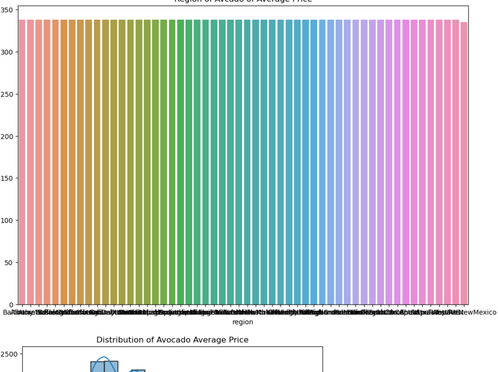
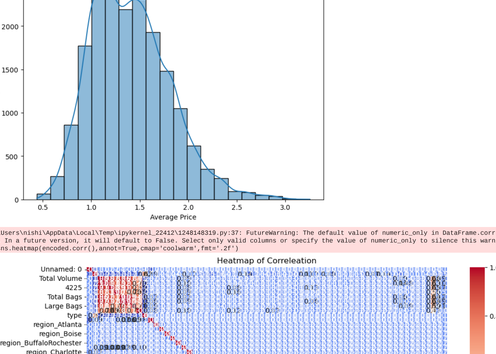
In [1]:

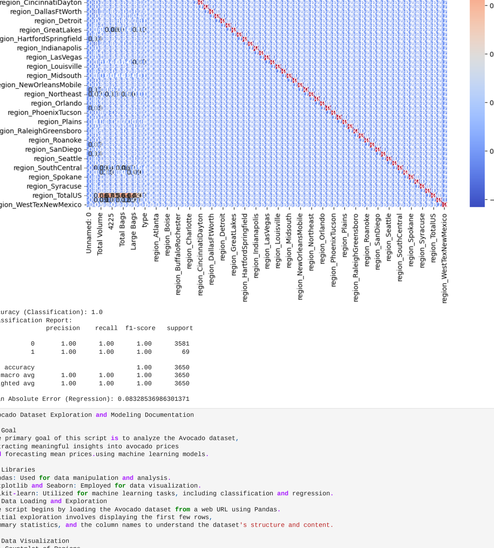
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier,RandomForestRegressor
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report,mean_absolute_error

web_url = "https://github.com/dsrscientist/Data-Science-ML-Capstone-Projects/raw/master/avocado.csv.zip"
data = pd.read_csv(web_url, compression='zip')

print(data.head())
print(data.describe())

print(data.columns)

plt.figure(figsize=(12,8))
sns.countplot(x='region',data=data)
plt.title('Region of Avcado of Average Price')
plt.show()

# Assuming 'data' is your DataFrame containing the 'AveragePrice' column


plt.figure(figsize=(8, 6))
sns.histplot(data['AveragePrice'], bins=20, kde=True)
plt.title('Distribution of Avocado Average Price')
plt.xlabel('Average Price')
plt.ylabel('Frequency')
plt.show()

le=LabelEncoder()
data['type']=le.fit_transform(data['type'])
encoded=pd.get_dummies(data, columns= ['region'], drop_first=True)


plt.figure(figsize=(12,8))
sns.heatmap(encoded.corr(),annot=True,cmap='coolwarm',fmt='.2f')
plt.title('Heatmap of Correleation')
plt.show()

X=encoded.drop(columns=['region_orlando' and 'Date'])
y = encoded['region_Orlando']

X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.8, random_state=42)

model=RandomForestClassifier(random_state=42)
model.fit(X_train,y_train)

# Evaluate Classification Model
y_pred = model.predict(X_test)
accuracy = model.score(X_test, y_test)
print('Accuracy (Classification):', accuracy)
print("Classification Report:\n", classification_report(y_test, y_pred))

# Step 5: Regression Task - Predict AveragePrice
# Prepare data for regression
X_reg = data.drop(columns=['Date', 'AveragePrice'])
y_reg = data['AveragePrice']

# Handle categorical variables with one-hot encoding
X_reg = pd.get_dummies(X_reg, drop_first=True)

# Drop non-numeric columns
X_reg = X_reg.select_dtypes(include='number')

# Train-test split for regression
X_reg_train, X_reg_test, y_reg_train, y_reg_test = train_test_split(X_reg, y_reg, test_size=0.2, random_state=42)

# Model Training - Random Forest Regressor
regressor = RandomForestRegressor(random_state=42)
regressor.fit(X_reg_train, y_reg_train)

# Evaluate Regression Model
y_reg_pred = regressor.predict(X_reg_test)
mae = mean_absolute_error(y_reg_test, y_reg_pred)
print("Mean Absolute Error (Regression):", mae)
```

```
   Unnamed: 0        Date  AveragePrice  Total Volume     4046     4225  \
0           0  2015-12-27          1.33      64236.62  1036.74  54454.85
1           1  2015-12-20          1.35      54876.98   674.28  44638.81
2           2  2015-12-13          0.93     118220.22   794.70  109149.67
3           3  2015-12-06          1.08      78992.15  1132.00   71976.41
4           4  2015-11-29          1.28      51039.60   941.48   43838.39

       4770  Total Bags  Small Bags  Large Bags  XLarge Bags          type  \
0     48.16     8696.87     8603.62       93.25          0.0  conventional
1     58.33     9505.56     9408.07       97.49          0.0  conventional
2    130.50     8145.35     8042.21      103.14          0.0  conventional
3     72.58     5811.16     5677.40      133.76          0.0  conventional
4     75.78     6183.95     5986.26      197.69          0.0  conventional

   year  region
0  2015  Albany
1  2015  Albany
2  2015  Albany
3  2015  Albany
4  2015  Albany
       Unnamed: 0  AveragePrice  Total Volume          4046          4225  \
count  18249.000000  18249.000000  1.824900e+04  1.824900e+04  1.824900e+04
mean      24.232232      1.405978  8.506440e+05  2.930084e+05  2.951546e+05
std       15.481045      0.402677  3.453545e+06  1.264989e+06  1.204120e+06
min        0.000000      0.440000  8.456000e+01  0.000000e+00  0.000000e+00
25%       10.000000      1.100000  1.083858e+04  8.540700e+02  3.008780e+03
50%       24.000000      1.370000  1.073768e+05  8.645300e+03  2.906102e+04
75%       38.000000      1.660000  4.329623e+05  1.110202e+05  1.502069e+05
max       52.000000      3.250000  6.250565e+07  2.274362e+07  2.047057e+07

               4770    Total Bags    Small Bags    Large Bags    XLarge Bags  \
count  1.824900e+04  1.824900e+04  1.824900e+04  1.824900e+04  18249.000000
mean   2.283974e+04  2.396392e+05  1.821947e+05  5.433809e+04    3106.426507
std    1.074641e+05  9.862424e+05  7.461785e+05  2.439664e+05   17692.894652
min    0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00       0.000000
25%    0.000000e+00  5.088640e+03  2.849420e+03  1.274700e+02       0.000000
50%    1.849900e+02  3.974383e+04  2.636282e+04  2.647710e+03       0.000000
75%    6.243420e+03  1.107834e+05  8.333767e+04  2.202925e+04     132.500000
max    2.546439e+06  1.937313e+07  1.338459e+07  5.719097e+06  551693.650000

              year
count  18249.000000
mean    2016.147899
std        0.939938
min     2015.000000
25%     2015.000000
50%     2016.000000
75%     2017.000000
max     2018.000000
Index(['Unnamed: 0', 'Date', 'AveragePrice', 'Total Volume', '4046', '4225',
       '4770', 'Total Bags', 'Small Bags', 'Large Bags', 'XLarge Bags', 'type',
       'year', 'region'],
      dtype='object')
```






```
C:\Users\nishi\AppData\Local\Temp\ipykernel_22412\1248148319.py:37: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  sns.heatmap(encoded.corr(),annot=True,cmap='coolwarm',fmt='.2f')
```




```
Accuracy (Classification): 1.0
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      3581
           1       1.00      1.00      1.00        69

    accuracy                           1.00      3650
   macro avg       1.00      1.00      1.00      3650
weighted avg       1.00      1.00      1.00      3650

Mean Absolute Error (Regression): 0.08328536986301371
```

In [ ]:

```
Avocado Dataset Exploration and Modeling Documentation

1. Goal
The primary goal of this script is to analyze the Avocado dataset,
extracting meaningful insights into avocado prices
and forecasting mean prices.using machine learning models.

2. Libraries
Pandas: Used for data manipulation and analysis.
Matplotlib and Seaborn: Employed for data visualization.
Scikit-learn: Utilized for machine learning tasks, including classification and regression.
3. Data Loading and Exploration
The script begins by loading the Avocado dataset from a web URL using Pandas.
Initial exploration involves displaying the first few rows,
summary statistics, and the column names to understand the dataset's structure and content.

4. Data Visualization
4.1 Countplot of Regions
A countplot is created to visualize the distribution of avocado prices across different regions.
This helps identify patterns and variations in avocado prices based on geographical locations.

4.2 Histogram of Average Price
A histogram is generated to display the distribution of average prices of avocados.
This visualization provides insights into the central tendency and spread of avocado prices.

5. Data Preprocessing
5.1 Label Encoding and One-Hot Encoding
Categorical variables, such as 'type' and 'region,' are encoded using label encoding
and one-hot encoding techniques. This sures suitability for machine learning algorithms

5.2 Correlation Heatmap
A heatmap of correlation is created to visualize the relationships between different features in the dataset.
This helps find potential dependencies between variables.

6. Classification Task - Predict Region (Orlando)
A Random Forest Classifier is employed to predict whether avocados are from the Orlando region.
The model is trained, and its accuracy is evaluated. A classification report provides detailed metrics on model performance.

7. Regression Task - Predict Average Price
A Random Forest Regressor is used to predict the average price of avocados.
The script prepares the data for regression, including one-hot encoding
and train-test splitting. The model is trained, and its performance is assessed using the Mean Absolute Error (MAE).

8. Conclusion
The script offers a comprehensive analysis of the Avocado dataset,
combining exploratory data analysis, visualization, and machine learning tasks.
Theoretical documentation provides an understanding of each step,facilitating access for users seeking insights from the dataset.
```