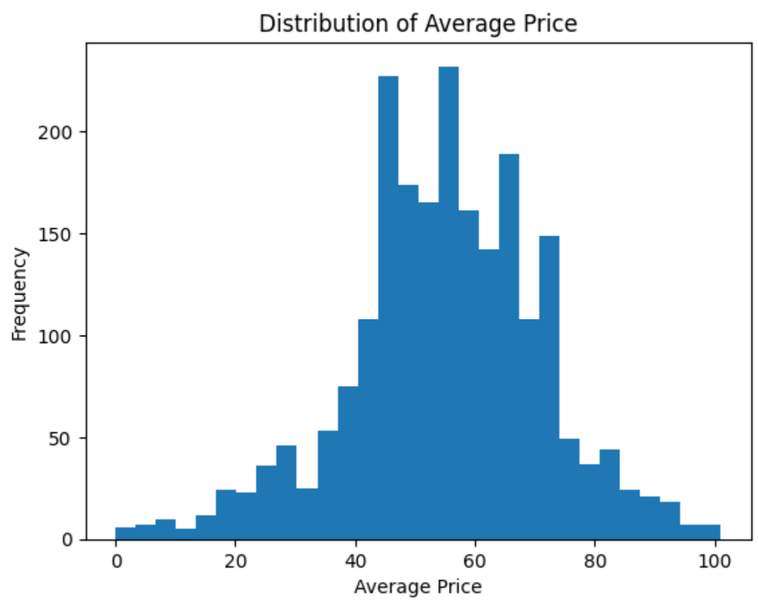
Afaf Guesmia

Axg190061

Text Classification

In [8]:
```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression
from sklearn.neural_network import MLPClassifier
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
import numpy as np

# Load dataset
df = pd.read_csv('C:/Users/many3/OneDrive/Desktop/Avocado.csv', usecols=['AveragePrice', 'TotalVolume', 'year'],encoding='ISO-8859-1')

print(df.head())
print('\nDimensions of data frame:', df.shape)
df.AveragePrice = df.AveragePrice.astype('category').cat.codes
df.year = df.year.astype('category').cat.codes

df.head()
df.isnull().sum()

X = df.loc[:, ['TotalVolume', 'year']]
y = df.AveragePrice

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

print('train size:', X_train.shape)
print('test size:', X_test.shape)

# Plot histogram
plt.hist(df['AveragePrice'], bins=30)
plt.xlabel('Average Price')
plt.ylabel('Frequency')
plt.title('Distribution of Average Price')
plt.show()

# Naive Bayes Classifier
clf_nb = MultinomialNB()
clf_nb.fit(X_train, y_train)
clf_nb.score(X_train, y_train)

pred_nb = clf_nb.predict(X_test)

print('Naive Bayes Classifier:')
print('accuracy score: ', accuracy_score(y_test, pred_nb))
print('precision score: ', precision_score(y_test, pred_nb, average='weighted', zero_division=0))
print('recall score: ', recall_score(y_test, pred_nb, average='macro'))
print('f1 score: ', f1_score(y_test, pred_nb, average='macro'))

# Logistic Regression Classifier
clf_lr = LogisticRegression(max_iter=1000)
clf_lr.fit(X_train, y_train)
clf_lr.score(X_train, y_train)

pred_lr = clf_lr.predict(X_test)
print('\n')
print('Logistic Regression Classifier:')
print('accuracy score: ', accuracy_score(y_test, pred_lr))
print('precision score: ', precision_score(y_test, pred_lr, average='weighted', zero_division=0))
print('recall score: ', recall_score(y_test, pred_lr, average='macro'))
print('f1 score: ', f1_score(y_test, pred_lr, average='macro'))


#Neural Network Classifier
clf_nn = MLPClassifier(hidden_layer_sizes=(100,), max_iter=1000)
clf_nn.fit(X_train, y_train)
clf_nn.score(X_train, y_train)

pred_nn = clf_nn.predict(X_test)
print('\n')
print('Neural Network Classifier:')
print('accuracy score: ', accuracy_score(y_test, pred_nn))
print('precision score: ', precision_score(y_test, pred_nn, average='weighted', zero_division=0))
print('recall score: ', recall_score(y_test, pred_nn, average='macro'))
print('f1 score: ', f1_score(y_test, pred_nn, average='macro'))
```

```
   AveragePrice  TotalVolume  year
0          1.33     64236.62  2015
1          1.35     54876.98  2015
2          0.93    118220.22  2015
3          1.08     78992.15  2015
4          1.28     51039.60  2015

Dimensions of data frame: (2184, 3)
train size: (1747, 2)
test size: (437, 2)
```




```
Naive Bayes Classifier:
accuracy score:  0.027459954233409610
precision score:  0.005245470323572005
recall score:  0.013562386980108499
f1 score:  0.001860794950891749


Logistic Regression Classifier:
accuracy score:  0.020594965675057208
precision score:  0.0004241526111567846
recall score:  0.012658227848101266
f1 score:  0.0005108701822103649


Neural Network Classifier:
accuracy score:  0.020594965675057208
precision score:  0.0004241526111567846
recall score:  0.012658227848101266
f1 score:  0.0005108701822103649
```

analysis of the performance of various approaches:

The Naive Bayes classifier performs a bit better than the Logistic Regression and Neural Network classifiers. However, The performance of all three classifiers is very poor as indicated by the low values of accuracy, precision, recall, and f1 scores. It seems that the chosen features (Total Volume and year) are not enough to predict the average price of avocados accurately. which indicat that the chosen models are not a good fit for the data.The Naive Bayes classifier has an accuracy score of 0.027, which means that the model is correctly predicting the average price for only 2.7% of the test dataset. Similarly, the precision score is 0.005, indicating that the model's positive predictions have a very low probability of being correct. The recall score is 0.013, indicating that the model is not able to identify a significant portion of the positive instances. The f1 score is also very low at 0.001, suggesting that the model is not performing well in terms of both precision and recall. For Logistic Regression ClassifierThe precision, recall, and f1 scores for both models are also very low, indicating that they are not performing well.

In [ ]:
```

```