```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
from google.colab import files
uploaded = files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving merged_data_cleaned.csv to merged_data_cleaned (1).csv
```

```
import pandas as pd
import io
data = pd.read_csv(io.BytesIO(uploaded['merged_data_cleaned.csv']))
data.head()
```

	Unnamed: 0	Species	Owner	Country.of.Origin	Farm.Name	Lot.Number	Mill	ICO.Number	Company	Altitude	Regior
0	0	Arabica	metad plc	Ethiopia	metad plc	NaN	metad plc	2014/2015	metad agricultural developmet plc	1950-2200	guji-hambela

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1339 entries, 0 to 1338
Data columns (total 44 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Unnamed: 0            1339 non-null   int64  
 1   Species               1339 non-null   object 
 2   Owner                 1332 non-null   object 
 3   Country.of.Origin     1338 non-null   object 
 4   Farm.Name             980 non-null    object 
 5   Lot.Number            276 non-null    object 
 6   Mill                  1021 non-null   object 
 7   ICO.Number            1182 non-null   object 
 8   Company               1130 non-null   object 
 9   Altitude              1113 non-null   object 
 10  Region                1280 non-null   object 
 11  Producer              1107 non-null   object 
 12  Number.of.Bags        1339 non-null   int64  
 13  Bag.Weight            1339 non-null   object 
 14  In.Country.Partner    1339 non-null   object 
 15  Harvest.Year          1292 non-null   object 
 16  Grading.Date          1339 non-null   object 
 17  Owner.1               1332 non-null   object 
 18  Variety               1113 non-null   object 
 19  Processing.Method     1169 non-null   object 
 20  Aroma                 1339 non-null   float64
 21  Flavor                1339 non-null   float64
 22  Aftertaste            1339 non-null   float64
 23  Acidity               1339 non-null   float64
 24  Body                  1339 non-null   float64
 25  Balance               1339 non-null   float64
 26  Uniformity             1339 non-null   float64
 27  Clean.Cup              1339 non-null   float64
 28  Sweetness              1339 non-null   float64
 29  Cupper.Points          1339 non-null   float64
 30  Total.Cup.Points       1339 non-null   float64
 31  Moisture               1339 non-null   float64
 32  Category.One.Defects   1339 non-null   int64
 33  Quakers                1338 non-null   float64
 34  Color                  1121 non-null   object
 35  Category.Two.Defects   1339 non-null   int64
 36  Expiration             1339 non-null   object
 37  Certification.Body     1339 non-null   object
 38  Certification.Address  1339 non-null   object
 39  Certification.Contact  1339 non-null   object
 40  unit_of_measurement    1339 non-null   object
 41  altitude_low_meters    1109 non-null   float64
 42  altitude_high_meters   1109 non-null   float64
 43  altitude_mean_meters   1109 non-null   float64
dtypes: float64(16), int64(4), object(24)
memory usage: 460.4+ KB
```

```
data['Unnamed: 0'].value_counts()
```

```
1338    1
439     1
441     1
442     1
443     1
       ..
893     1
894     1
895     1
896     1
0       1
Name: Unnamed: 0, Length: 1339, dtype: int64
```

```
from sklearn.preprocessing import LabelEncoder , OneHotEncoder
```

```
le = LabelEncoder()
```

```
data['Species'] = le.fit_transform(data['Species'])
data['Species'].value_counts()
```

```
0    1311
1      28
Name: Species, dtype: int64
```

```
le.classes_
```

```
array(['Arabica', 'Robusta'], dtype=object)
```

```
data['Owner'].value_counts()
```

```
juan luis alvarado romero           155
racafe & cia s.c.a                   60
exportadora de cafe condor s.a       54
kona pacific farmers cooperative     52
ipanema coffees                      50
                                   ... 
yasmin cofffee plantation plc         1
gregorio sebba                        1
semiramis casas velazquez             1
virginia gordillo gordillo            1
hector gabriel barreda nader          1
Name: Owner, Length: 315, dtype: int64
```

One hot encoder

```
one_hot = OneHotEncoder()
transformed_data = one_hot.fit_transform(data['Species'].values.reshape(-1,1)).toarray()
one_hot.categories_
```

```
[array([0, 1])]
```

```
transformed_data = pd.DataFrame(transformed_data ,
                                columns = ['0','1'])
```

```
transformed_data.head()
```

	0	1
0	1.0	0.0
1	1.0	0.0
2	1.0	0.0
3	1.0	0.0
4	1.0	0.0

```
transformed_data.iloc[90 , ]
```

```
0    1.0
1    0.0
Name: 90, dtype: float64
```

```
data['Species'][90]
```

```
0
```

Normalization & standard deviation

```
# This is formatted as code
```

```
numeric_columns = [c for c in data.columns if data[c].dtype != np.dtype('O')]
len(numeric_columns) , len(data.columns)
```

```
(21, 44)
```

```
numeric_columns
```

```
['Unnamed: 0',
 'Species',
 'Number.of.Bags',
 'Aroma',
 'Flavor',
 'Aftertaste',
 'Acidity',
 'Body',
 'Balance',
 'Uniformity',
 'Clean.Cup',
 'Sweetness',
 'Cupper.Points',
 'Total.Cup.Points',
 'Moisture',
 'Category.One.Defects',
 'Quakers',
 'Category.Two.Defects',
 'altitude_low_meters',
 'altitude_high_meters',
 'altitude_mean_meters']
```

```
numeric_columns.remove('altitude_high_meters')
print(numeric_columns)
numeric_columns.remove('Clean.Cup')
print(numeric_columns)
```

```
['Unnamed: 0', 'Species', 'Number.of.Bags', 'Aroma', 'Flavor', 'Aftertaste', 'Acidity', 'Body', 'Balance', 'Uniformity', 'Clean
['Unnamed: 0', 'Species', 'Number.of.Bags', 'Aroma', 'Flavor', 'Aftertaste', 'Acidity', 'Body', 'Balance', 'Uniformity', 'Sweet
```

```
temp_data = data[numeric_columns]
temp_data
```

	Unnamed: 0	Species	Number.of.Bags	Aroma	Flavor	Aftertaste	Acidity	Body	Balance	Uniformity	Sweetness	Cupper.Points
0	0	0	300	8.67	8.83	8.67	8.75	8.50	8.42	10.00	10.00	8.75
1	1	0	300	8.75	8.67	8.50	8.58	8.42	8.42	10.00	10.00	8.58
2	2	0	5	8.42	8.50	8.42	8.42	8.33	8.42	10.00	10.00	9.25
3	3	0	320	8.17	8.58	8.42	8.42	8.50	8.25	10.00	10.00	8.67
4	4	0	300	8.25	8.50	8.25	8.50	8.42	8.33	10.00	10.00	8.58
...	...	...	...	...	...	...	...	...	...	...	...	..
1334	1334	1	1	7.75	7.58	7.33	7.58	5.08	7.83	10.00	7.75	7.83
1335	1335	1	1	7.50	7.67	7.75	7.75	5.17	5.25	10.00	8.42	8.58
1336	1336	1	1	7.33	7.33	7.17	7.42	7.50	7.17	9.33	7.42	7.17
1337	1337	1	1	7.42	6.83	6.75	7.17	7.25	7.00	9.33	7.08	6.92
1338	1338	1	1	6.75	6.67	6.50	6.83	6.92	6.83	9.33	6.67	7.92

Normalization

```
from sklearn.preprocessing import StandardScaler , MinMaxScaler
```

```
import warnings
warnings.filterwarnings('ignore')
```

```
normalizer = MinMaxScaler()
temp_data.dropna(axis = 1 , inplace = True)
normalized_data = normalizer.fit_transform(temp_data)
pd.DataFrame(normalized_data , columns = temp_data.columns)
```

	Unnamed: 0	Species	Number.of.Bags	Aroma	Flavor	Aftertaste	Acidity	Body	Balance	Uniformity	Sweetness	Cupp
0	0.000000	0.0	0.282486	0.990857	1.000000	1.000000	1.000000	0.990676	0.962286	1.000	1.000	
1	0.000747	0.0	0.282486	1.000000	0.981880	0.980392	0.980571	0.981352	0.962286	1.000	1.000	
2	0.001495	0.0	0.004708	0.962286	0.962627	0.971165	0.962286	0.970862	0.962286	1.000	1.000	
3	0.002242	0.0	0.301318	0.933714	0.971687	0.971165	0.962286	0.990676	0.942857	1.000	1.000	
4	0.002990	0.0	0.282486	0.942857	0.962627	0.951557	0.971429	0.981352	0.952000	1.000	1.000	
...	...	...	...	...	...	...	...	...	...	...	...	
1334	0.997010	1.0	0.000942	0.885714	0.858437	0.845444	0.866286	0.592075	0.894857	1.000	0.775	
1335	0.997758	1.0	0.000942	0.857143	0.868630	0.893887	0.885714	0.602564	0.600000	1.000	0.842	
1336	0.998505	1.0	0.000942	0.837714	0.830125	0.826990	0.848000	0.874126	0.819429	0.933	0.742	
1337	0.999253	1.0	0.000942	0.848000	0.773499	0.778547	0.819429	0.844988	0.800000	0.933	0.708	
1338	1.000000	1.0	0.000942	0.771429	0.755379	0.749712	0.780571	0.806527	0.780571	0.933	0.667	

1339 rows × 16 columns

Standardization

```
standard_scaler = StandardScaler()
standardized_data = standard_scaler.fit_transform(temp_data)
pd.DataFrame(standardized_data , columns = temp_data.columns)
```

	Unnamed: 0	Species	Number.of.Bags	Aroma	Flavor	Aftertaste	Acidity	Body	Balance	Uniformity	Sweetness
0	-1.730758	-0.146143	1.122199	2.923259	3.287965	3.138457	3.198164	2.655944	2.206476	0.29785	0.232692
1	-1.728171	-0.146143	1.122199	3.135225	2.886251	2.717990	2.750424	2.439684	2.206476	0.29785	0.232692
2	-1.725584	-0.146143	-1.148103	2.260865	2.459430	2.520123	2.329022	2.196392	2.206476	0.29785	0.232692
3	-1.722997	-0.146143	1.276118	1.598472	2.660287	2.520123	2.329022	2.655944	1.790615	0.29785	0.232692
4	-1.720409	-0.146143	1.122199	1.810438	2.459430	2.099656	2.539723	2.439684	1.986314	0.29785	0.232692
...	...	...	...	...	...	...	...	...	...	...	...
1334	1.720409	6.842619	-1.178887	0.485650	0.149574	-0.175812	0.116661	-6.589155	0.763194	0.29785	-3.420665
1335	1.722997	6.842619	-1.178887	-0.176744	0.375538	0.862989	0.564400	-6.345863	-5.548106	0.29785	-2.332777

Handling missing values

```
data.isnull().sum()
```

```
Unnamed: 0                 0
Species                    0
Owner                      7
Country.of.Origin          1
Farm.Name                359
Lot.Number              1063
Mill                     318
ICO.Number               157
Company                  209
Altitude                 226
Region                    59
Producer                 232
Number.of.Bags             0
Bag.Weight                 0
In.Country.Partner         0
Harvest.Year              47
Grading.Date               0
Owner.1                    7
```

```
Variety                 226
Processing.Method       170
Aroma                     0
Flavor                    0
Aftertaste                0
Acidity                   0
Body                      0
Balance                   0
Uniformity                0
Clean.Cup                 0
Sweetness                 0
Cupper.Points             0
Total.Cup.Points          0
Moisture                  0
Category.One.Defects      0
Quakers                   1
Color                   218
Category.Two.Defects      0
Expiration                0
Certification.Body        0
Certification.Address     0
Certification.Contact     0
unit_of_measurement       0
altitude_low_meters     230
altitude_high_meters    230
altitude_mean_meters    230
dtype: int64
```

```
data['Quakers'].isnull().sum()
```

```
1
```

Simple Imputer

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values=np.nan , strategy='mean')
lotnum_col = imputer.fit_transform(data['Quakers'].values.reshape(-1,1))
pd.DataFrame(lotnum_col).isnull().sum()
```

```
0    0
dtype: int64
```

```
data['Quakers'].isnull().sum()
```

```
1
```

Discretization

```
from sklearn.preprocessing import KBinsDiscretizer
temp_data.head()
```

	Unnamed: 0	Species	Number.of.Bags	Aroma	Flavor	Aftertaste	Acidity	Body	Balance	Uniformity	Sweetness	Cupper.Points	T
0	0	0	300	8.67	8.83	8.67	8.75	8.50	8.42	10.0	10.0	8.75	
1	1	0	300	8.75	8.67	8.50	8.58	8.42	8.42	10.0	10.0	8.58	
2	2	0	5	8.42	8.50	8.42	8.42	8.33	8.42	10.0	10.0	9.25	
3	3	0	320	8.17	8.58	8.42	8.42	8.50	8.25	10.0	10.0	8.67	
4	4	0	300	8.25	8.50	8.25	8.50	8.42	8.33	10.0	10.0	8.58	

Quantile Discretization Transform

```
trans = KBinsDiscretizer(n_bins =10 , encode = 'ordinal' , strategy='quantile')
new_data = trans.fit_transform(temp_data)
pd.DataFrame(new_data,columns = temp_data.columns )
```

	Unnamed: 0	Species	Number.of.Bags	Aroma	Flavor	Aftertaste	Acidity	Body	Balance	Uniformity	Sweetness	Cupper.Points
0	0.0	0.0	8.0	9.0	8.0	9.0	8.0	8.0	8.0	1.0	0.0	9.0
1	0.0	0.0	8.0	9.0	8.0	9.0	8.0	8.0	8.0	1.0	0.0	9.0
2	0.0	0.0	1.0	9.0	8.0	9.0	8.0	8.0	8.0	1.0	0.0	9.0
3	0.0	0.0	8.0	9.0	8.0	9.0	8.0	8.0	8.0	1.0	0.0	9.0
4	0.0	0.0	8.0	9.0	8.0	9.0	8.0	8.0	8.0	1.0	0.0	9.0
...	...	...	...	...	...	...	...	...	...	...	...	..
1334	9.0	0.0	0.0	7.0	5.0	4.0	5.0	0.0	7.0	1.0	0.0	8.0
1335	9.0	0.0	0.0	4.0	6.0	8.0	7.0	0.0	0.0	1.0	0.0	9.0
1336	9.0	0.0	0.0	2.0	2.0	2.0	3.0	4.0	1.0	1.0	0.0	1.0
1337	9.0	0.0	0.0	3.0	0.0	0.0	1.0	1.0	0.0	1.0	0.0	0.0
1338	9.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	8.0

Uniform Discretization Transform

```
trans = KBinsDiscretizer(n_bins =10 , encode = 'ordinal' , strategy='uniform')
new_data = trans.fit_transform(temp_data)

pd.DataFrame(new_data,columns = temp_data.columns )
```

```
      Unnamed:  Species  Number.of.Bags  Aroma  Flavor  Aftertaste  Acidity  Body  Balance  Uniformity  Sweetness  Cupper.Points
             0
   0       0.0      0.0             2.0    9.0     9.0         9.0      9.0   9.0      9.0         9.0        9.0            8.0
   1       0.0      0.0             2.0    9.0     9.0         9.0      9.0   9.0      9.0         9.0        9.0            8.0
   2       0.0      0.0             0.0    9.0     9.0         9.0      9.0   9.0      9.0         9.0        9.0            9.0
   3       0.0      0.0             3.0    9.0     9.0         9.0      9.0   9.0      9.0         9.0        9.0            8.0
   4       0.0      0.0             2.0    9.0     9.0         9.0      9.0   9.0      9.0         9.0        9.0            8.0
 ...       ...      ...             ...    ...     ...         ...      ...   ...      ...         ...        ...             ..
1334       9.0      9.0             0.0    8.0     8.0         8.0      8.0   5.0      8.0         9.0        7.0            7.0
1335       9.0      9.0             0.0    8.0     8.0         8.0      8.0   6.0      6.0         9.0        8.0            8.0
```

KMeans Discretization Transform

```
trans = KBinsDiscretizer(n_bins =10 , encode = 'ordinal' , strategy='kmeans')
new_data = trans.fit_transform(temp_data)

pd.DataFrame(new_data,columns = temp_data.columns )
```

	Unnamed: 0	Species	Number.of.Bags	Aroma	Flavor	Aftertaste	Acidity	Body	Balance	Uniformity	Sweetness	Cupper.Points
0	0.0	0.0	6.0	9.0	8.0	9.0	9.0	8.0	9.0	6.0	7.0	8.0
1	0.0	0.0	6.0	9.0	8.0	8.0	8.0	8.0	9.0	6.0	7.0	8.0
2	0.0	0.0	1.0	9.0	8.0	8.0	8.0	8.0	9.0	6.0	7.0	9.0
3	0.0	0.0	6.0	9.0	8.0	8.0	8.0	8.0	9.0	6.0	7.0	8.0
4	0.0	0.0	6.0	9.0	8.0	7.0	8.0	8.0	9.0	6.0	7.0	8.0
...	...	...	...	...	...	...	...	...	...	...	...	..
1334	9.0	1.0	0.0	8.0	5.0	4.0	5.0	1.0	9.0	6.0	4.0	6.0
1335	9.0	1.0	0.0	8.0	5.0	5.0	5.0	2.0	1.0	6.0	5.0	8.0
1336	9.0	1.0	0.0	7.0	4.0	3.0	5.0	5.0	7.0	5.0	4.0	4.0
1337	9.0	1.0	0.0	7.0	3.0	2.0	4.0	5.0	5.0	5.0	3.0	3.0
1338	9.0	1.0	0.0	3.0	2.0	1.0	3.0	4.0	3.0	5.0	3.0	6.0

1339 rows × 16 columns

✓ 0s completed at 22:16