

```

42
43     return dadosPaa
44
45
46 def ApplyPca(df, samples):
47
48     pca = PCA(n_components=len(df.columns)).fit(df)
49     explained_var = pca.explained_variance_ratio_ #
    variancia explicada do PCA
50     for exp_var_count in range(1, 7):
51         sum_var = sum([explained_var[i] for i in range(
exp_var_count)])
52         print("Variância total dos primeiros {}
componentes: {}".format(exp_var_count, sum_var))
53
54         var1 = np.cumsum(np.round(pca.
explained_variance_ratio_, decimals=4) * 100)
55         #plt.plot(var1)
56         #plt.show()
57         pca = PCA(n_components=20).fit(df) # aplica a
quantidade de componentes prevista pelo teste com as
amostras
58         reduced_data = pca.fit_transform(df) # aplicação do
pca
59         pca_samples = pca.fit_transform(samples) # idem, am
ans amostras
60
61         reduced_data: DataFrame = pd.DataFrame(reduced_data)
62
63         plt.subplot(212)
64         plt.plot(reduced_data)
65         plt.title('Com PCA Sallen Key')
66         plt.show()
67
68         plt.plot(pca_samples)
69         plt.title('Amostras para validação de resultados')
70         plt.show()
71
72         return(reduced_data, pca_samples)
73
74 def SupervisedPreds(df, clf):
75     '''
76     Aplica um único método de cada vez.
77     :param df: dataframe com dados em que se deseja
aplicar o aprendizado

```