Champo Carpet Assignment

June 23, 2024

1 Aaron Vo

[1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.metrics import silhouette_score

from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix,␣
 ↪accuracy_score
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
```

[2]:
```
df_raw_order = pd.read_excel("champoo.xlsx", 'Raw Data-Order and Sample')
df_sample_order = pd.read_excel("champoo.xlsx", 'Data on Sample ONLY')
```

```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/openpyxl/worksheet/_reader.py:329: UserWarning: Unknown extension is
not supported and will be removed
  warn(msg)
```

[3]:
```
df_raw_order.head()
```

[3]:

	OrderType	OrderCategory	CustomerCode	CountryName	CustomerOrderNo	\
0	Area Wise	Order	H-1	USA	1873354	
1	Area Wise	Order	H-1	USA	1873354	
2	Area Wise	Order	H-1	USA	1873354	
3	Area Wise	Order	H-1	USA	1918436	
4	Area Wise	Order	H-1	USA	1873354	

	Custorderdate	UnitName	QtyRequired	TotalArea	Amount	ITEM_NAME	\

```
0    2017-01-16       Ft            2       6.00   12.00  HAND TUFTED
1    2017-01-16       Ft            2       9.00   18.00  HAND TUFTED
2    2017-01-16       Ft            2      54.00  108.00  HAND TUFTED
3    2017-02-01       Ft            5      54.00  270.00  HAND TUFTED
4    2017-01-16       Ft            5      71.25  356.25  HAND TUFTED

             QualityName         DesignName ColorName ShapeName  Unnamed: 15  \
0  TUFTED 30C HARD TWIST  OLD LONDON [3715]     BEIGE       REC            1
1  TUFTED 30C HARD TWIST  OLD LONDON [3715]     BEIGE       REC            1
2  TUFTED 30C HARD TWIST  OLD LONDON [3715]     BEIGE       REC            1
3  TUFTED 30C HARD TWIST  OLD LONDON [3715]     BEIGE       REC            1
4  TUFTED 30C HARD TWIST  OLD LONDON [3715]     BEIGE       REC            1

   AreaFt
0    6.00
1    9.00
2   54.00
3   54.00
4   71.25
```

[6]:
```
df_sample_order.head()
```

[6]:
```
  CustomerCode CountryName  USA   UK  Italy  Belgium  Romania  Australia  \
0           CC       INDIA  0.0  0.0    0.0      0.0      0.0        0.0
1         M-1          USA  1.0  0.0    0.0      0.0      0.0        0.0
2         M-1          USA  1.0  0.0    0.0      0.0      0.0        0.0
3         M-1          USA  1.0  0.0    0.0      0.0      0.0        0.0
4         M-1          USA  1.0  0.0    0.0      0.0      0.0        0.0

   India  QtyRequired  … Knotted  Jacquard  Handloom  Other  ShapeName  REC  \
0    1.0            1  …       0         0         0      0        REC    1
1    0.0            1  …       0         0         0      0        REC    1
2    0.0            2  …       0         0         0      0        REC    1
3    0.0            1  …       0         0         0      0        REC    1
4    0.0            1  …       0         0         0      0        REC    1

   Round  Square  AreaFt Order Conversion
0      0       0    80.0                1
1      0       0    80.0                1
2      0       0    80.0                1
3      0       0    80.0                1
4      0       0    80.0                1

[5 rows x 25 columns]
```

[7]:
```
df_raw_order['QualityName'] = df_raw_order['QualityName'].astype(str)
df_raw_order['DesignName'] = df_raw_order['DesignName'].astype(str)
```

```
df_raw_order['ColorName'] = df_raw_order['ColorName'].astype(str)
df_raw_order['ShapeName'] = df_raw_order['ShapeName'].astype(str)
df_raw_order['ITEM_NAME'] = df_raw_order['ITEM_NAME'].astype(str)
```

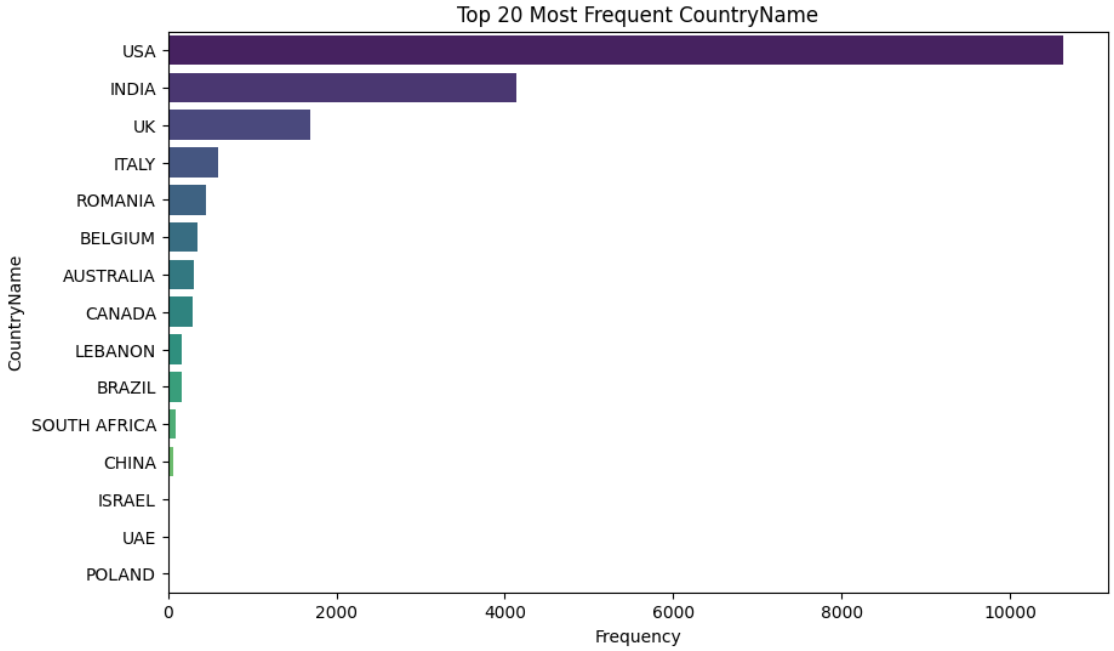
[8]:
```
# Group by the combination of QualityName, DesignName, ColorName, and ShapeName
classification_features = ['CountryName', 'ITEM_NAME','QualityName',␣
 ↪'DesignName', 'ColorName', 'ShapeName']
numeric_features = ['QtyRequired', 'TotalArea', 'Amount']

top_n = 20
for i in classification_features:
    print(f'Top {top_n} most in {i}')
    print(df_raw_order[i].value_counts().head(top_n))
    print('--------------')
```

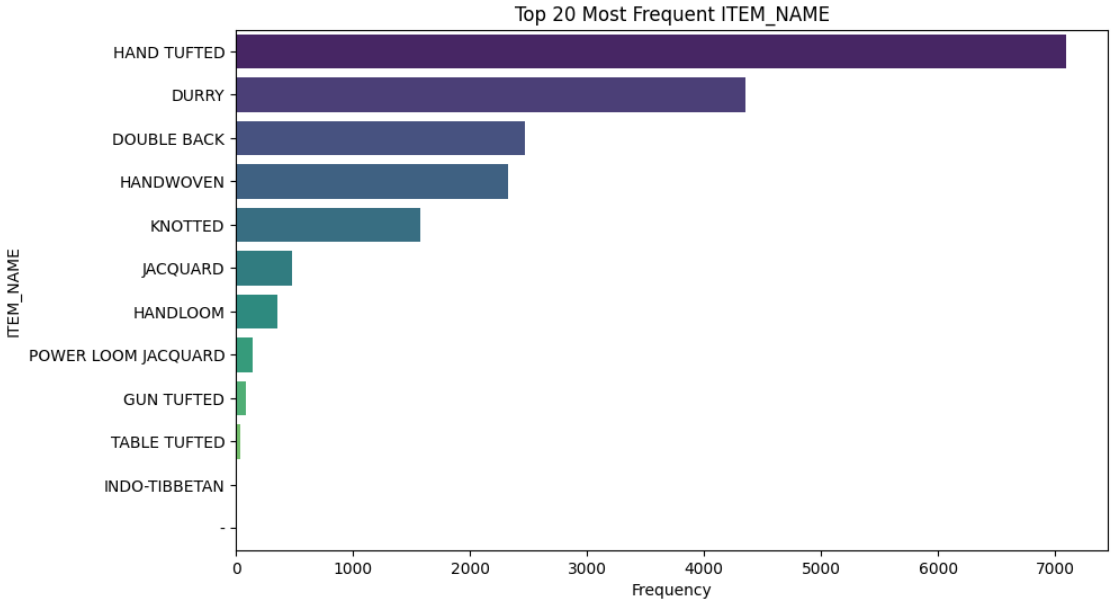
```
Top 20 most in CountryName
USA              10626
INDIA             4135
UK                1694
ITALY              596
ROMANIA            456
BELGIUM            346
AUSTRALIA          311
CANADA             287
LEBANON            168
BRAZIL             165
SOUTH AFRICA        94
CHINA               58
ISRAEL              12
UAE                  4
POLAND               3
Name: CountryName, dtype: int64
--------------
Top 20 most in ITEM_NAME
HAND TUFTED            7095
DURRY                  4355
DOUBLE BACK            2474
HANDWOVEN              2330
KNOTTED                1575
JACQUARD                477
HANDLOOM                357
POWER LOOM JACQUARD     144
GUN TUFTED               91
TABLE TUFTED             42
INDO-TIBBETAN            11
-                         4
Name: ITEM_NAME, dtype: int64
```

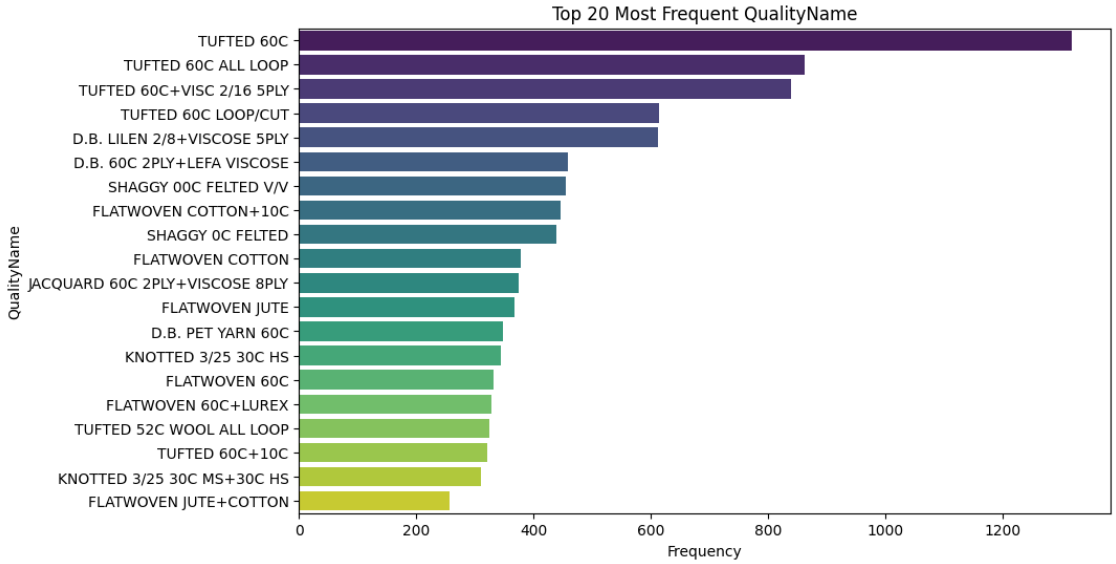
```
--------------
Top 20 most in QualityName
TUFTED 60C                          1319
TUFTED 60C ALL LOOP                  862
TUFTED 60C+VISC 2/16 5PLY            840
TUFTED 60C LOOP/CUT                  614
D.B. LILEN 2/8+VISCOSE 5PLY          613
D.B. 60C 2PLY+LEFA VISCOSE           459
SHAGGY 00C FELTED V/V                455
FLATWOVEN COTTON+10C                 447
SHAGGY 0C FELTED                     439
FLATWOVEN COTTON                     378
JACQUARD 60C 2PLY+VISCOSE 8PLY       374
FLATWOVEN JUTE                       367
D.B. PET YARN 60C                    348
KNOTTED 3/25 30C HS                  344
FLATWOVEN 60C                        332
FLATWOVEN 60C+LUREX                  329
TUFTED 52C WOOL ALL LOOP             324
TUFTED 60C+10C                       322
KNOTTED 3/25 30C MS+30C HS           310
FLATWOVEN JUTE+COTTON                257
Name: QualityName, dtype: int64
--------------
Top 20 most in DesignName
PLAIN                  819
HOMER                  459
TEXTURE LOOP           437
ELOQ GARDEN [8517]     350
MODASA                 236
SAMPLE                 225
DOUBLE DIAMOND         225
NAHLA                  212
KOTA                   201
PALI                   163
STELLA                 153
PP-45                  144
PET BRAID              139
MOROCCAN ZIGZAG        127
AMARI TRELLIS          125
DIAMOND                122
CHEVRON                122
OD DOUBLE DIAMOND      117
MASINISSA [TRIBAL]     116
NUMA                   110
Name: DesignName, dtype: int64
--------------
Top 20 most in ColorName
```

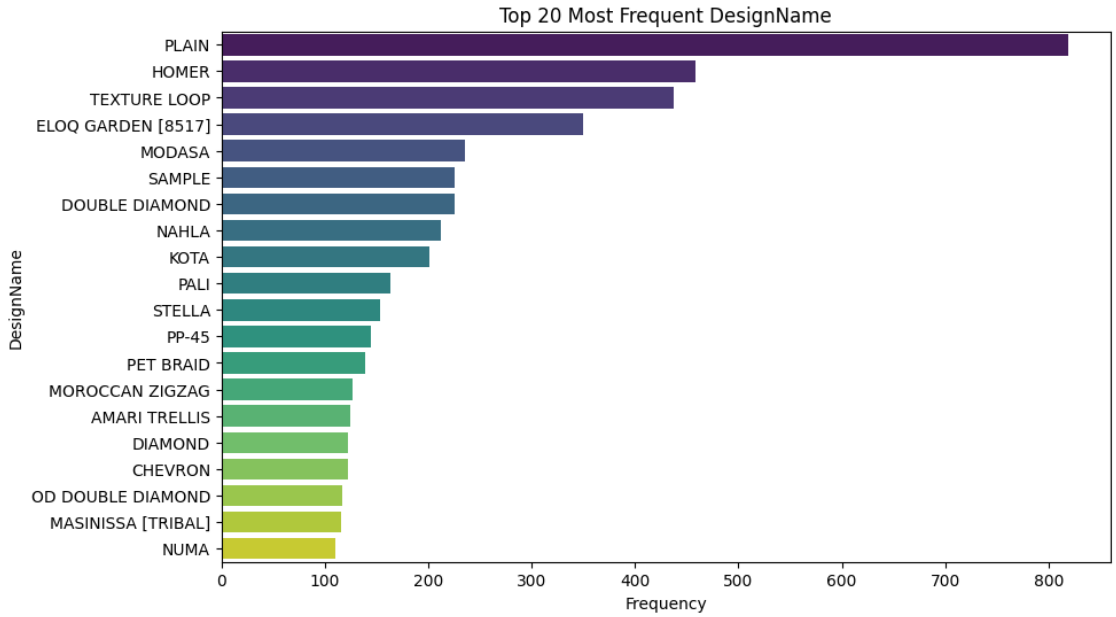
```
GREY         1334
MULTI        1254
BLUE         1014
SILVER        743
BEIGE         648
NAVY          580
BLACK         503
CHARCOAL      454
NATURAL       418
IVORY         371
INDIGO        240
NEUTRAL       233
CREAM         222
DENIM         216
PINK          216
TEAL          201
SAND          198
WHITE         168
RED           166
PETAL         154
Name: ColorName, dtype: int64
--------------
Top 20 most in ShapeName
REC        18518
ROUND        362
SQUARE        72
OCTAGON        2
OVAL           1
Name: ShapeName, dtype: int64
--------------
```

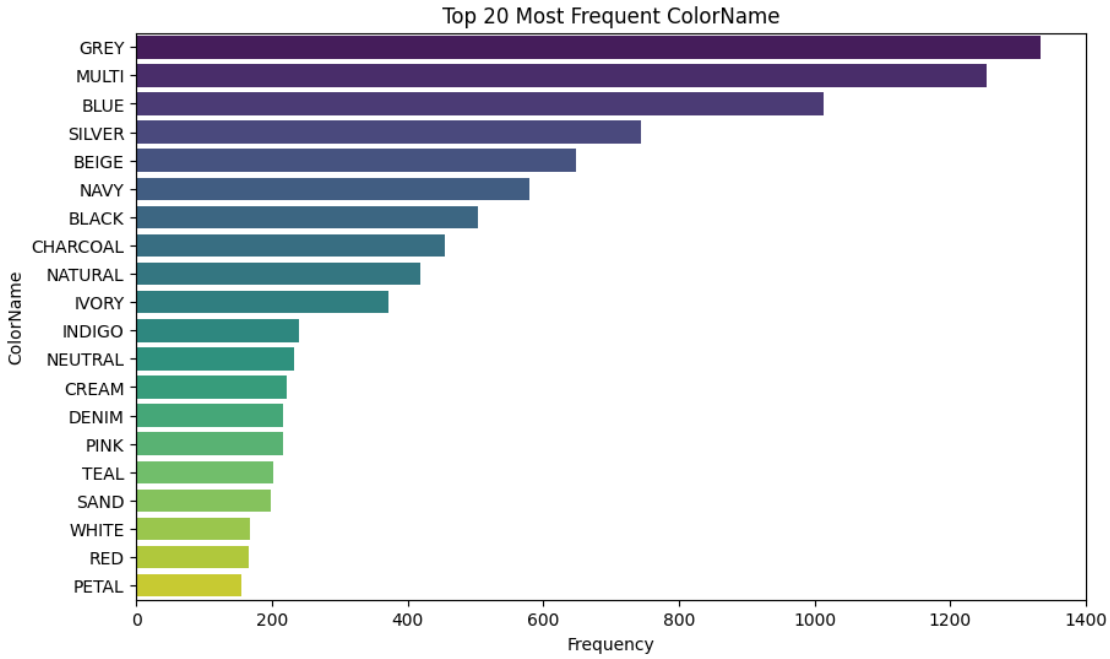
[9]:
```
for i in classification_features:
    # Calculate the frequency of each category
    quality_counts = df_raw_order[i].value_counts().head(top_n)
    plt.figure(figsize=(10, 6))
    sns.barplot(x=quality_counts.values, y=quality_counts.index,␣
 ↪palette='viridis')
    plt.title(f'Top {top_n} Most Frequent {i}')
    plt.xlabel('Frequency')
    plt.ylabel(i)
    plt.show()
```

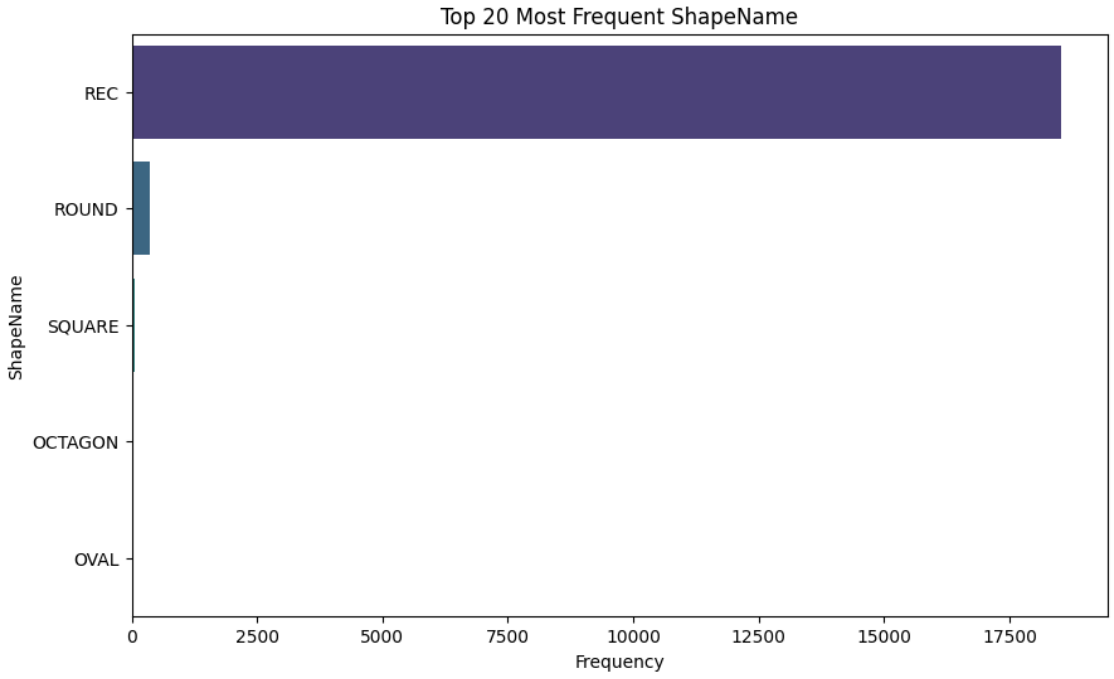

Top 20 Most Frequent CountryName
USA
INDIA
UK
ITALY
ROMANIA
BELGIUM
AUSTRALIA
CANADA
LEBANON
BRAZIL
SOUTH AFRICA
CHINA
ISRAEL
UAE
POLAND
CountryName
0
2000
4000
6000
8000
10000
Frequency


Top 20 Most Frequent ITEM_NAME
HAND TUFTED
DURRY
DOUBLE BACK
HANDWOVEN
KNOTTED
JACQUARD
HANDLOOM
POWER LOOM JACQUARD
GUN TUFTED
TABLE TUFTED
INDO-TIBBETAN
-
ITEM_NAME
0
1000
2000
3000
4000
5000
6000
7000
Frequency


Top 20 Most Frequent QualityName
TUFTED 60C
TUFTED 60C ALL LOOP
TUFTED 60C+VISC 2/16 5PLY
TUFTED 60C LOOP/CUT
D.B. LILEN 2/8+VISCOSE 5PLY
D.B. 60C 2PLY+LEFA VISCOSE
SHAGGY 00C FELTED V/V
FLATWOVEN COTTON+10C
SHAGGY 0C FELTED
FLATWOVEN COTTON
JACQUARD 60C 2PLY+VISCOSE 8PLY
FLATWOVEN JUTE
D.B. PET YARN 60C
KNOTTED 3/25 30C HS
FLATWOVEN 60C
FLATWOVEN 60C+LUREX
TUFTED 52C WOOL ALL LOOP
TUFTED 60C+10C
KNOTTED 3/25 30C MS+30C HS
FLATWOVEN JUTE+COTTON
QualityName
0
200
400
600
800
1000
1200
Frequency

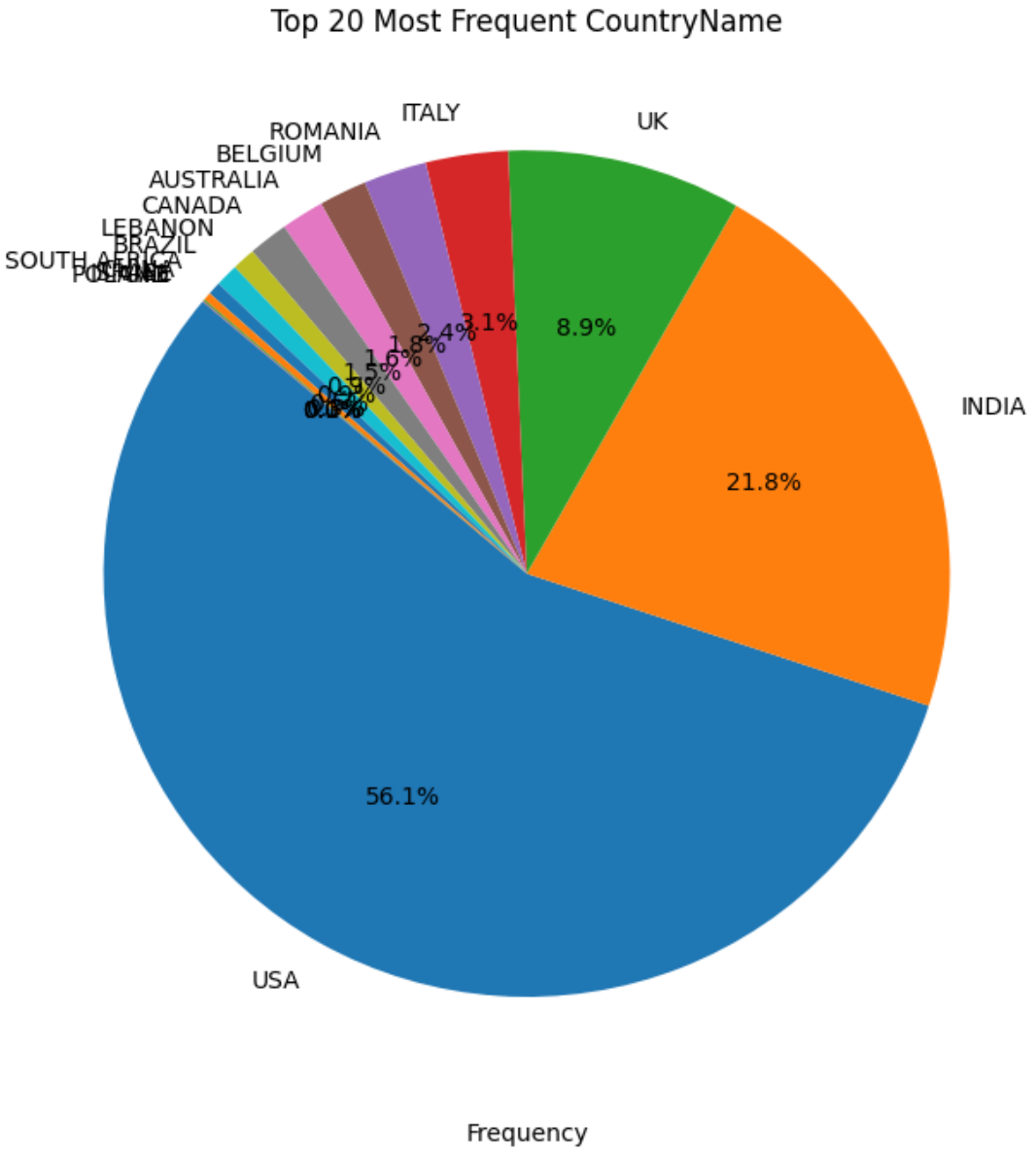

Top 20 Most Frequent DesignName
PLAIN
HOMER
TEXTURE LOOP
ELOQ GARDEN [8517]
MODASA
SAMPLE
DOUBLE DIAMOND
NAHLA
KOTA
PALI
STELLA
PP-45
PET BRAID
MOROCCAN ZIGZAG
AMARI TRELLIS
DIAMOND
CHEVRON
OD DOUBLE DIAMOND
MASINISSA [TRIBAL]
NUMA
DesignName
0
100
200
300
400
500
600
700
800
Frequency

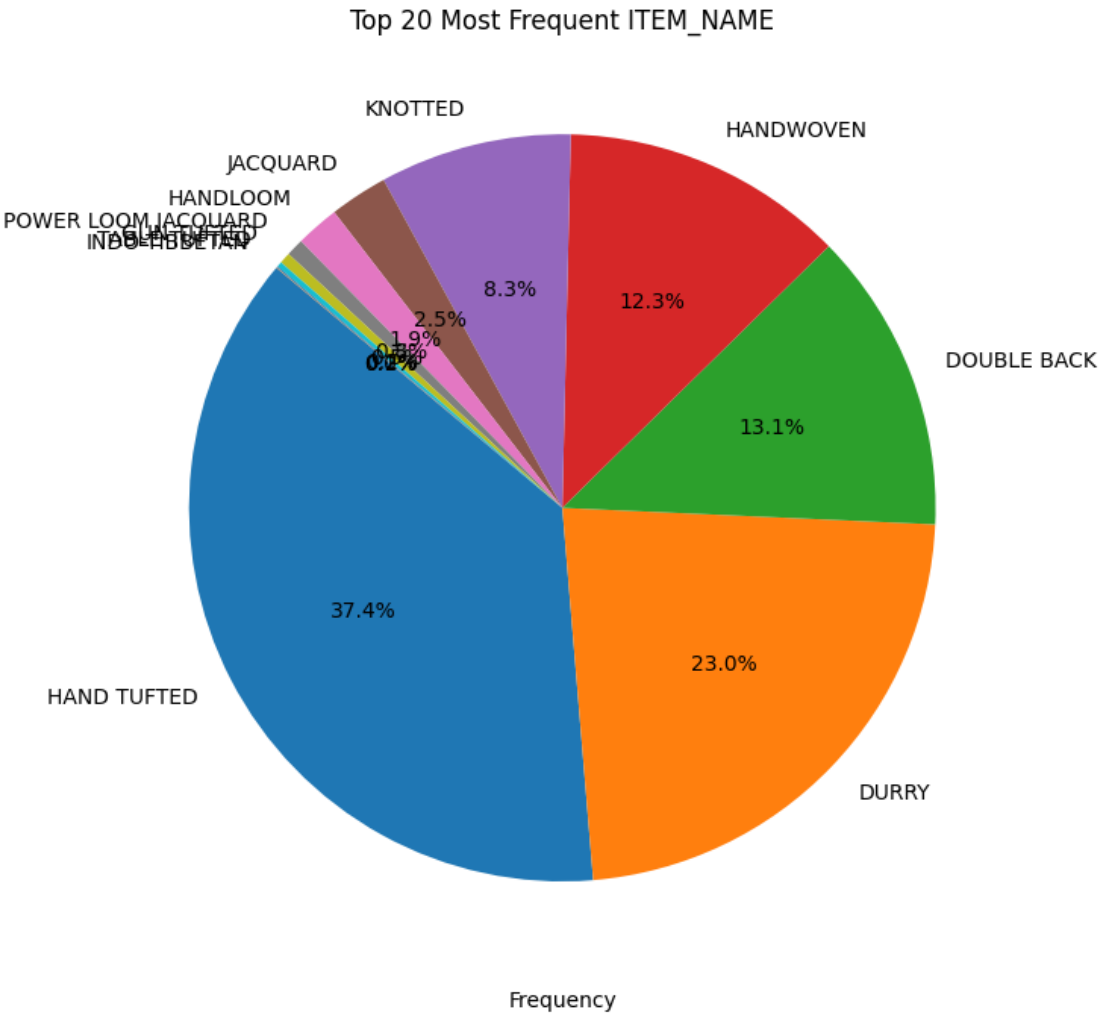


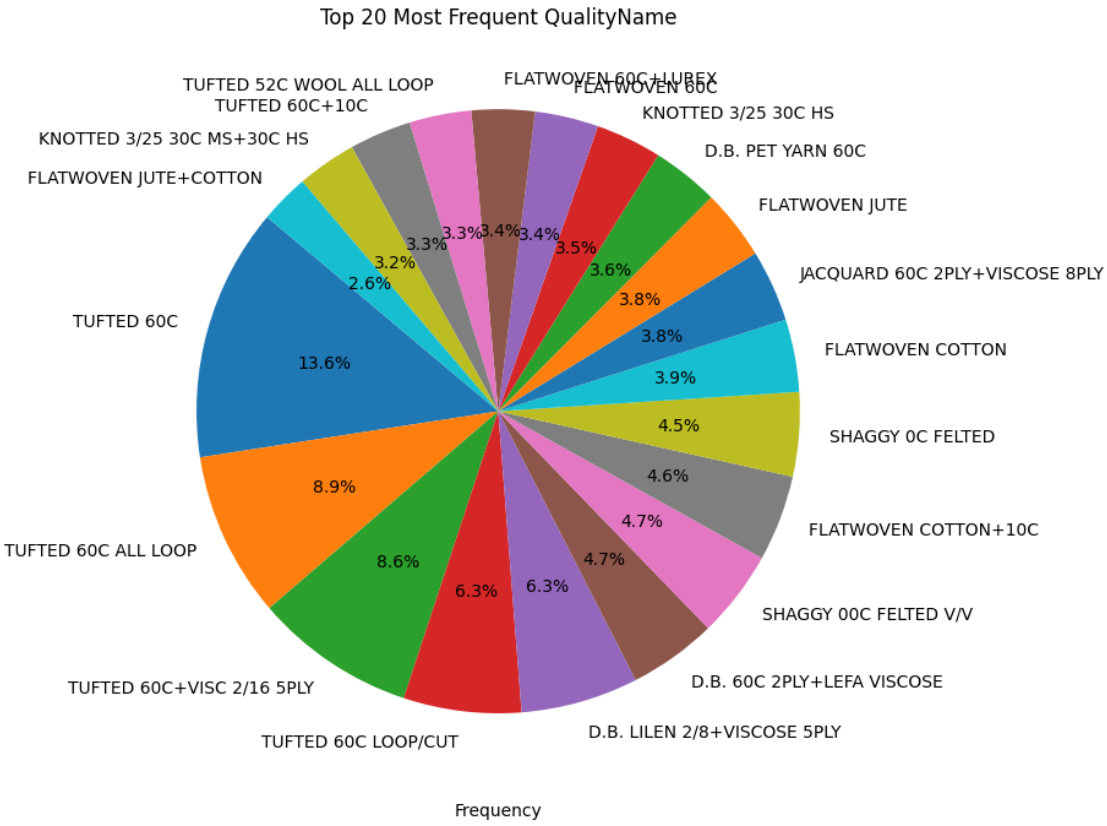


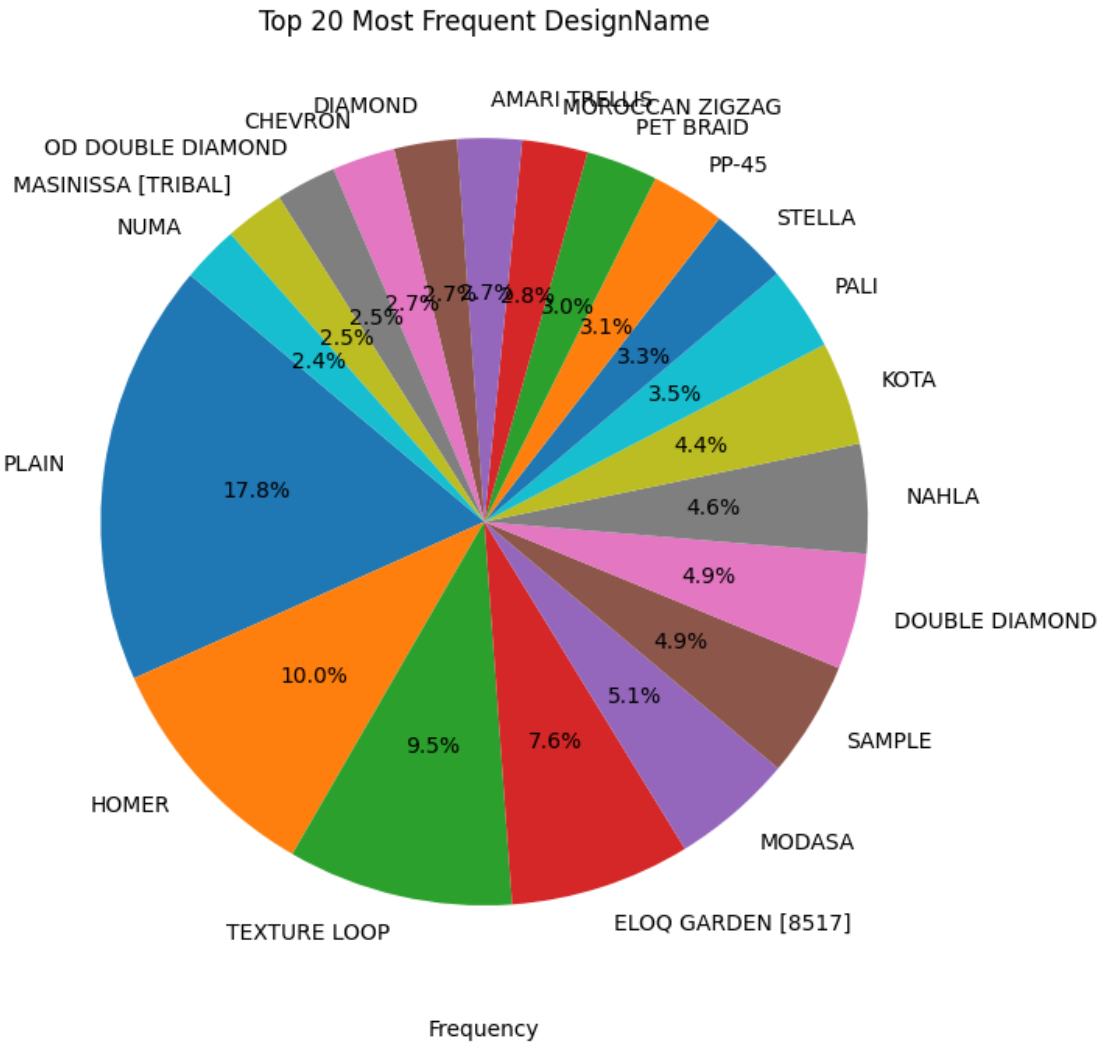
[10]:
```
for i in classification_features:
    plt.figure(figsize=(8, 8))
```

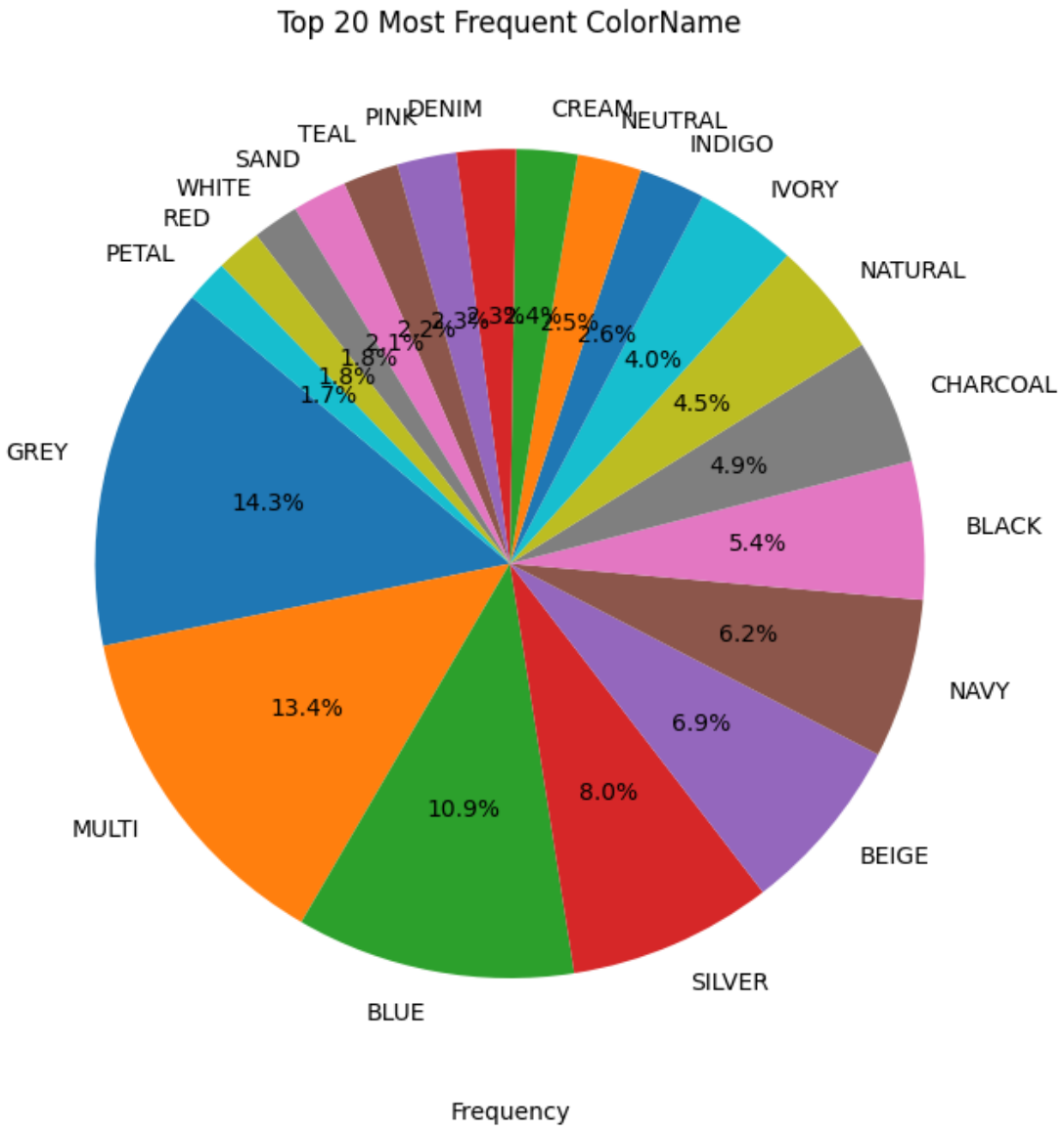
```
    df_raw_order[i].value_counts().head(top_n).plot.pie(autopct='%1.1f%%',␣
↪startangle=140)
    plt.title(f'Top {top_n} Most Frequent {i}')
    plt.xlabel('Frequency')
    plt.ylabel('')
    plt.show()
```

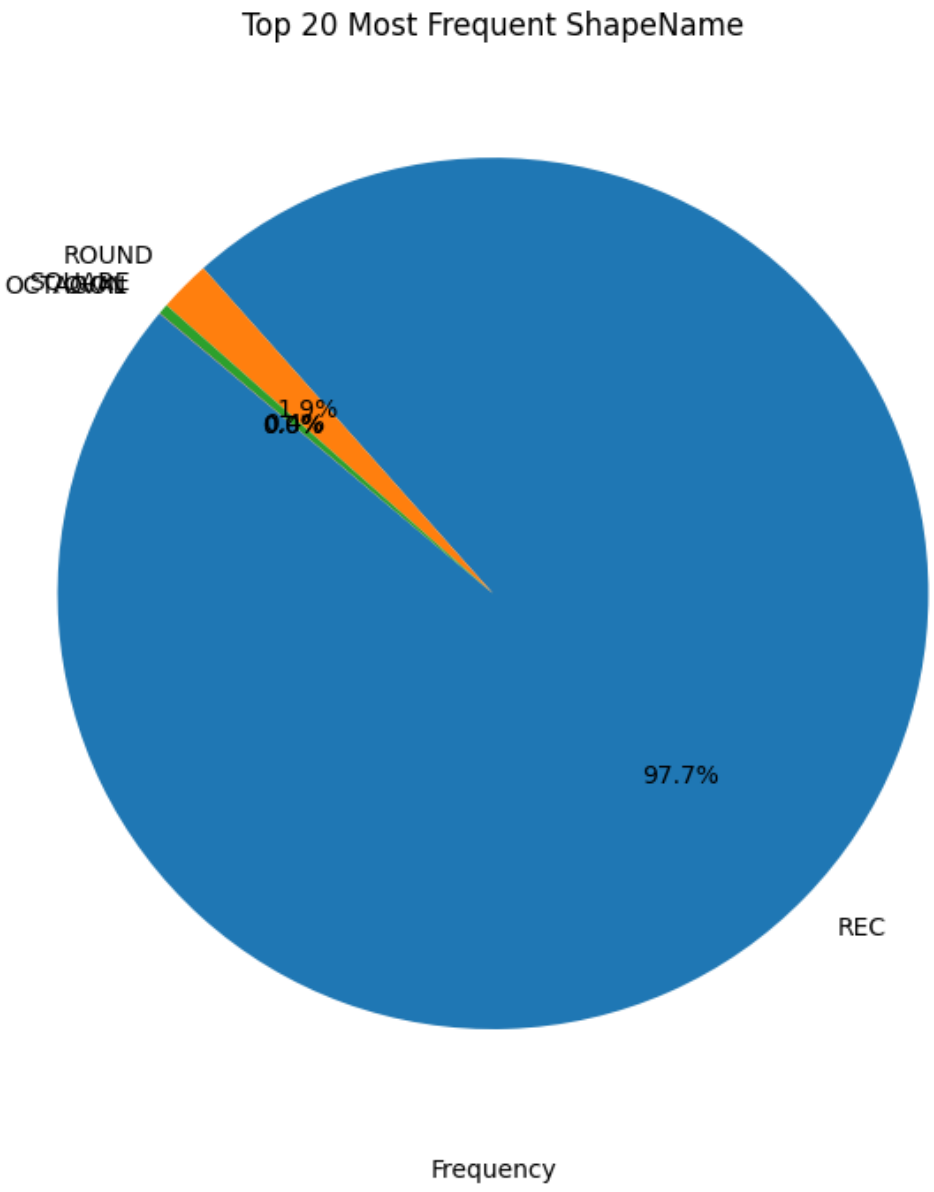



Top 20 Most Frequent ITEM_NAME
KNOTTED
HANDWOVEN
JACQUARD
HANDLOOM
POWER LOOM JACQUARD
8.3%
12.3%
2.5%
DOUBLE BACK
13.1%
37.4%
23.0%
HAND TUFTED
DURRY
Frequency


Top 20 Most Frequent QualityName
TUFTED 52C WOOL ALL LOOP
TUFTED 60C+10C
KNOTTED 3/25 30C MS+30C HS
FLATWOVEN JUTE+COTTON
TUFTED 60C
TUFTED 60C ALL LOOP
TUFTED 60C+VISC 2/16 5PLY
TUFTED 60C LOOP/CUT
D.B. LILEN 2/8+VISCOSE 5PLY
D.B. 60C 2PLY+LEFA VISCOSE
SHAGGY 00C FELTED V/V
FLATWOVEN COTTON+10C
SHAGGY 0C FELTED
FLATWOVEN COTTON
JACQUARD 60C 2PLY+VISCOSE 8PLY
FLATWOVEN JUTE
D.B. PET YARN 60C
KNOTTED 3/25 30C HS
FLATWOVEN 60C
FLATWOVEN 60C+LUREX
13.6%
8.9%
8.6%
6.3%
6.3%
4.7%
4.7%
4.6%
4.5%
3.9%
3.8%
3.8%
3.6%
3.5%
3.4%
3.4%
3.3%
3.3%
3.2%
2.6%
Frequency


Top 20 Most Frequent DesignName
DIAMOND
AMARI TRELLIS
MOROCCAN ZIGZAG
CHEVRON
PET BRAID
OD DOUBLE DIAMOND
PP-45
MASINISSA [TRIBAL]
STELLA
NUMA
PALI
2.7%
2.7%
2.8%
3.0%
3.1%
2.5%
2.5%
3.3%
2.4%
KOTA
3.5%
4.4%
PLAIN
17.8%
NAHLA
4.6%
4.9%
DOUBLE DIAMOND
4.9%
10.0%
5.1%
SAMPLE
9.5%
7.6%
HOMER
MODASA
TEXTURE LOOP
ELOQ GARDEN [8517]
Frequency

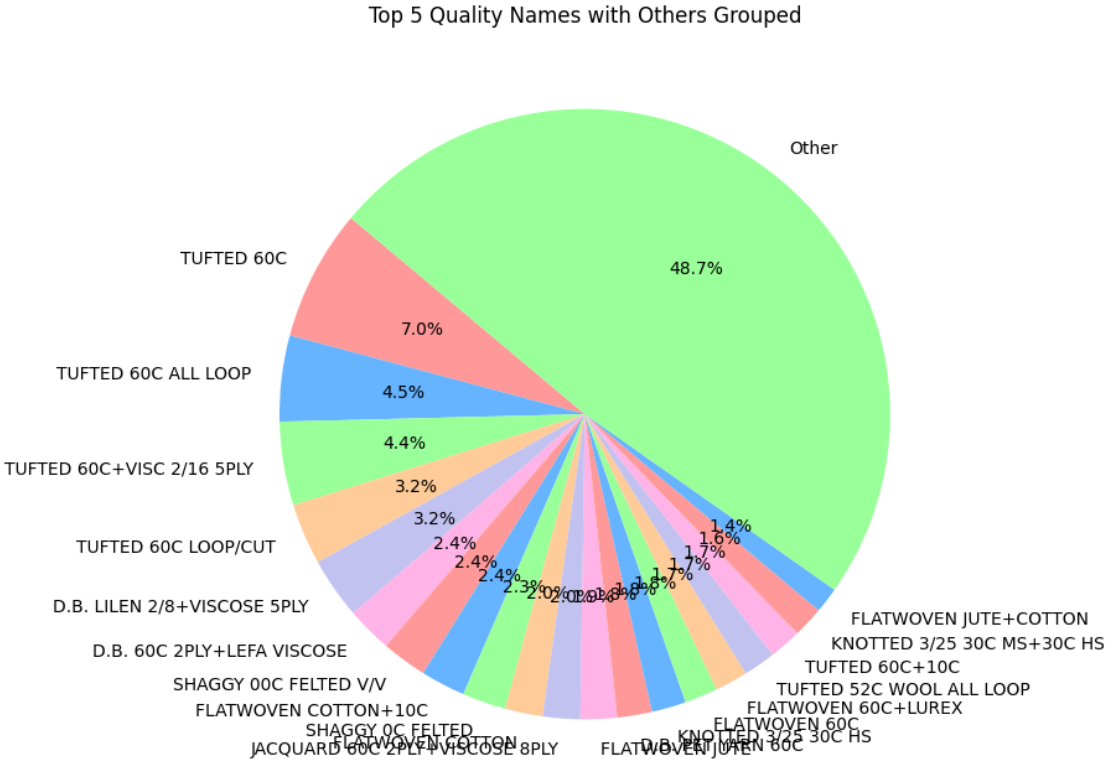

Top 20 Most Frequent ColorName
GREY
14.3%
MULTI
13.4%
BLUE
10.9%
SILVER
8.0%
BEIGE
6.9%
NAVY
6.2%
BLACK
5.4%
CHARCOAL
4.9%
NATURAL
4.5%
IVORY
4.0%
INDIGO
2.6%
NEUTRAL
2.5%
CREAM
2.4%
DENIM
2.3%
PINK
2.2%
TEAL
2.1%
SAND
1.8%
WHITE
1.8%
RED
1.7%
PETAL
Frequency



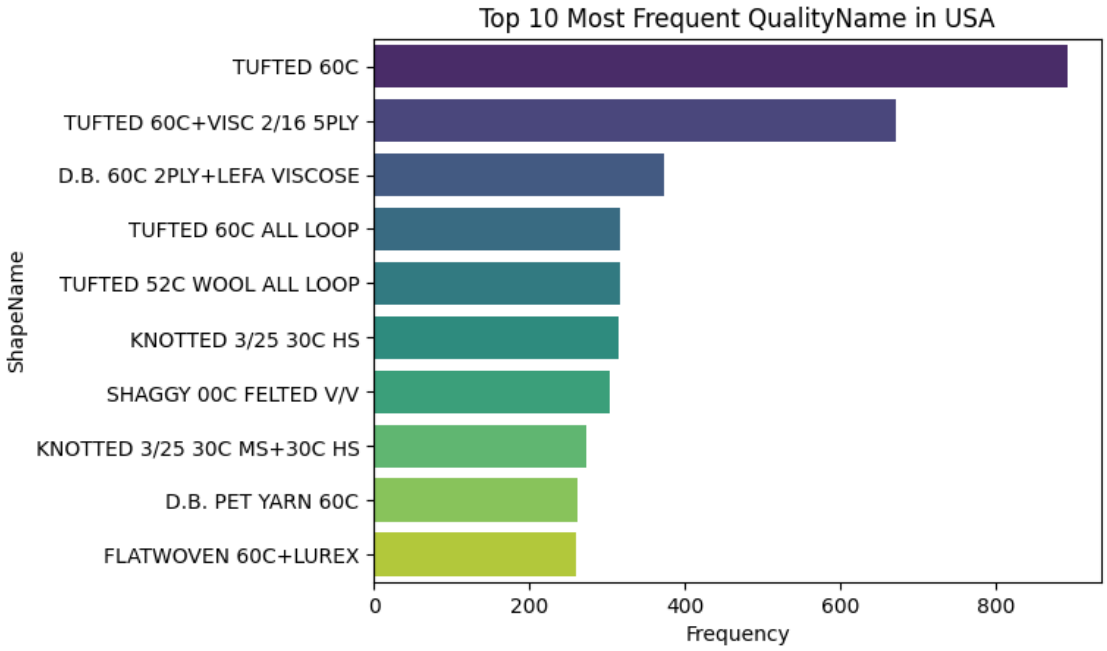
[11]:
```python
def getTopNFeatureInTopNCountry(featureName, topN):
    top_country = ['USA', 'INDIA', 'UK', 'ITALY', 'ROMANIA', 'BELGIUM']

    for country in top_country:
        country_data = df_raw_order[df_raw_order['CountryName'] == country]

        counts = country_data[featureName].value_counts().head(topN)
        sns.barplot(x=counts.values, y=counts.index, palette='viridis')
        plt.title(f'Top {topN} Most Frequent {featureName} in {country}')
        plt.xlabel('Frequency')
```

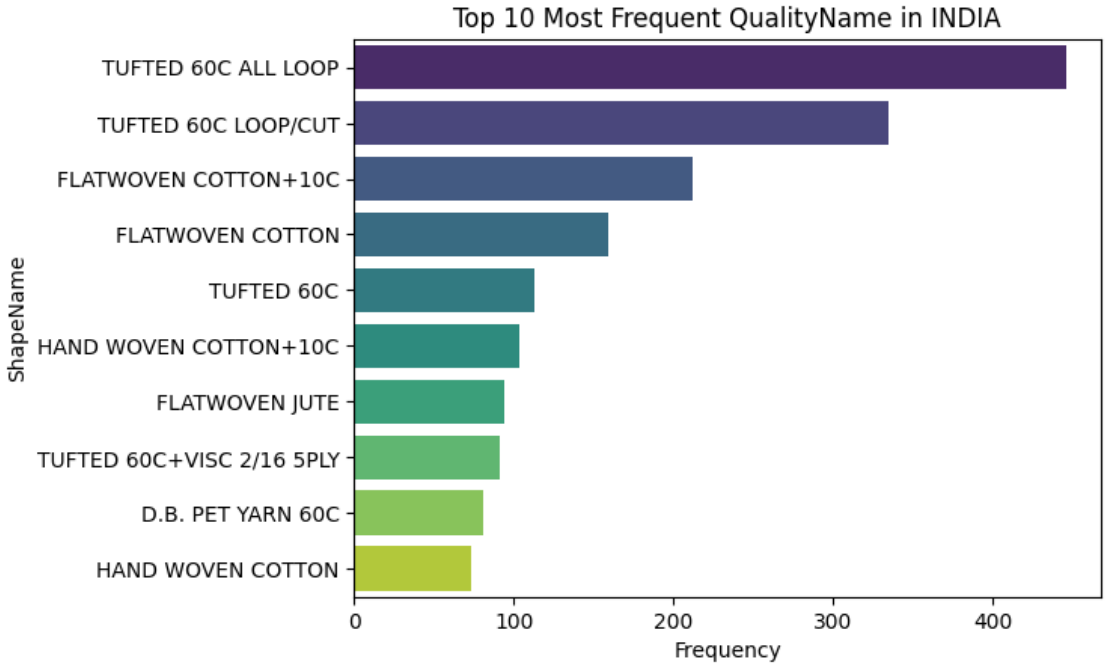
```
        plt.ylabel(i)
        plt.show()
```

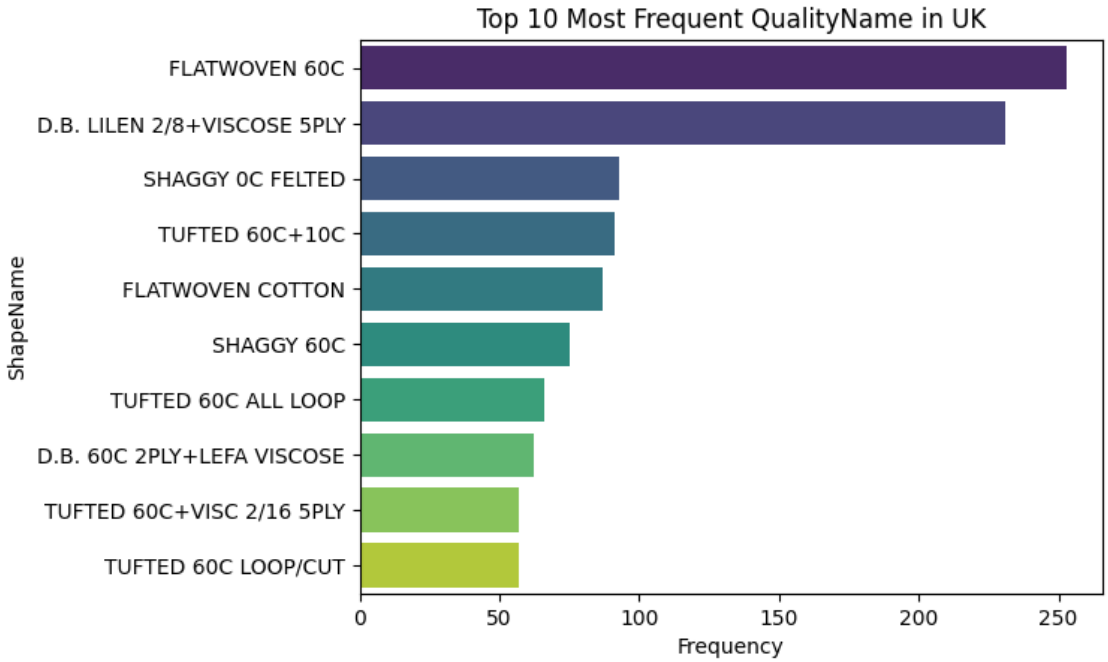
[12]:
```
# Count occurrences of each QualityName
quality_counts = df_raw_order['QualityName'].value_counts()

# Get the top 5 categories
top_5 = quality_counts.head(20)

# Sum the rest and add as 'Other'
other_count = quality_counts[20:].sum()
top_5['Other'] = other_count

plt.figure(figsize=(10, 8))
top_5.plot.pie(autopct='%1.1f%%', startangle=140,␣
 ↪colors=['#ff9999','#66b3ff','#99ff99','#ffcc99','#c2c2f0', '#ffb3e6'])
plt.title('Top 5 Quality Names with Others Grouped')
plt.ylabel('')
plt.show()
```

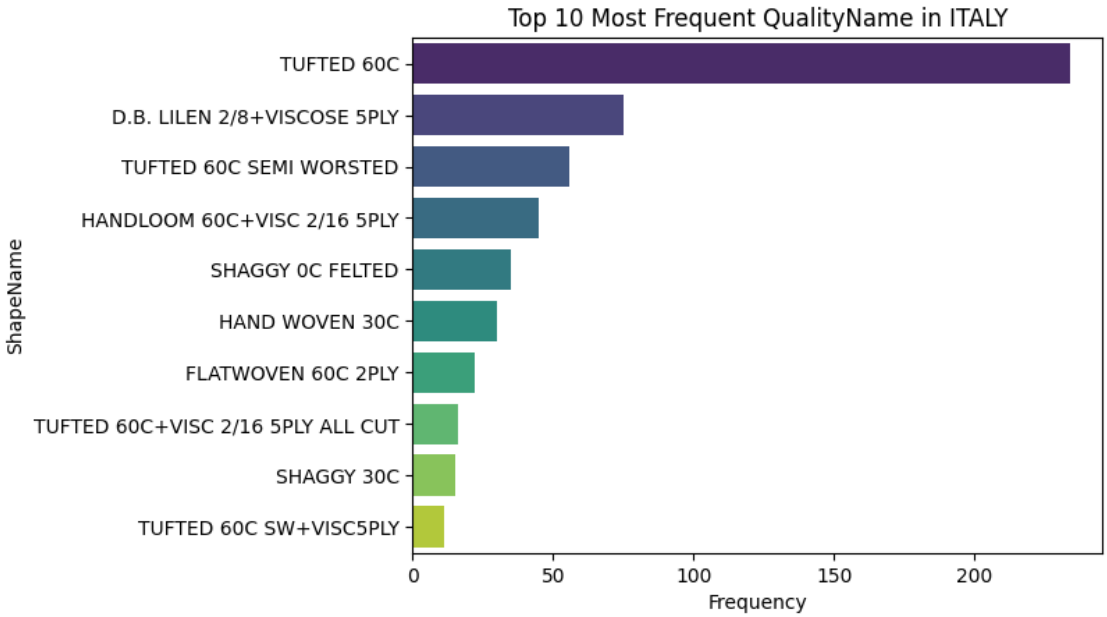


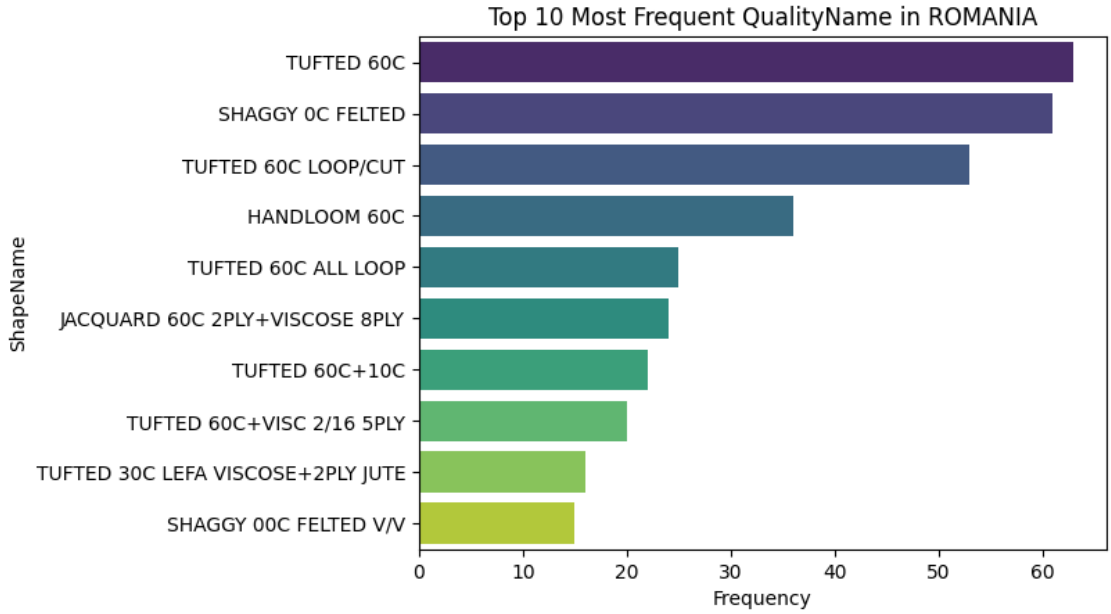
[13]:
```
getTopNFeatureInTopNCountry('QualityName', 10)
```

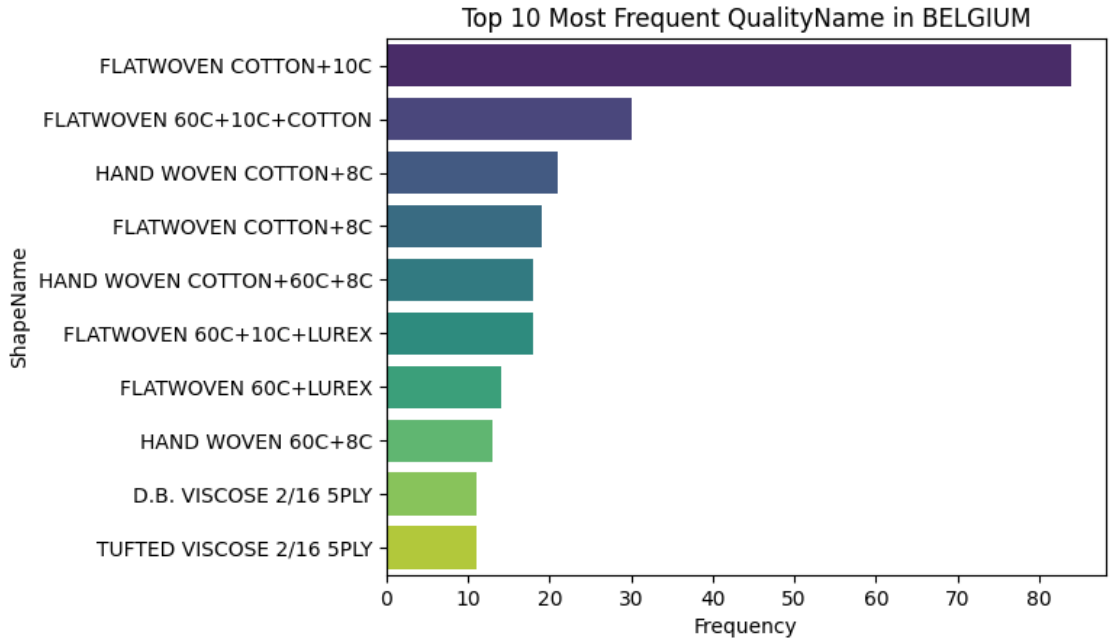

Top 10 Most Frequent QualityName in USA
TUFTED 60C
TUFTED 60C+VISC 2/16 5PLY
D.B. 60C 2PLY+LEFA VISCOSE
TUFTED 60C ALL LOOP
TUFTED 52C WOOL ALL LOOP
KNOTTED 3/25 30C HS
SHAGGY 00C FELTED V/V
KNOTTED 3/25 30C MS+30C HS
D.B. PET YARN 60C
FLATWOVEN 60C+LUREX
ShapeName
0
200
400
600
800
Frequency


Top 10 Most Frequent QualityName in INDIA
TUFTED 60C ALL LOOP
TUFTED 60C LOOP/CUT
FLATWOVEN COTTON+10C
FLATWOVEN COTTON
TUFTED 60C
HAND WOVEN COTTON+10C
FLATWOVEN JUTE
TUFTED 60C+VISC 2/16 5PLY
D.B. PET YARN 60C
HAND WOVEN COTTON
ShapeName
0
100
200
300
400
Frequency

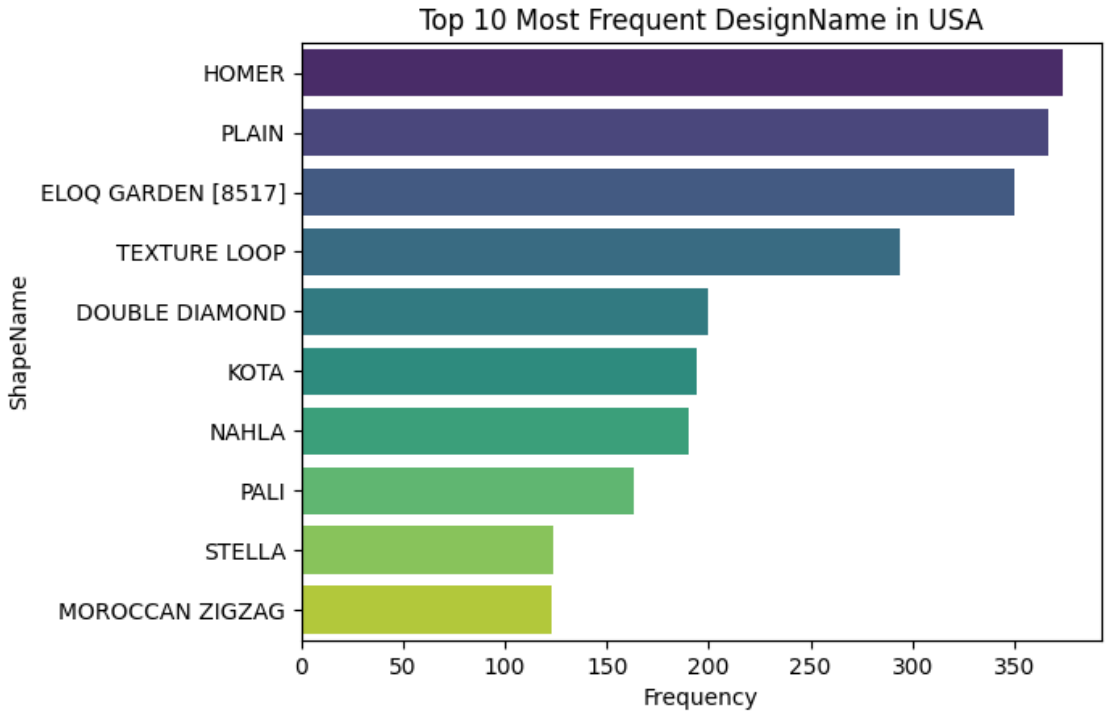

Top 10 Most Frequent QualityName in UK
FLATWOVEN 60C
D.B. LILEN 2/8+VISCOSE 5PLY
SHAGGY 0C FELTED
TUFTED 60C+10C
FLATWOVEN COTTON
SHAGGY 60C
TUFTED 60C ALL LOOP
D.B. 60C 2PLY+LEFA VISCOSE
TUFTED 60C+VISC 2/16 5PLY
TUFTED 60C LOOP/CUT
ShapeName
0
50
100
150
200
250
Frequency

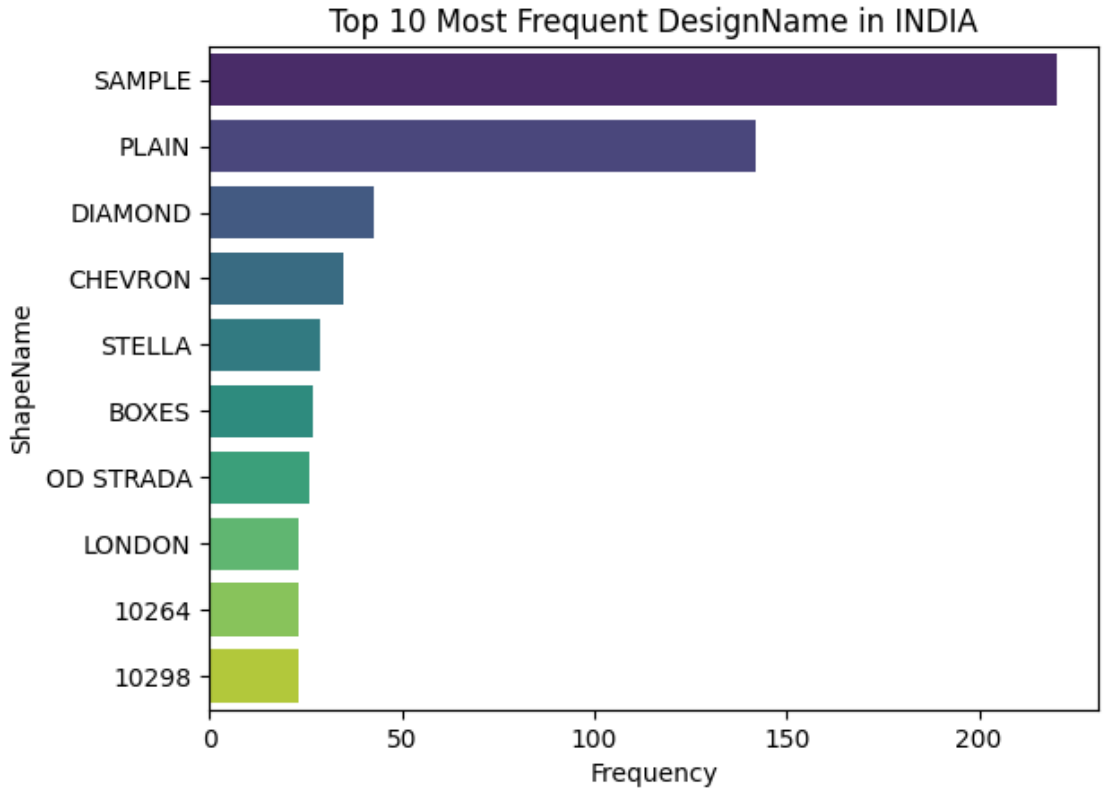

Top 10 Most Frequent QualityName in ITALY
TUFTED 60C
D.B. LILEN 2/8+VISCOSE 5PLY
TUFTED 60C SEMI WORSTED
HANDLOOM 60C+VISC 2/16 5PLY
SHAGGY 0C FELTED
HAND WOVEN 30C
FLATWOVEN 60C 2PLY
TUFTED 60C+VISC 2/16 5PLY ALL CUT
SHAGGY 30C
TUFTED 60C SW+VISC5PLY
ShapeName
0
50
100
150
200
Frequency

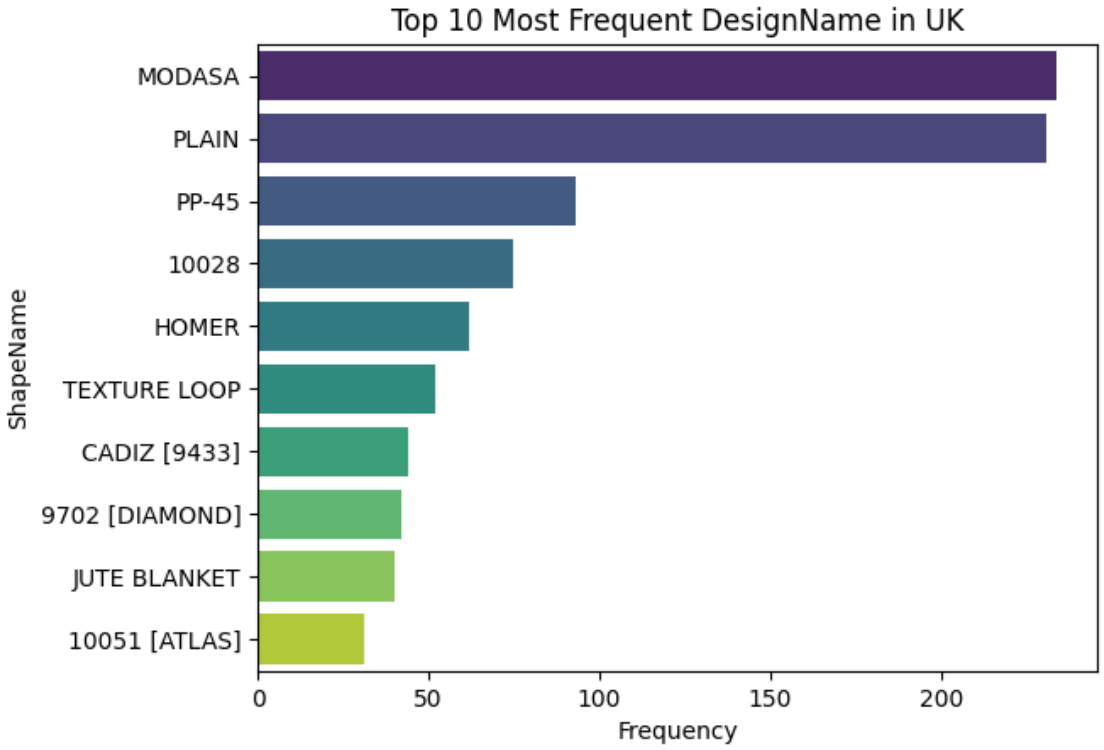


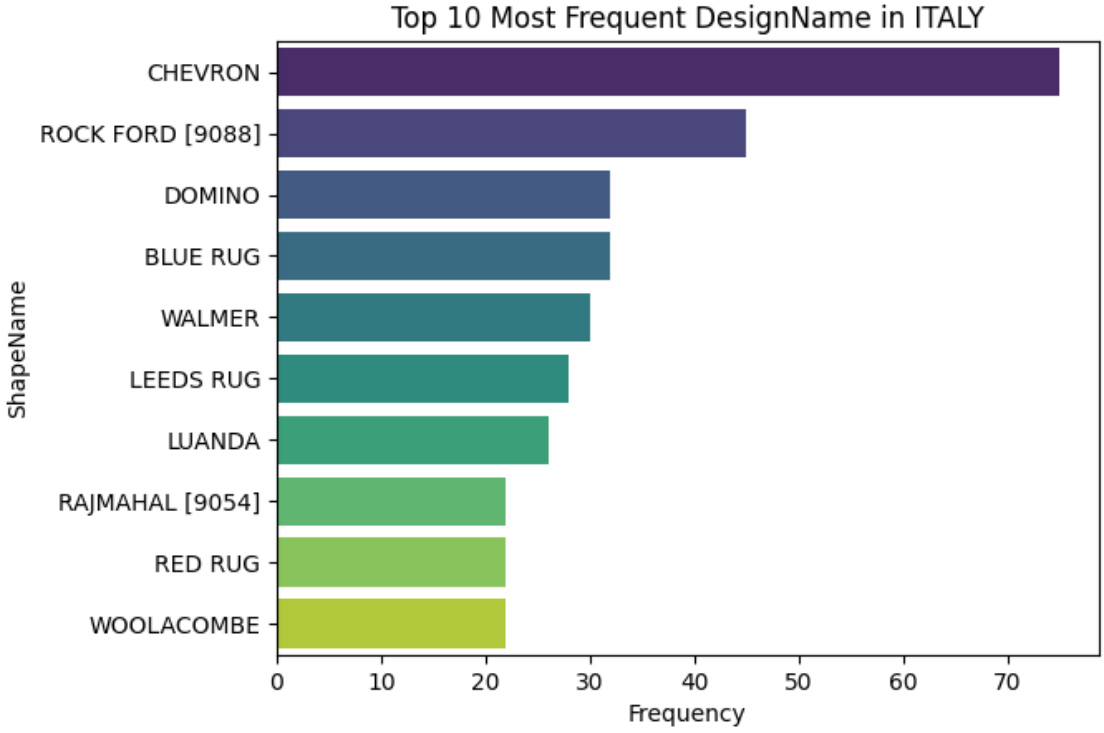


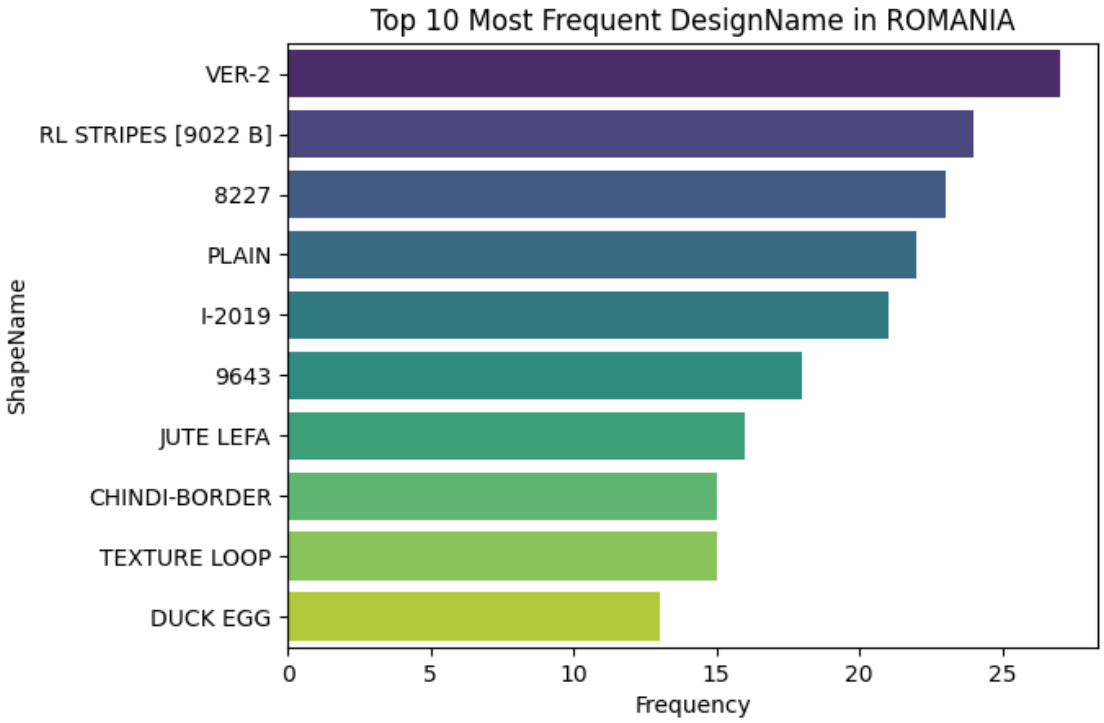
[14]:
```
getTopNFeatureInTopNCountry('DesignName', 10)
```

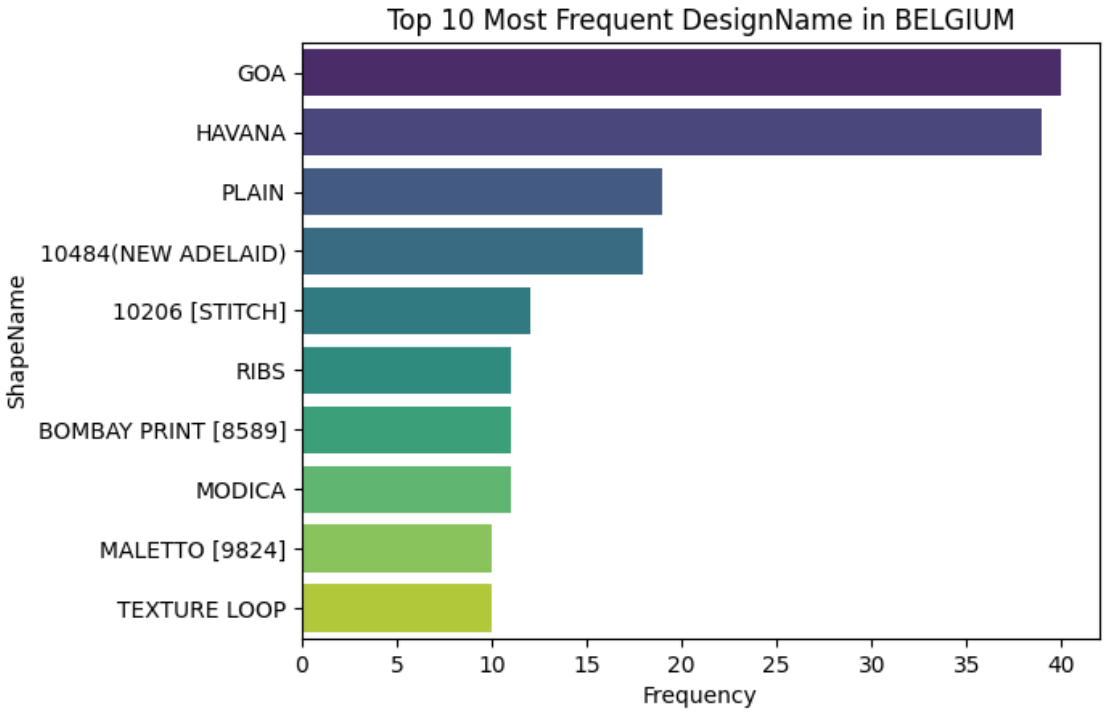

Top 10 Most Frequent DesignName in USA
HOMER
PLAIN
ELOQ GARDEN [8517]
TEXTURE LOOP
DOUBLE DIAMOND
KOTA
NAHLA
PALI
STELLA
MOROCCAN ZIGZAG
ShapeName
0
50
100
150
200
250
300
350
Frequency


Top 10 Most Frequent DesignName in INDIA
SAMPLE
PLAIN
DIAMOND
CHEVRON
STELLA
BOXES
OD STRADA
LONDON
10264
10298
ShapeName
0
50
100
150
200
Frequency


Top 10 Most Frequent DesignName in UK
MODASA
PLAIN
PP-45
10028
HOMER
TEXTURE LOOP
CADIZ [9433]
9702 [DIAMOND]
JUTE BLANKET
10051 [ATLAS]
ShapeName
0
50
100
150
200
Frequency


Top 10 Most Frequent DesignName in ITALY
CHEVRON
ROCK FORD [9088]
DOMINO
BLUE RUG
WALMER
LEEDS RUG
LUANDA
RAJMAHAL [9054]
RED RUG
WOOLACOMBE
ShapeName
0
10
20
30
40
50
60
70
Frequency


Top 10 Most Frequent DesignName in ROMANIA
VER-2
RL STRIPES [9022 B]
8227
PLAIN
I-2019
9643
JUTE LEFA
CHINDI-BORDER
TEXTURE LOOP
DUCK EGG
ShapeName
0
5
10
15
20
25
Frequency


Top 10 Most Frequent DesignName in BELGIUM
GOA
HAVANA
PLAIN
10484(NEW ADELAID)
10206 [STITCH]
RIBS
BOMBAY PRINT [8589]
MODICA
MALETTO [9824]
TEXTURE LOOP
ShapeName
0
5
10
15
20
25
30
35
40
Frequency

1.1 1.1.3. Develop ML models to help identify features that contribute toward conversion (or non-conversion) of samples sent to customers. (You can use Rapid Miner, R, or Python: We recommend Rapid Miner; see instructions to download below.)

1.1.1 Decision Trees and Random Forest for Conversion Rate

[15]:
```
def getValueCounts(df, column):
    print(df[column].value_counts())
```

[16]:
```
country_code = {
    'INDIA': 1,
    'USA': 2,
    'UK': 3,
    'BELGIUM': 4,
    'ITALY': 5,
    'ROMANIA': 6,
    'CANADA': 7,
    'AUSTRALIA': 8,
    'SOUTH AFRICA': 9,
    'BRAZIL': 10,
    'ISRAEL': 11,
    'POLAND': 12,
    'UAE': 13,
    'CHINA': 14
}

shape_code = {
    'REC': 1,
    'ROUND': 2,
    'SQUARE': 3,
}

df_sample_order['CountryCode'] = df_sample_order['CountryName'].
 ↪replace(country_code)
df_sample_order['ShapeCode'] = df_sample_order['ShapeName'].replace(shape_code)


getValueCounts(df_sample_order, 'ShapeCode')

df_sample_order_formatted = df_sample_order.drop(['REC', 'Round', 'Square',␣
 ↪'CountryName', 'USA', 'UK', 'Italy', 'Belgium', 'Romania', 'Australia',␣
 ↪'India', 'QtyRequired', 'AreaFt', 'ShapeName', 'ITEM_NAME', 'CustomerCode'],␣
 ↪axis=1)

df_sample_order_formatted.head()
```

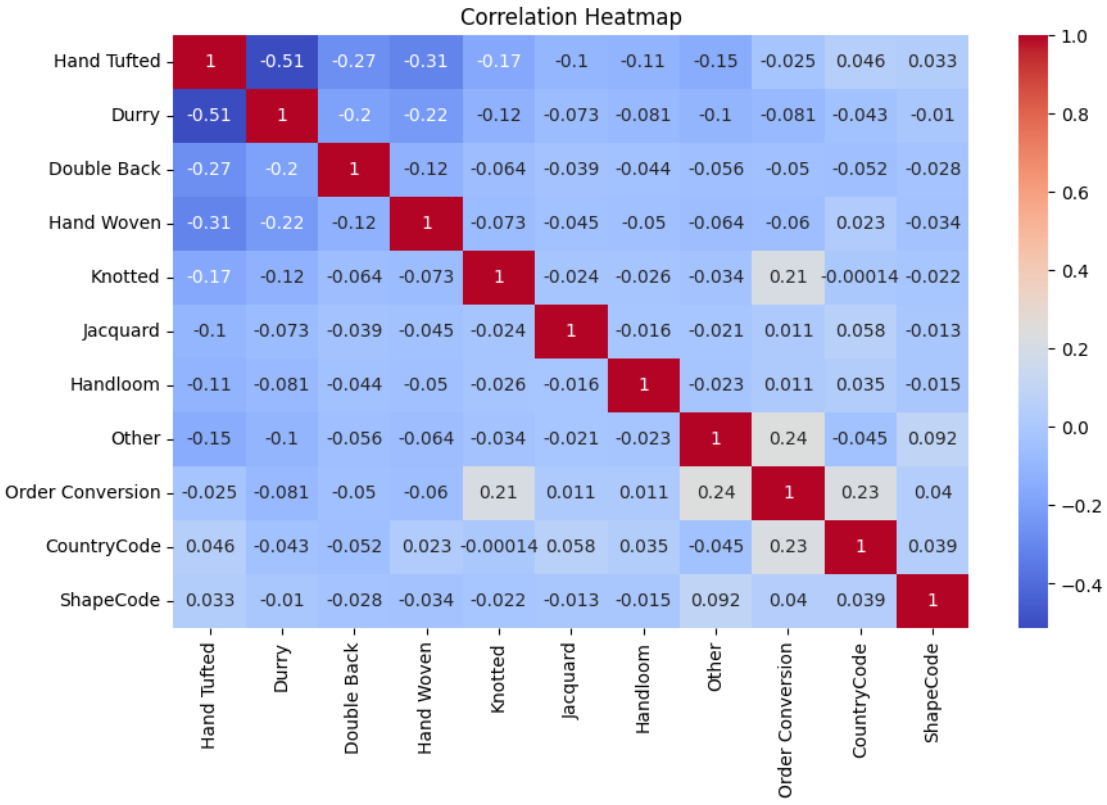
```
1    5741
2      57
3      22
Name: ShapeCode, dtype: int64
```

[16]:

	Hand Tufted	Durry	Double Back	Hand Woven	Knotted	Jacquard	Handloom	\
0	1	0	0	0	0	0	0	
1	1	0	0	0	0	0	0	
2	1	0	0	0	0	0	0	
3	1	0	0	0	0	0	0	
4	1	0	0	0	0	0	0	

	Other	Order Conversion	CountryCode	ShapeCode
0	0	1	1	1
1	0	1	2	1
2	0	1	2	1
3	0	1	2	1
4	0	1	2	1

[17]:
```
plt.figure(figsize=(10, 6))
correlation_matrix = df_sample_order_formatted.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```



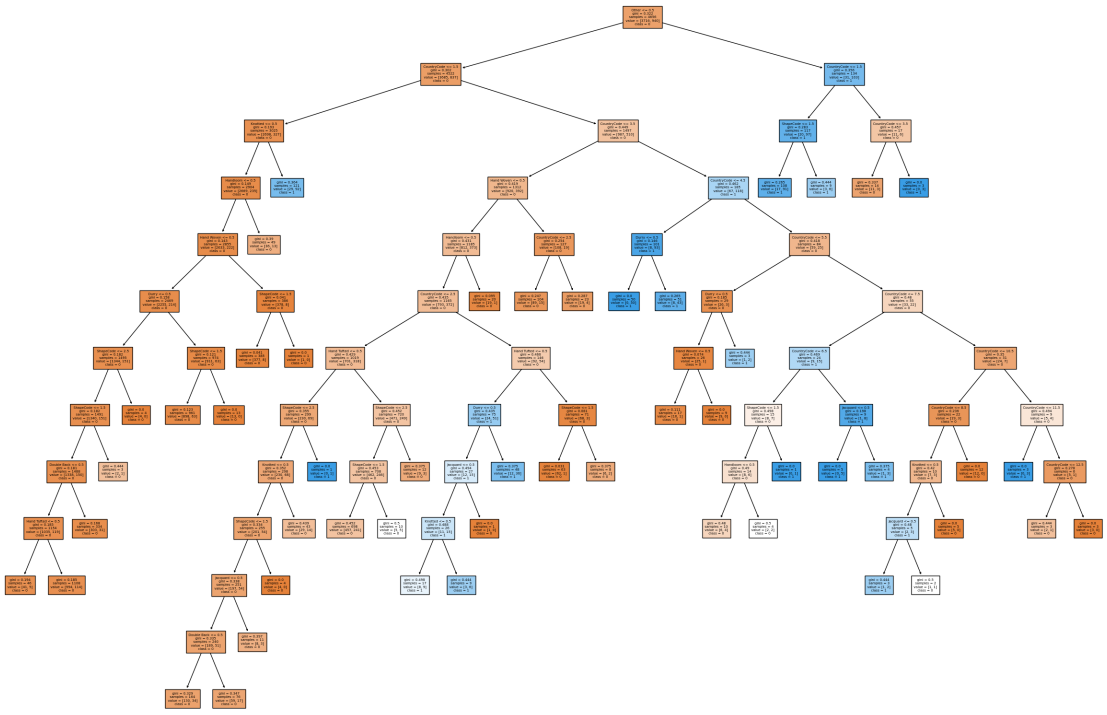
[53]:
```
X = df_sample_order_formatted.drop('Order Conversion', axis=1)
X_train, X_test, y_train, y_test = train_test_split(X,␣
↪df_sample_order_formatted['Order Conversion'], test_size = 0.2, random_state␣
↪= 5)
```

[52]:
```
clf = DecisionTreeClassifier(random_state=5)
clf.fit(X_train, y_train)
```

[52]:
```
DecisionTreeClassifier(random_state=5)
```

Decision Tree

[54]:
```
plt.figure(figsize = (30, 20))
plot_tree(clf, filled = True, feature_names = X_train.columns, class_names =␣
↪list(map(str, clf.classes_)), fontsize = 5)
plt.show()
```



[43]:
```
param_grid = {'max_leaf_nodes': [5, 10, 15, 20, 25]}
grid_search = GridSearchCV(clf, param_grid, cv=5, scoring='accuracy')
grid_search.fit(X_train, y_train)
```

```
best_params = grid_search.best_params_
print(f"Best Parameters: {best_params}")

best_classifier = grid_search.best_estimator_

plt.figure(figsize = (20, 15))
plot_tree(best_classifier, filled = True, feature_names = X_train.columns,␣
 ↪class_names=list(map(str, best_classifier.classes_)), fontsize=10)
plt.show()
```

```
Best Parameters: {'max_leaf_nodes': 25}
```

[44]:
```
y_pred = best_classifier.predict(X_test)
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.87      0.98      0.92       935
           1       0.80      0.39      0.52       229

    accuracy                           0.86      1164
   macro avg       0.83      0.68      0.72      1164
```

```
weighted avg       0.85      0.86      0.84      1164
```

[55]:
```
rf_clf = RandomForestClassifier(n_estimators=100, random_state=5,␣
 ↪max_leaf_nodes = 25)
rf_clf.fit(X_train, y_train)
rd_predict = rf_clf.predict(X_test)
print(classification_report(y_test, rd_predict))
```

```
              precision    recall  f1-score   support

           0       0.87      0.97      0.92       932
           1       0.78      0.40      0.53       232

    accuracy                           0.86      1164
   macro avg       0.82      0.68      0.72      1164
weighted avg       0.85      0.86      0.84      1164
```

1.1.2 K Means Clustering

[56]:
```
numeric_transformer = Pipeline(steps=[
    ('scaler', StandardScaler())
])
categorical_transformer = Pipeline(steps=[
    ('onehot', OneHotEncoder(handle_unknown='ignore'))
])

preprocessor = ColumnTransformer(
    transformers=[
        ('num', numeric_transformer, numeric_features),
        ('cat', categorical_transformer, classification_features)
    ])

# Pipeline with preprocessing and K-means clustering
pipeline = Pipeline(steps=[('preprocessor', preprocessor),
                           ('kmeans', KMeans(n_clusters=
                                             5, random_state=5))])

# Fit pipeline
pipeline.fit(df_raw_order[['ITEM_NAME','CountryName', 'QualityName',␣
 ↪'DesignName', 'ColorName', 'ShapeName', 'QtyRequired', 'TotalArea',␣
 ↪'Amount']])
```

```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of
`n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
```

```
  warnings.warn(
```

[56]:
```
Pipeline(steps=[('preprocessor',
                 ColumnTransformer(transformers=[('num',
                                                  Pipeline(steps=[('scaler',
StandardScaler())]),
                                                  ['QtyRequired', 'TotalArea',
                                                   'Amount']),
                                                 ('cat',
                                                  Pipeline(steps=[('onehot',
OneHotEncoder(handle_unknown='ignore'))]),
                                                  ['CountryName', 'ITEM_NAME',
                                                   'QualityName', 'DesignName',
                                                   'ColorName',
                                                   'ShapeName'])])),
                ('kmeans', KMeans(n_clusters=5, random_state=5))])
```

[59]:
```python
# Assign cluster labels to original data
df_raw_order['Cluster'] = pipeline['kmeans'].labels_

first_order_cluster = df_raw_order.iloc[0]['Cluster']
similar_orders = df_raw_order[df_raw_order['Cluster'] == first_order_cluster]

print(f"Similar Orders:\n{similar_orders}")

# Example: Predict cluster for new data (e.g., new order)
new_order = pd.DataFrame({
    'CountryName': 'USA',
    'QtyRequired': [3],
    'TotalArea': [10],
    'Amount': [20],
    'QualityName': ['TUFTED 30C HARD TWIST'],
    'DesignName': ['OLD LONDON [3715]'],
    'ColorName': ['BEIGE'],
    'ShapeName': ['REC'],
    'ITEM_NAME': 'HAND TUFTED'
})

predicted_cluster = pipeline.predict(new_order)

print(f"Predicted Cluster for New Order: {predicted_cluster[0]}")
```

```
Similar Orders:
       OrderType OrderCategory CustomerCode CountryName CustomerOrderNo  \
0      Area Wise         Order          H-1         USA         1873354
1      Area Wise         Order          H-1         USA         1873354
2      Area Wise         Order          H-1         USA         1873354
3      Area Wise         Order          H-1         USA         1918436
```

```
8      Area Wise         Order          H-1         USA         1873354
…           …            …           …          …              …
18950  Area Wise        Sample          T-2       ITALY          S 1278
18951  Area Wise        Sample          T-2       ITALY          S 1278
18952  Area Wise        Sample          T-2       ITALY          S 1278
18953  Area Wise        Sample          A-9         USA          S 1280
18954  Area Wise        Sample           CC       INDIA          S 1281

      Custorderdate UnitName  QtyRequired  TotalArea  Amount    ITEM_NAME  \
0        2017-01-16       Ft            2     6.0000    12.0  HAND TUFTED
1        2017-01-16       Ft            2     9.0000    18.0  HAND TUFTED
2        2017-01-16       Ft            2    54.0000   108.0  HAND TUFTED
3        2017-02-01       Ft            5    54.0000   270.0  HAND TUFTED
8        2017-01-16       Ft            2    36.0000    72.0  HAND TUFTED
…              …       …           …         …      …           …
18950    2020-02-13      Mtr            1     0.2500     0.0  HAND TUFTED
18951    2020-02-13      Mtr            1     0.2500     0.0  HAND TUFTED
18952    2020-02-13      Mtr            1     0.2500     0.0  HAND TUFTED
18953    2020-02-14       Ft            2     6.0000     0.0  HAND TUFTED
18954    2020-02-14       Ft            1    39.8125     0.0  HAND TUFTED

                            QualityName         DesignName    ColorName  \
0                  TUFTED 30C HARD TWIST  OLD LONDON [3715]        BEIGE
1                  TUFTED 30C HARD TWIST  OLD LONDON [3715]        BEIGE
2                  TUFTED 30C HARD TWIST  OLD LONDON [3715]        BEIGE
3                  TUFTED 30C HARD TWIST  OLD LONDON [3715]        BEIGE
8                  TUFTED 30C HARD TWIST  OLD LONDON [3715]  GREEN/IVORY
…                                    …                 …          …
18950  TUFTED 60C+VISC 2/16 5PLY ALL CUT         MONOGRAMMA        GREEN
18951                TUFTED 60C ALL CUT         MONOGRAMMA  BLACK/BLACK
18952  TUFTED 60C+VISC 2/16 5PLY ALL CUT         MONOGRAMMA        IVORY
18953                        TUFTED 30C             9164 B   IVORY BLUE
18954                TUFTED 60C ALL LOOP              10807    BLUE GREY

      ShapeName  Unnamed: 15   AreaFt  Cluster
0           REC            1   6.0000        1
1           REC            1   9.0000        1
2           REC            1  54.0000        1
3           REC            1  54.0000        1
8         ROUND            1  36.0000        1
…           …              …       …        …
18950       REC            1   2.7778        1
18951       REC            1   2.7778        1
18952       REC            1   2.7778        1
18953       REC            1   6.0000        1
18954       REC            1  39.8125        1

[5706 rows x 18 columns]
```

```
Predicted Cluster for New Order: 1
```