

In [1]:

```

import matplotlib.pyplot as plt
import numpy as np
import os
import pandas as pd
import seaborn as sns

from sklearn.pipeline import Pipeline
from sklearn.preprocessing import OneHotEncoder
from sklearn.model_selection import train_test_split, GridSearchCV, KFold
from sklearn.metrics import average_precision_score, precision_recall_curve, PrecisionRecall

import warnings
warnings.filterwarnings("ignore")

sns.set({"figure.figsize": (20, 15)})

```

In [2]:

```
df = pd.read_csv("Telco Custo Churn.csv")
```

In [3]:

```
df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype  
---  -
 0   customerID            7043 non-null   object  
 1   gender                 7043 non-null   object  
 2   SeniorCitizen          7043 non-null   int64   
 3   Partner                7043 non-null   object  
 4   Dependents             7043 non-null   object  
 5   tenure                 7043 non-null   int64   
 6   PhoneService           7043 non-null   object  
 7   MultipleLines          7043 non-null   object  
 8   InternetService        7043 non-null   object  
 9   OnlineSecurity         7043 non-null   object  
10   OnlineBackup           7043 non-null   object  
11   DeviceProtection       7043 non-null   object  
12   TechSupport            7043 non-null   object  
13   StreamingTV            7043 non-null   object  
14   StreamingMovies        7043 non-null   object  
15   Contract               7043 non-null   object  
16   PaperlessBilling       7043 non-null   object  
17   PaymentMethod          7043 non-null   object  
18   MonthlyCharges         7043 non-null   float64  
19   TotalCharges           7043 non-null   object  
20   Churn                  7043 non-null   object  
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB

```

In [4]:

```
df['TotalCharges'].value_counts()
```

Out[4]:

```

20.2      11
19.75      9
20.05      8
19.9       8
..
6849.4     1
692.35     1
130.15     1
3211.9     1
6844.5     1
Name: TotalCharges, Length: 6531, dtype: int64
```

In [5]:

```

# object to numerical
df['TotalCharges'] = df['TotalCharges'].apply(lambda x: np.nan if x == ' ' else x)
df['TotalCharges'] = df['TotalCharges'].astype(np.float)
```

In [6]:

```

# numerical columns
df.describe()
```

Out[6]:

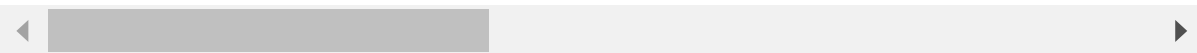
	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
count	7043.000000	7043.000000	7043.000000	7032.000000
mean	0.162147	32.371149	64.761692	2283.300441
std	0.368612	24.559481	30.090047	2266.771362
min	0.000000	0.000000	18.250000	18.800000
25%	0.000000	9.000000	35.500000	401.450000
50%	0.000000	29.000000	70.350000	1397.475000
75%	0.000000	55.000000	89.850000	3794.737500
max	1.000000	72.000000	118.750000	8684.800000

In [7]:

```
# categorical columns  
df.describe(include='object')
```

Out[7]:

	customerID	gender	Partner	Dependents	PhoneService	MultipleLines	InternetService
count	7043	7043	7043	7043	7043	7043	7043
unique	7043	2	2	2	2	3	3
top	7590-VHVEG	Male	No	No	Yes	No	Fiber optic
freq	1	3555	3641	4933	6361	3390	3096



In [8]:

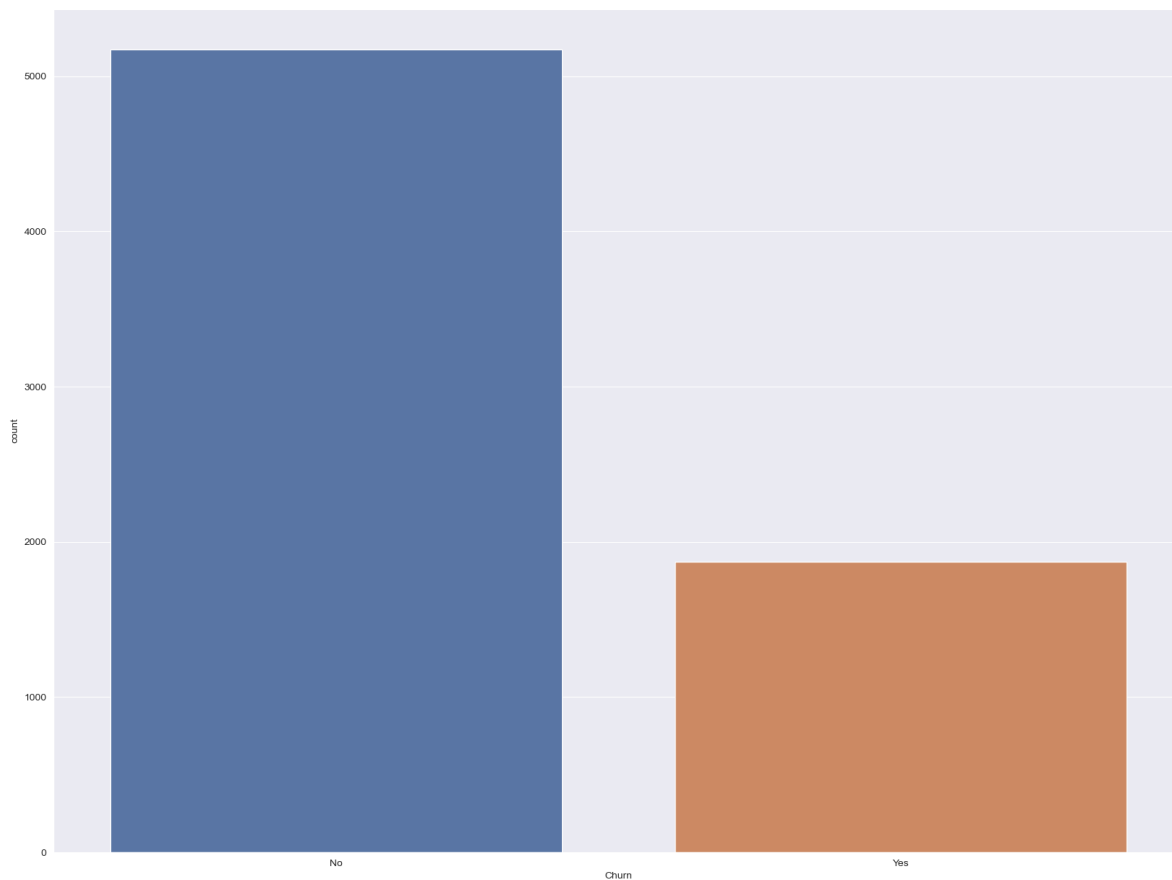
```
numerical = ['TotalCharges', 'tenure', 'MonthlyCharges']  
target = 'Churn'  
exception_col = numerical + [target]  
categorical = [col for col in df.columns if col not in exception_col]
```

In [9]:

```
sns.countplot(df[target])
```

Out[9]:

<AxesSubplot:xlabel='Churn', ylabel='count'>



In [10]:

```
100 * df[target].value_counts() / df.shape[0]
```

Out[10]:

```
No      73.463013
Yes     26.536987
Name: Churn, dtype: float64
```

In [11]:

```
df = df.drop(columns=['customerID'])
```

In [12]:

```
rotation_angle = 0
hue_column = target

fig, ax = plt.subplots(nrows=5, ncols=3, **{'figsize': (30, 30)})

graph_1 = sns.countplot(data=df, x='gender', hue=hue_column, ax=ax[0][0])
graph_1.set_xticklabels(graph_1.get_xticklabels(), rotation=rotation_angle)

graph_2 = sns.countplot(data=df, x='Partner', hue=hue_column, ax=ax[0][1])
graph_2.set_xticklabels(graph_2.get_xticklabels(), rotation=rotation_angle)

graph_3 = sns.countplot(data=df, x='Dependents', hue=hue_column, ax=ax[0][2])
graph_3.set_xticklabels(graph_3.get_xticklabels(), rotation=rotation_angle)

graph_4 = sns.countplot(data=df, x='PhoneService', hue=hue_column, ax=ax[1][0])
graph_4.set_xticklabels(graph_4.get_xticklabels(), rotation=rotation_angle)

graph_5 = sns.countplot(data=df, x='MultipleLines', hue=hue_column, ax=ax[1][1])
graph_5.set_xticklabels(graph_5.get_xticklabels(), rotation=rotation_angle)

graph_6 = sns.countplot(data=df, x='InternetService', hue=hue_column, ax=ax[1][2])
graph_6.set_xticklabels(graph_6.get_xticklabels(), rotation=rotation_angle)

graph_7 = sns.countplot(data=df, x='OnlineSecurity', hue=hue_column, ax=ax[2][0])
graph_7.set_xticklabels(graph_7.get_xticklabels(), rotation=rotation_angle)

graph_8 = sns.countplot(data=df, x='OnlineBackup', hue=hue_column, ax=ax[2][1])
graph_8.set_xticklabels(graph_8.get_xticklabels(), rotation=rotation_angle)

graph_9 = sns.countplot(data=df, x='DeviceProtection', hue=hue_column, ax=ax[2][2])
graph_9.set_xticklabels(graph_9.get_xticklabels(), rotation=rotation_angle)
graph_10 = sns.countplot(data=df, x='TechSupport', hue=hue_column, ax=ax[3][0])
graph_10.set_xticklabels(graph_10.get_xticklabels(), rotation=rotation_angle)

graph_11 = sns.countplot(data=df, x='StreamingTV', hue=hue_column, ax=ax[3][1])
graph_11.set_xticklabels(graph_11.get_xticklabels(), rotation=rotation_angle)

graph_12 = sns.countplot(data=df, x='StreamingMovies', hue=hue_column, ax=ax[3][2])
graph_12.set_xticklabels(graph_12.get_xticklabels(), rotation=rotation_angle)

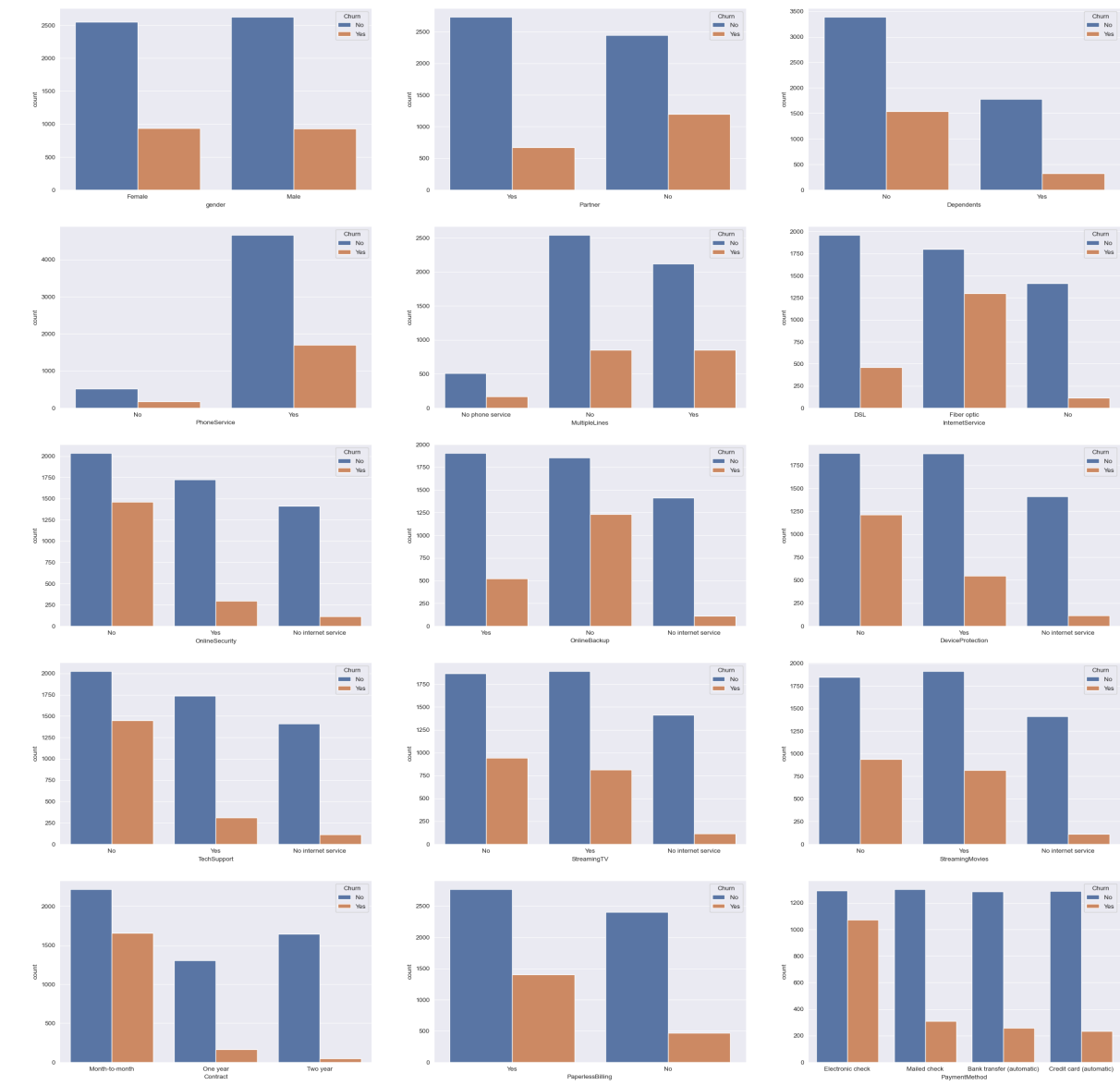
graph_13 = sns.countplot(data=df, x='Contract', hue=hue_column, ax=ax[4][0])
graph_13.set_xticklabels(graph_13.get_xticklabels(), rotation=rotation_angle)

graph_14 = sns.countplot(data=df, x='PaperlessBilling', hue=hue_column, ax=ax[4][1])
graph_14.set_xticklabels(graph_14.get_xticklabels(), rotation=rotation_angle)

graph_15 = sns.countplot(data=df, x='PaymentMethod', hue=hue_column, ax=ax[4][2])
graph_15.set_xticklabels(graph_15.get_xticklabels(), rotation=rotation_angle)
```

Out[12]:

```
[Text(0, 0, 'Electronic check'),
 Text(1, 0, 'Mailed check'),
 Text(2, 0, 'Bank transfer (automatic)'),
 Text(3, 0, 'Credit card (automatic))]
```

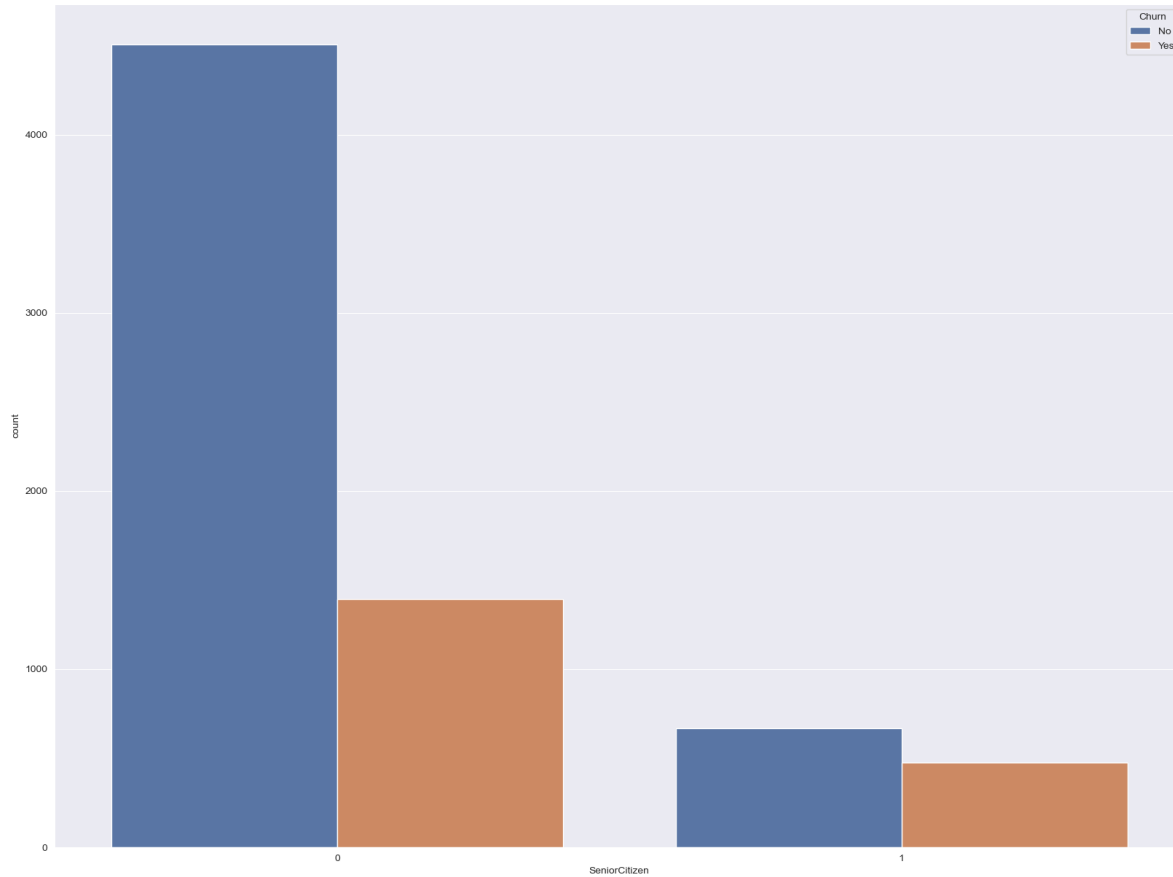


In [13]:

```
graph_16 = sns.countplot(data=df, x='SeniorCitizen', hue=hue_column)  
graph_16.set_xticklabels(graph_16.get_xticklabels(),rotation=rotation_angle)
```

Out[13]:

```
[Text(0, 0, '0'), Text(1, 0, '1')]
```



In [14]:

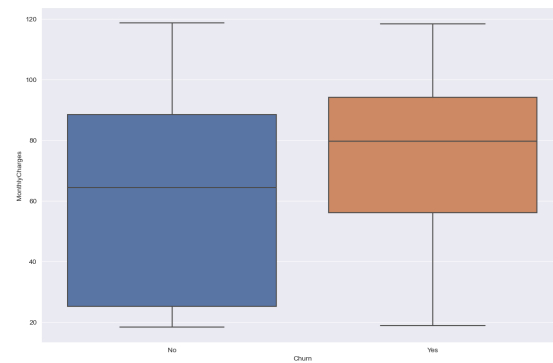
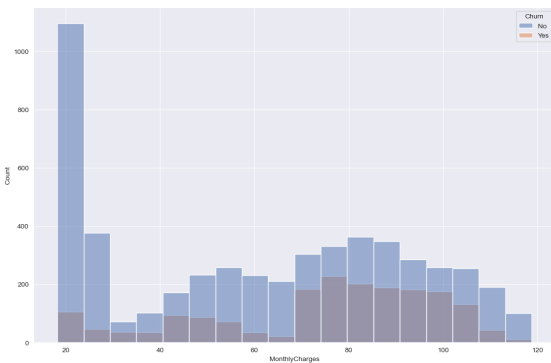
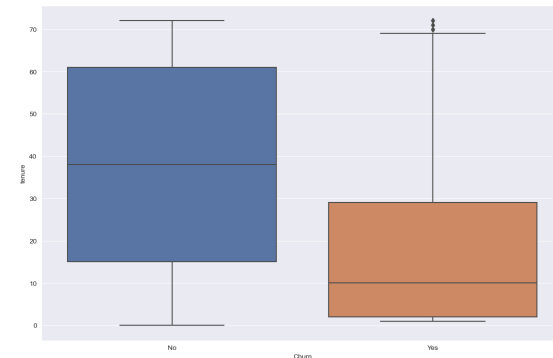
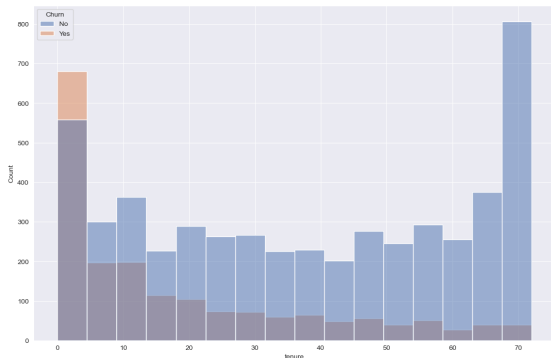
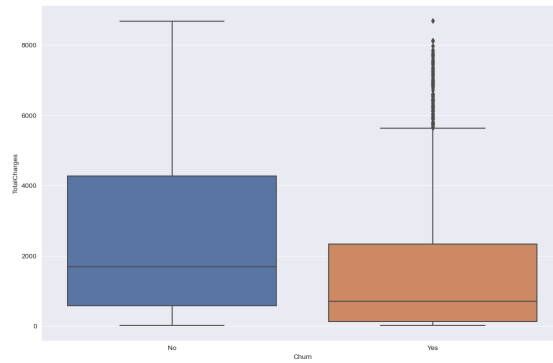
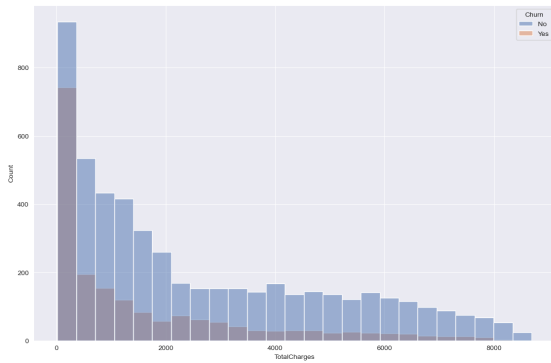
```

types_graph = {'hist', 'box'}

fig, ax = plt.subplots(nrows=3, ncols=2, **{'figsize': (30, 30)})

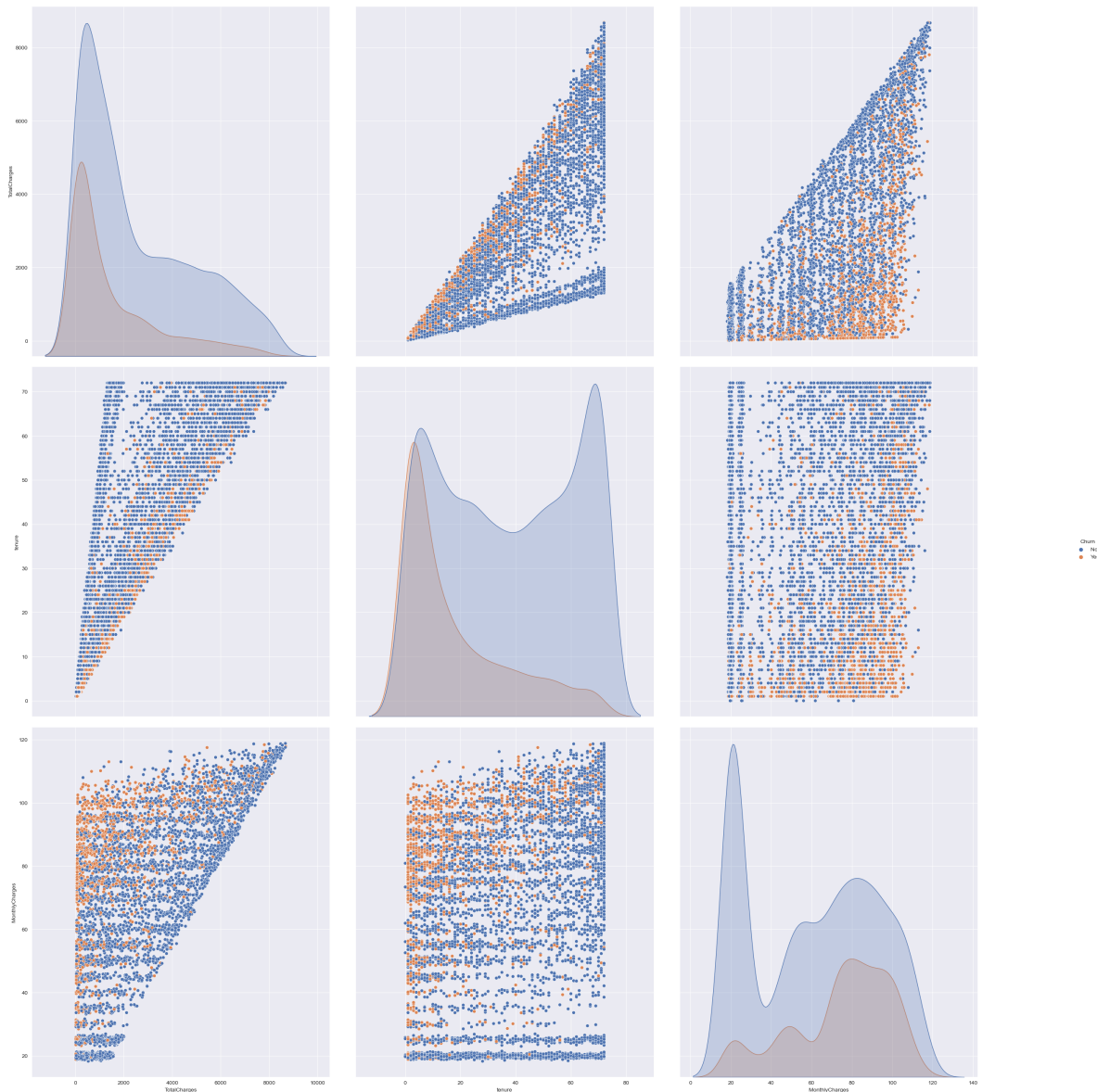
for i, col in enumerate(numerical):
    for j, graph in enumerate(types_graph):
        tmp_ax = ax[i][j]
        if graph == 'hist':
            sns.histplot(data=df, x=col, hue=hue_column, ax=tmp_ax)
        else:
            sns.boxplot(data=df, y=col, x=hue_column, ax=tmp_ax)

```



In [15]:

```
num_pair_graph = sns.pairplot(data=df[numerical + [hue_column]], hue=hue_column)
num_pair_graph.fig.set_size_inches(30, 30)
```



In [16]:

```
charges_median = df['TotalCharges'].median()
df['TotalCharges'] = df['TotalCharges'].fillna(df['TotalCharges'])
```