0686
Epoch 7/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0790 - val_loss: 0.0501
Epoch 8/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0750 - val_loss: 0.0464
Epoch 9/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0690 - val_loss: 0.0491
Epoch 10/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0683 - val_loss: 0.0477
Epoch 11/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0601 - val_loss: 0.0454
Epoch 12/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0594 - val_loss: 0.0414
Epoch 13/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 0.0574 - val_loss: 0.0426
Epoch 14/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - loss: 0.0549 - val_loss: 0.0444
Epoch 15/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0535 - val_loss: 0.0400
Epoch 16/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0515 - val_loss: 0.0398
Epoch 17/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - loss: 0.0506 - val_loss: 0.0382
Epoch 18/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - loss: 0.0488 - val_loss: 0.0362
Epoch 19/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.0493 - val_loss: 0.0368
Epoch 20/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0468 - val_loss: 0.0362
Epoch 21/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0462 - val_loss: 0.0359
Epoch 22/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 0.0444 - val_loss: 0.0399
Epoch 23/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0452 - val_loss: 0.0372
Epoch 24/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 0.0453 - val_loss: 0.0367
Epoch 25/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0434 - val_loss: 0.0340
Epoch 26/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - loss: 0.0414 - val_loss: 0.0355
Epoch 27/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.0427 - val_loss: 0.0355
Epoch 28/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0404 - val_loss: 0.0371
Epoch 29/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0395 - val_loss: 0.0337
Epoch 30/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0412 - val_loss: 0.0334
```

```
Epoch 31/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0398 - val_loss: 0.0334
Epoch 32/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 1s 11ms/step - loss: 0.0389 - val_loss: 0.0337
Epoch 33/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0378 - val_loss: 0.0338
Epoch 34/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0351 - val_loss: 0.0348
Epoch 35/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0364 - val_loss: 0.0372
Epoch 36/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0346 - val_loss: 0.0324
Epoch 37/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - loss: 0.0340 - val_loss: 0.0394
Epoch 38/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 1s 9ms/step - loss: 0.0346 - val_loss: 0.0326
Epoch 39/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0332 - val_loss: 0.0340
Epoch 40/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0337 - val_loss: 0.0330
Epoch 41/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 0.0324 - val_loss: 0.0344
Epoch 42/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0335 - val_loss: 0.0333
Epoch 43/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 0.0333 - val_loss: 0.0332
Epoch 44/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0318 - val_loss: 0.0331
Epoch 45/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0319 - val_loss: 0.0339
Epoch 46/150
53/53 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 0.0305 - val_loss: 0.0334
```

Out[29]: <keras.src.callbacks.history.History at 0x2ae003633b0>

In [31]:
```
losses = pd.DataFrame(model.history.history)
losses[['loss','val_loss']].plot();
```

In [32]:
```
dnn_pred = model.predict(X_test)
```

```
172/172 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step
```

Results table

In [35]:
```
results.loc['Deep Neural Network']=[round(mean_absolute_error(y_test,dnn_pred),3
                                    round(r2_score(y_test,dnn_pred),3)]
results
```

Out[35]:

	MAE	MSE	R2
Deep Neural Network	0.126	0.032	0.805

In [36]:
```
f"10% of mean of target variable is {np.round(0.1 * data.AveragePrice.mean(),3)}
```

Out[36]:
```
'10% of mean of target variable is 0.141'
```

In [39]:
```
results.sort_values('R2-score',ascending=False).style.background_gradient(cmap='
```

```
---------------------------------
KeyErrorTraceback (most recent call last)
~\AppData\Local\Temp\ipykernel_7096\2095228853.py in ?()
----> 1 results.sort_values('R2-score',ascending=False).style.background_gradient
(cmap='Greens',subset=['R2-score'])

~\anaconda3\Lib\site-packages\pandas\core\frame.py in ?(self, by, axis, ascendin
g, inplace, kind, na_position, ignore_index, key)
   7185             )
   7186         elif len(by):
   7187             # len(by) == 1
   7188
-> 7189             k = self._get_label_or_level_values(by[0], axis=axis)
   7190
   7191             # need to rewrap column in Series to apply key function
   7192             if key is not None:

~\anaconda3\Lib\site-packages\pandas\core\generic.py in ?(self, key, axis)
   1907             values = self.xs(key, axis=other_axes[0])._values
   1908         elif self._is_level_reference(key, axis=axis):
   1909             values = self.axes[axis].get_level_values(key)._values
   1910         else:
-> 1911             raise KeyError(key)
   1912
   1913         # Check for duplicates
   1914         if values.ndim > 1:

KeyError: 'R2-score'
```

In [40]:
```
results.sort_values('R2', ascending=False).style.background_gradient(cmap='Green
```

Out[40]:

	MAE	MSE	R2
Deep Neural Network	0.126000	0.032000	0.805000

In [41]:
```
results = results.rename(columns={'R2': 'R2-score'})
```

In [42]:
```
results.sort_values('R2-score', ascending=False).style.background_gradient(cmap=
```

Out[42]:

	MAE	MSE	R2-score
Deep Neural Network	0.126000	0.032000	0.805000

In [43]:
```
print(results.columns)
```

```
Index(['MAE', 'MSE', 'R2-score'], dtype='object')
```

In [ ]: