#**Correlation of full dataset**

scaler = StandardScaler()

scaled= scaler.fit\_transform(X)

scaled\_dataframe=pd.DataFrame(data=scaled[1:,0:])

scaled\_dataframe.columns=['blueWardsPlaced', 'blueWardsDestroyed', 'blueFirstBlood', 'blueKills',

'blueDeaths', 'blueAssists', 'blueEliteMonsters', 'blueDragons',

'blueHeralds', 'blueTowersDestroyed', 'blueTotalGold', 'blueAvgLevel',

'blueTotalExperience', 'blueTotalMinionsKilled',

'blueTotalJungleMinionsKilled', 'blueGoldDiff', 'blueExperienceDiff',

'blueCSPerMin', 'blueGoldPerMin', 'redWardsPlaced', 'redWardsDestroyed',

'redFirstBlood', 'redKills', 'redDeaths', 'redAssists',

'redEliteMonsters', 'redDragons', 'redHeralds', 'redTowersDestroyed',

'redTotalGold', 'redAvgLevel', 'redTotalExperience',

'redTotalMinionsKilled', 'redTotalJungleMinionsKilled', 'redGoldDiff',

'redExperienceDiff', 'redCSPerMin', 'redGoldPerMin']

plt.figure(figsize=(18,14))

corr\_plot=sns.heatmap(round(scaled\_dataframe.corr(),2), cmap='Blues', annot=False)

plt.show()

figure = corr\_plot.get\_figure()

figure.savefig('corr\_plot.png', dpi=400)

**#Violin plot for intermediate dataset**

cleaned\_data=pd.read\_csv("cleaned.csv")

cleaned\_data=cleaned\_data.drop(["blueWins"],axis=1)

cleaned\_data.columns

scaler = StandardScaler()

new\_scaled= scaler.fit\_transform(cleaned\_data)

new\_scaled\_dataframe=pd.DataFrame(data=new\_scaled[1:,0:])

new\_scaled\_dataframe=pd.concat([Y,new\_scaled\_dataframe],axis=1)

new\_scaled\_dataframe.columns=['blueWins','blueFirstBlood', 'exblueWardsPlaced', 'exblueWardsDestroyed',

'exblueKills', 'exblueDeaths', 'exblueAssists', 'exblueDragons',

'exblueHeralds', 'exblueTowersDestroyed', 'exblueTotalGold',

'exblueAvgLevel', 'exblueTotalExperience', 'exblueTotalMinionsKilled',

'exblueTotalJungleMinionsKilled', 'exblueCSPerMin', 'exblueGoldPerMin']

new\_scaled\_dataframe

plt.figure(figsize=(20,5))

new\_scaled\_dataframe = pd.melt(new\_scaled\_dataframe, id\_vars='blueWins', var\_name='Features', value\_name='Values')

colors = ["#FF0B04", "#4374B3"]

sns.set\_palette(sns.color\_palette(colors))

new\_scaled\_dataframe

ax = sns.violinplot(x="Features", y="Values", hue="blueWins",data=new\_scaled\_dataframe ,split=True)

figure = ax.get\_figure()

figure.savefig('violin\_\_plot.png', dpi=400)

**#Correlation plot for cleaned data**

final=pd.read\_csv("cleaned\_final.csv")

final=final.drop(['blueWins'],axis=1)

final.columns

scaler = StandardScaler()

final\_scaled =scaler.fit\_transform(final)

final\_data=pd.DataFrame(data=final\_scaled[1:,0:])

final\_data.columns=['blueFirstBlood', 'blueActiveWards', 'blueKills', 'blueAssists',

'blueDragons', 'blueHeralds', 'blueTowersDestroyed', 'blueAvgLevel',

'blueTotalExperience', 'blueTotalMinionsKilled',

'blueTotalJungleMinionsKilled', 'blueCSPerMin', 'blueGoldPerMin',

'redActiveWards', 'redKills', 'redAssists', 'redDragons', 'redHeralds',

'redTowersDestroyed', 'redAvgLevel', 'redTotalExperience',

'redTotalMinionsKilled', 'redTotalJungleMinionsKilled', 'redCSPerMin',

'redGoldPerMin']

final\_data

plt.figure(figsize=(18,14))

final\_corr\_plot=sns.heatmap(round(final\_data.corr(),2), cmap='Blues', annot=False)

plt.show()

figure = final\_corr\_plot.get\_figure()

figure.savefig('final\_corr\_plot.png', dpi=400)

#Number of blueWins & redWins

colors = ["#FF0B04", "#4374B3"]

sns.set\_palette(sns.color\_palette(colors))

ax = sns.countplot(Y, label='Count')

figure = ax.get\_figure()

figure.savefig('blueWins.png', dpi=400)

**#Box plot - blueWins vs blueGoldPerMin**

final\_data\_with\_response=pd.concat([Y,final],axis=1)

final\_data\_with\_response

figure1= sns.boxplot(x="blueWins", y="blueGoldPerMin", data=final\_data\_with\_response)

figure = figure1.get\_figure()

figure.savefig('blueWins\_VS\_blueGold.png', dpi=400)

**#Box plot - blueWins vs redGoldPerMin**

figure2= sns.boxplot(x="blueWins", y="redGoldPerMin", data=final\_data\_with\_response)

figure = figure2.get\_figure()

figure.savefig('blueWins\_VS\_redGold.png', dpi=400)

**#Box plot - blueWins vs blueTotalExperience**

figure3= sns.boxplot(x="blueWins", y="blueTotalExperience", data=final\_data\_with\_response)

figure = figure3.get\_figure()

figure.savefig('blueWins\_VS\_blueExperience.png', dpi=400)

**#Box plot - blueWins vs redTotalExperience**

figure4= sns.boxplot(x="blueWins", y="redTotalExperience", data=final\_data\_with\_response)

figure = figure4.get\_figure()

figure.savefig('blueWins\_VS\_redExperience.png', dpi=400)

**#Scatter plot - blueGoldPerMin vs blueTotalExperience with respect to blueWins**

figure5=sns.scatterplot(data=final\_data\_with\_response, x="blueGoldPerMin", y="blueTotalExperience", hue="blueWins")

figure = figure5.get\_figure()

figure.savefig('blueExperience\_VS\_blueGold.png', dpi=400)

**#Scatter plot - redGoldPerMin vs redTotalExperience with respect to blueWins**

figure6=sns.scatterplot(data=final\_data\_with\_response, x="redGoldPerMin", y="redTotalExperience", hue="blueWins")

figure = figure6.get\_figure()

figure.savefig('redExperience\_VS\_redGold.png', dpi=400)

**#Scatter plot - redGoldPerMin vs redAvgLevel with respect to blueWins**

figure7=sns.scatterplot(data=final\_data\_with\_response, x="redGoldPerMin", y="redAvgLevel", hue="blueWins")

figure = figure7.get\_figure()

figure.savefig('redLevel\_VS\_redGold.png', dpi=400)

**#Scatter plot - blueGoldPerMin vs blueAvgLevel with respect to blueWins**

figure8=sns.scatterplot(data=final\_data\_with\_response, x="blueGoldPerMin", y="blueAvgLevel", hue="blueWins")

figure = figure8.get\_figure()

figure.savefig('blueLevel\_VS\_blueGold.png', dpi=400)

**#Box plot - blueWins vs blueKills**

figure9= sns.boxplot(x="blueWins", y="blueKills", data=final\_data\_with\_response)

figure = figure9.get\_figure()

figure.savefig('blueWins\_VS\_blueKills.png', dpi=400)

**#Box plot - blueWins vs redKills**

figure10= sns.boxplot(x="blueWins", y="redKills", data=final\_data\_with\_response)

figure = figure10.get\_figure()

figure.savefig('blueWins\_VS\_redKills.png', dpi=400)