

1) Importando as Funções Necessárias

In [1]:

```
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
```

2) Lendo os Dados e Transformando em Dataframes

In [2]:

```
treino = pd.read_csv("training.csv")
teste = pd.read_csv("testing.csv")
reduce = pd.read_csv("reducedSet.csv")

target_treino = pd.get_dummies(treino["Class"])
target_treino.drop('unsuccessful', axis=1, inplace=True)
target_teste = pd.get_dummies(teste["Class"])
target_teste.drop('unsuccessful', axis=1, inplace=True)

treino = treino[reduce['x']]
teste = teste[reduce['x']]
```

4) Criando o Modelo: Logístico

In [3]:

```
logmodel = LogisticRegression()
logmodel.fit(treino, target_treino.values.ravel())
```

Out[3]:

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                    intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
                    penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
                    verbose=0, warm_start=False)
```

5) Teste do Modelo Logístico

In [4]:

```
predictions = logmodel.predict(teste)
```

a) Precisão

In [5]:

```
print(classification_report(target_teste, predictions))
```

	precision	recall	f1-score	support
0	0.88	0.86	0.87	329
1	0.77	0.79	0.78	189
avg / total	0.84	0.84	0.84	518

b) Matriz de Confusão

In [6]:

```
print(confusion_matrix(target_teste,predictions))
```

```
[[283  46]
 [ 39 150]]
```

6) Criando o Modelo: Rede Neural

In [7]:

```
clf=MLPClassifier(activation = "logistic", alpha=0.0, batch_size =50,hidden_layer_sizes=(126,),
                  learning_rate='constant',learning_rate_init=0.009,max_iter=200, momentum=0.0)
```

In [8]:

```
clf.fit(treino,target_treino.values.ravel())
```

Out[8]:

```
MLPClassifier(activation='logistic', alpha=0.0, batch_size=50, beta_1=0.9,
              beta_2=0.999, early_stopping=False, epsilon=1e-08,
              hidden_layer_sizes=(126,), learning_rate='constant',
              learning_rate_init=0.009, max_iter=200, momentum=0.0,
              nesterovs_momentum=True, power_t=0.5, random_state=None,
              shuffle=True, solver='adam', tol=0.0001, validation_fraction=0.1,
              verbose=False, warm_start=False)
```

7) Teste da Rede Neural

In [9]:

```
pred = clf.predict(teste)
```

a) Precisão

In [10]:

```
print(classification_report(target_teste,pred))
```

	precision	recall	f1-score	support
0	0.92	0.85	0.88	329
1	0.77	0.87	0.81	189
avg / total	0.86	0.86	0.86	518

b) Matriz de Confusão

In [11]:

```
print(confusion_matrix(target_teste,pred))
```

```
[[279  50]
 [ 25 164]]
```