

```

115 supervisionado
116         # modelo: 8 modelos diferentes; em destaque:
        NaiveBayes
117
        # =====
        =====
118         from sklearn.ensemble import AdaBoostClassifier
119         from sklearn.tree import DecisionTreeClassifier
120         from sklearn.ensemble import
        RandomForestClassifier
121         from sklearn.naive_bayes import GaussianNB
122         from sklearn.neighbors import
        KNeighborsClassifier
123         from sklearn.linear_model import SGDClassifier
124         from sklearn.linear_model import
        LogisticRegression
125
126
127         classifiers = [DecisionTreeClassifier(
        random_state=20),AdaBoostClassifier(random_state=20),
128                             svm.SVC(kernel='linear', C=1,
        random_state=20),RandomForestClassifier(random_state=20),
129                             GaussianNB(),KNeighborsClassifier(
        ),SGDClassifier(random_state=20),
130                             LogisticRegression(random_state=20
        )]
131
        k=0
132         #classifiers = [GaussianNB()]
133         for clf in classifiers:
134             acc_train_results, acc_test_results, \
135             fscore_train_results, fscore_test_results, \
136             clfs = AuxiliaryFunctions.SupervisedPreds(
        dadosPaa, clf)
137
138             print("acurácia teste: {} \t acurácia treino
        : {} \n fscore teste: {} \t fscore treino: {} \n".format(
139                 acc_test_results, acc_train_results,
        fscore_test_results, fscore_train_results))
140             for ct in range(10):
141                 rd = np.random.randint(0, 3300)
142                 print("Predição de {}: {}".format(rd,
        clfs.predict(dadosPaa.iloc[rd, :].values.reshape(1, -1)))
        )
143
144             for j in range(3300):

```