Load Libraries

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import pickle
from sklearn import metrics
import seaborn as sns
import matplotlib.pyplot as plt
```

Load Datasets

In [3]:
```
dataset_pype = pd.read_excel('Comparison Dataset/Catalent BWI.xlsx', dtype = object, engine = 'openpyxl')
dataset_production = pd.read_excel('Comparison Dataset/Catalent BWI - Submittal Log.xlsx', dtype = object, engi

# PyPe Dataset
dataset_pype.head()
```

```
C:\Users\KishanT\Anaconda3\lib\site-packages\openpyxl\worksheet\_reader.py:312: UserWarning: Data Validation ex
tension is not supported and will be removed
  warn(msg)
```

Out[3]:

	S. No.	Spec #	Spec Name	Para	Sub Section Heading	Submittal Type	Submittal Description	Target Date	Subcontractor
0	44	024119	SELECTIVE DEMOLITION	1.10-A	WARRANTY	Warranty	Existing Warranties : Remove, replace, patch, ...	NaN	NaN
1	45	024119	SELECTIVE DEMOLITION	1.10-B	WARRANTY	Warranty	Notify warrantor on completion of selective de...	NaN	NaN
2	36	024119	SELECTIVE DEMOLITION	1.5-A	PREINSTALLATION MEETINGS	Meetings	Predemolition Conference : Conduct conference ...	NaN	NaN
3	37	024119	SELECTIVE DEMOLITION	1.6-A	INFORMATIONAL SUBMITTALS	Measurements	Proposed Protection Measures : Submit report, ...	NaN	NaN
4	38	024119	SELECTIVE DEMOLITION	1.6-B	INFORMATIONAL SUBMITTALS	Schedules	Schedule of Selective Demolition Activities : ...	NaN	NaN

In [4]:
```
# Production Dataset
dataset_production.head()
```

Out[4]:

	Submittal ID	Submittal Name	Unnamed: 2	Spec Section Number	Spec Sub Section	No. of Copies	Type Code	Preparation By Company Code	Preparation By Contact Code	Approved by Company Code	...	Forwar Comp C
0	NaN	Polished Concrete Finishing_Polishing	yes	033543	1.4-B	Polishing Schedule : Submit plan showing polis...	Schedule	Schedule	NaN	NaN	...	N
1	NaN	Polished Concrete Finishing_Product Requiring ...	yes	033543	1.4-C	NaN	Sample	NaN	NaN	NaN	...	N
2	NaN	Polished Concrete Finishing_Installer_Informat...	yes	033543	1.5-A	NaN	Qualifications	NaN	NaN	NaN	...	N
3	NaN	Polished Concrete Finishing_Repair Materials_I...	yes	033543	1.5-B-1	NaN	Certificate	NaN	NaN	NaN	...	N
4	NaN	Polished Concrete Finishing_Stain Materials_In...	yes	033543	1.5-B-2	NaN	Certificate	NaN	NaN	NaN	...	N

5 rows × 23 columns

Drop Unwanted Features

In [166...
```
dataset_pype = dataset_pype.drop(columns = dataset_pype.columns[[0, 2, 4, 5, 7, 8]])
dataset_pype.head()
```

Out[166...

	Spec #	Para	Submittal Description
0	024119	1.10-A	Existing Warranties : Remove, replace, patch, ...
1	024119	1.10-B	Notify warrantor on completion of selective de...
2	024119	1.5-A	Predemolition Conference : Conduct conference ...
3	024119	1.6-A	Proposed Protection Measures : Submit report, ...
4	024119	1.6-B	Schedule of Selective Demolition Activities : ...

Change Column Names

In [167...
```
dataset_pype.columns = ['Spec Section Number', 'Spec Sub Section','Submittal Description']
dataset_pype.head()
```

Out[167...

	Spec Section Number	Spec Sub Section	Submittal Description
0	024119	1.10-A	Existing Warranties : Remove, replace, patch, ...
1	024119	1.10-B	Notify warrantor on completion of selective de...
2	024119	1.5-A	Predemolition Conference : Conduct conference ...
3	024119	1.6-A	Proposed Protection Measures : Submit report, ...
4	024119	1.6-B	Schedule of Selective Demolition Activities : ...

Remove Different Records

In [168...
```
## Remove Different Rows
#
dataset_pype.drop(dataset_pype.index[641:654], inplace = True)
dataset_pype.reset_index(inplace = True, drop = True)
dataset_pype.drop(dataset_pype.index[691:693], inplace = True)
dataset_pype.reset_index(inplace = True, drop = True)
dataset_pype.drop(dataset_pype.index[662:663], inplace = True)
dataset_pype.reset_index(inplace = True, drop = True)
```

Change Datatype Of Features

In [169...
```
## Change DataType Of Feature
#
dataset_pype['Spec Section Number'] = dataset_pype['Spec Section Number'].astype('float')
```

Store PyPe Result

In [142...
```
## Store Pype Results
#
dataset_pype.to_excel('PYPE.xlsx')
```

In [170...
```
## Empty DataFrame
#
dataset_production_filtered = pd.DataFrame()
dataset_production_filtered.head()
```

Out[170... —

In [171...
```
## Filter and Create New Dataframe In Proper Format
#
def data_filter(row):
    spec_section_number = row[3]
    spec_subsection = row[4]
    submittal_type = row[6]

    previous = ""
    now = '-'.join(str(spec_subsection).split('-')[0:2])

    if(previous == now):
        pass
    else:
        previous = now
        return spec_section_number, previous, submittal_type

dataset_production_filtered[['Spec Section Number', 'Spec Sub Section', 'Submittal Type']] = dataset_productio
```

In [172...
```
## Filter and Create New Dataframe In Proper Format
#
def data_filter(row):
    spec_section_number = row[0]
    if(str(spec_section_number).endswith('.00')):
        return int(str(spec_section_number)[0:-3])
    else:
        return spec_section_number

dataset_production_filtered['Spec Section Number'] = dataset_production_filtered.apply(func = data_filter, axis
```

In [173...
```
## Change DataType Of Feature
#
dataset_production_filtered['Spec Section Number'] = dataset_production_filtered['Spec Section Number'].astype
```

In [174...
```
dataset_production_filtered['Submittal Type'].unique()

# ['Attic Stock', 'Calculations', 'Certificates', 'Color', 'Chart Delivery', 'Maintenance Data' , 'Manufacture
#            'Mix Design', 'Mockups', 'MSDS', 'Owner Training', 'Product Data', 'Pre-Install Meeting Minutes',
#            'Reports', 'Samples', 'Schedules', 'Shop Drawings', 'Certifications']
```

Out[174...
```
array(['Schedule', 'Sample', 'Qualifications', 'Certificate',
       'Product Data', 'Shop Drawings', 'Mix Design', 'Report',
       'Procedures', 'Record Drawing', 'Test Data', 'Warranty',
       'Leed Requirements', 'Calculations', 'MSDS', 'Maintenance Data',
       'report', 'product Data', 'Attic Stock'], dtype=object)
```

In [177...
```
dataset_production_filtered = dataset_production_filtered[(dataset_production_filtered['Submittal Type'] != 're
# & (dataset_production_filtered['Submittal Type'] == 'product Data')]
```

In [178...
```
dataset_production_filtered = dataset_production_filtered[(dataset_production_filtered['Submittal Type'] != 'pr
```

In [179...
```
dataset_production_filtered['Submittal Type'].unique()

# ['Attic Stock', 'Calculations', 'Certificates', 'Color', 'Chart Delivery', 'Maintenance Data' , 'Manufacture
#            'Mix Design', 'Mockups', 'MSDS', 'Owner Training', 'Product Data', 'Pre-Install Meeting Minutes',
#            'Reports', 'Samples', 'Schedules', 'Shop Drawings', 'Certifications']
```

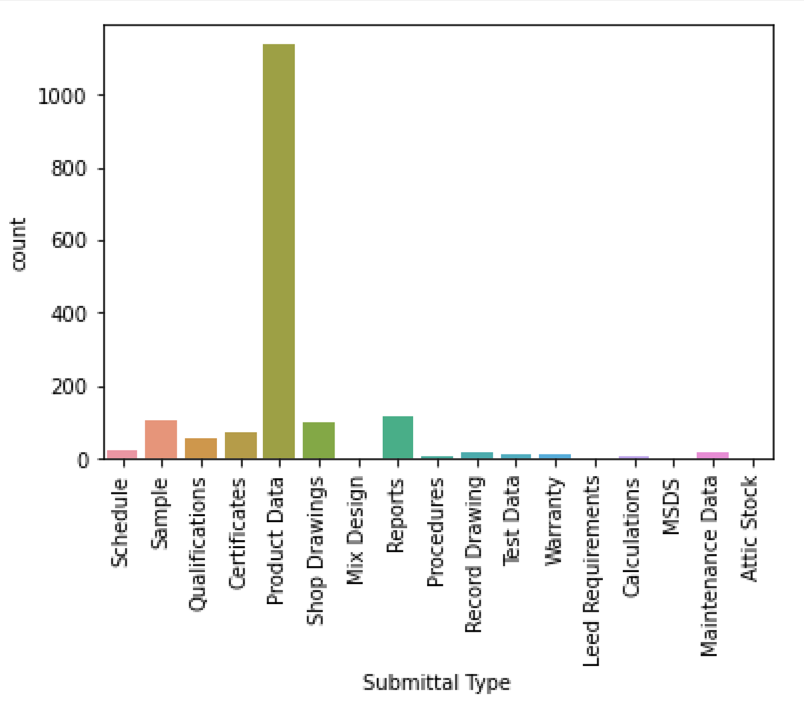
Out[179...
```
array(['Schedule', 'Sample', 'Qualifications', 'Certificate',
       'Product Data', 'Shop Drawings', 'Mix Design', 'Report',
       'Procedures', 'Record Drawing', 'Test Data', 'Warranty',
       'Leed Requirements', 'Calculations', 'MSDS', 'Maintenance Data',
       'Attic Stock'], dtype=object)
```

In [181...
```
## Data Correct
#
def data_correct(row):
    sub_tupe = row[2]
    if(sub_tupe == 'Certificate'):
        return 'Certificates'
    elif(sub_tupe == 'Report'):
        return 'Reports'
    else:
        return sub_tupe

dataset_production_filtered['Submittal Type'] = dataset_production_filtered.apply(func = data_correct, axis = 1
```

In [182...
```
## Visualize count
#
sns.countplot(data = dataset_production_filtered,
              x = 'Submittal Type')

plt.xticks(rotation = 90)
plt.show()
```

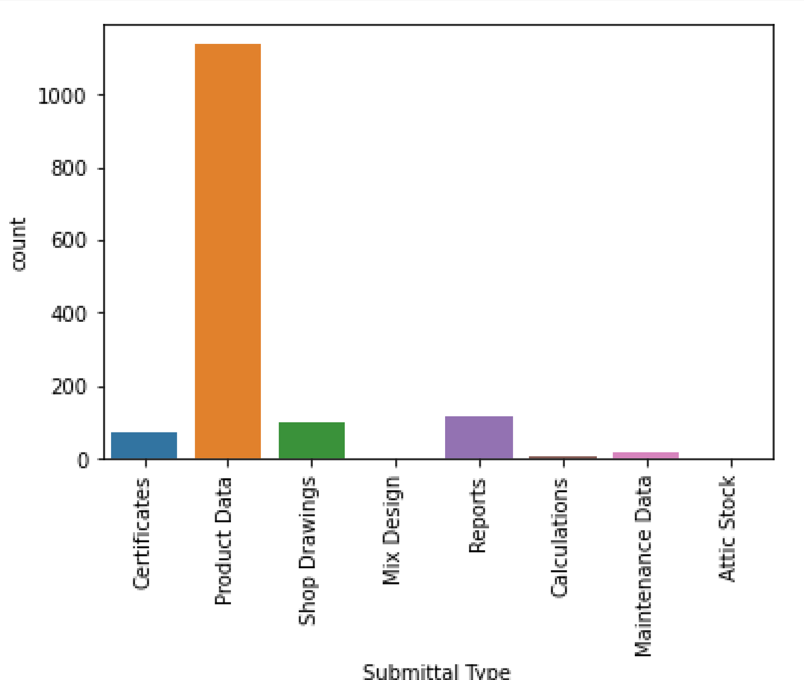


In [183...
```
options = ['Attic Stock', 'Calculations', 'Certificates', 'Certifications', 'Maintenance Data', 'Mix Design', '
           'Product Data', 'Reports', 'Samples', 'Schedules', 'Shop Drawings']
```

In [184...
```
dataset_production_filtered = dataset_production_filtered[dataset_production_filtered['Submittal Type'].isin(op
```

In [185...
```
## Visualize count
#
sns.countplot(data = dataset_production_filtered,
              x = 'Submittal Type')

plt.xticks(rotation = 90)
plt.show()
```



In [186...
```
## Merge Both Dataset On Spec Section Number and Spec Sub Section
#
final_dataset = pd.merge(dataset_production_filtered, dataset_pype, on = ['Spec Section Number', 'Spec Sub Sect
final_dataset.head()
```

Out[186...

	Spec Section Number	Spec Sub Section	Submittal Type	Submittal Description
0	33543.0	1.5-B	Certificates	Material Certificates : For each of the follow...
1	33543.0	1.5-B	Certificates	Material Certificates : For each of the follow...
2	33543.0	1.5-B	Certificates	Material Certificates : For each of the follow...
3	42200.0	1.5-B	Shop Drawings	Shop Drawings : For the following: \n1. Masonr...
4	42200.0	1.5-B	Shop Drawings	Shop Drawings : For the following: \n1. Masonr...

In [187...
```
## Remove Duplicate Records
#
final_dataset.drop_duplicates(inplace = True)
```

In [190...
```
## Store Merged Dataset For Prediction
#
final_dataset.to_excel('merged_test_dataset.xlsx')
```

In [191...
```
## Load Saved Model, Vectorizer and Encoder
#
with open("vectorizer.pickle", 'rb+') as file:
    vectorizer_saved = pickle.load(file)

with open("label_encoder.pickle", 'rb+') as file:
    encorder_saved = pickle.load(file)

with open("type_classifier.pickle", 'rb+') as file:
    classifier_saved = pickle.load(file)
```

In [192...
```
## Encoder Sumittal Type
#
y_test = encorder_saved.transform(final_dataset['Submittal Type'])
```

In [193...
```
X_test_tfidf = vectorizer_saved.transform(final_dataset['Submittal Description'])
```

In [194...
```
predictions = classifier_saved.predict(X_test_tfidf)
```

In [195...
```
score = metrics.accuracy_score(y_test, predictions) * 100
score
```

Out[195...
```
74.76190476190476
```

In [89]:
```
# final_dataset['Predicted Labels'] = encorder_saved.inverse_transform(predictions)
```

In [92]:
```
# final_dataset.to_excel("Predicted.xlsx")
```

In [ ]:
```
!jupyter nbconvert --to PDFviaHTML "production_comparison.ipynb"
```