

# Regressão Logística



**Prof Luiz Barboza**  
**luiz@barboza.me.uk**  
**@profluizbarboza**

+

+

.

.

.

.

.



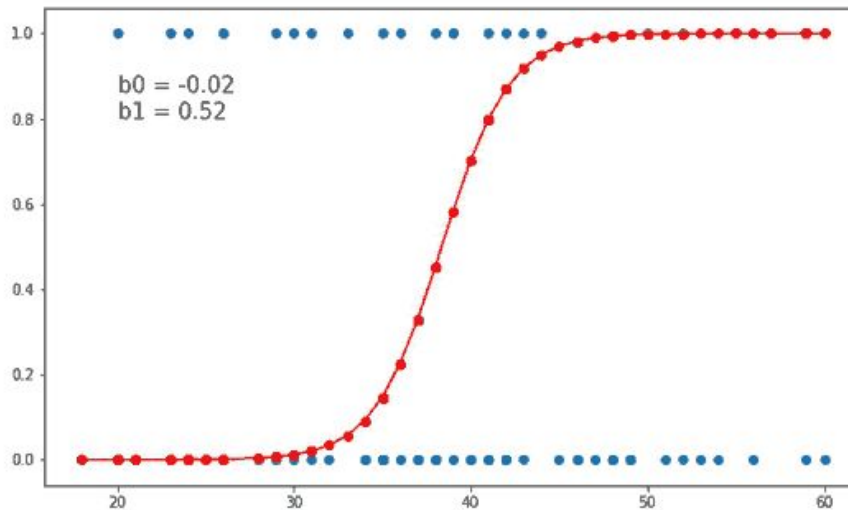
.

.

.

.

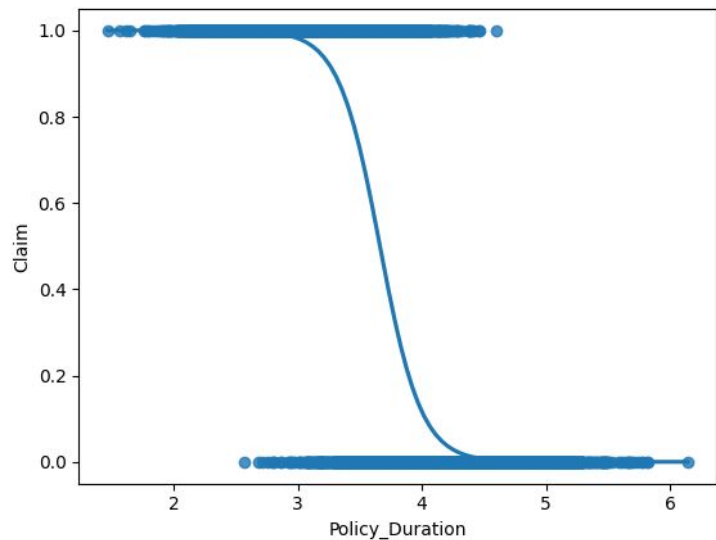
## Função Logística



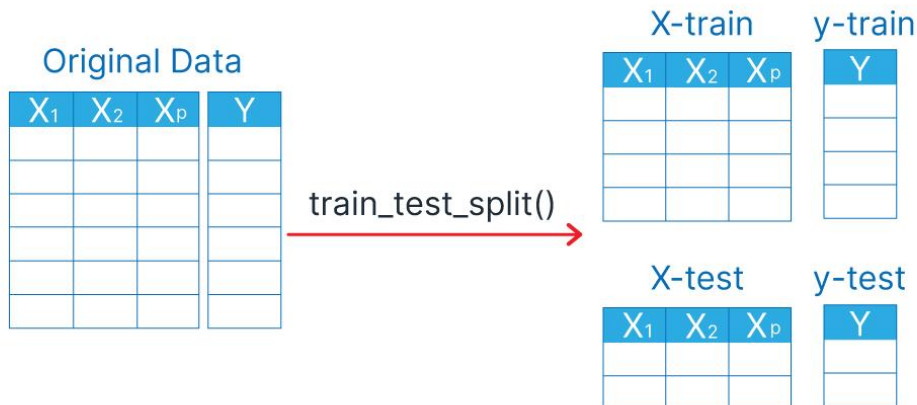
$$\bar{y}_i = p = \frac{1}{1 + e^{-(b_0 + b_1 x_i)}}$$

## Regressão Logística: Função Logística

```
import seaborn as sns  
sns.regplot(x="Policy_Duration",  
y="Claim", data=df, logistic=True)
```



## Separação Treino Teste



```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test =
train_test_split(df[['Premium', 'Policy_Duration']], df['Claim'], test_size=0.2)
X_train.head(3)
```

## Regressão Logística

```
import sklearn.linear_model as skl_lm
rlog = skl_lm.LogisticRegression()
rlog.fit(X_train,y_train)

train_accuracy= rlog.score(X_train,y_train)*100
train_accuracy
```



## Regressão Logística: Previsões

`rlog.predict(X_test)` usa um modelo de regressão logística treinado (`rlog`) para prever a probabilidade de um certo evento (por exemplo, um sinistro) para novos pontos de dados (`X_test`). A variável `preds` armazena as probabilidades previstas.

```
preds = rlog.predict(X_test)
preds
```

```
array([1, 1, 0, ..., 1, 1, 0])
```

# Accuracy

```
sum(y_test==preds)/len(preds) *100
```

```
from sklearn.metrics import accuracy_score  
test_accuracy = accuracy_score(y_test,preds)*100  
test_accuracy
```



## Matriz de Confusão

```

from sklearn.metrics import confusion_matrix
import seaborn as sns

cm = confusion_matrix(y_test, preds)

print(cm)

sns.heatmap(cm, annot=True, fmt="d")

```

		Predicted	
		0	1
Actual	0	TN	FP
	1	FN	TP

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$



## Classification Report: Precision e Recall

```
from sklearn.metrics import classification_report
print(classification_report(y_test,preds))
```

	precision	recall	f1-score	support
0	0.92	0.92	0.92	1011
1	0.92	0.92	0.92	989
accuracy			0.92	2000
macro avg	0.92	0.92	0.92	2000
weighted avg	0.92	0.92	0.92	2000

## Matriz de Confusão

		<u>Actual</u>		
		True	False	
<u>Predicted</u>	True	True Positive (TP)	False Positive (FP)	Precision: $\frac{TP}{TP + FP}$
	False	False Negative (FN)	True Negative (TN)	Negative Predictive Value: $\frac{TN}{TN + FN}$
Sensitivity:		Specificity:		Accuracy:
$\frac{TP}{TP + FN}$		$\frac{TN}{TN + FP}$		$\frac{TP + TN}{TP + TN + FP + FN}$

- treino: <https://raw.githubusercontent.com/lcbjrrr/data/main/RiscoCredito%20-%20okk.csv>  
test: <https://raw.githubusercontent.com/lcbjrrr/data/main/RiscoCredito%20-%20prever2.csv>

A stylized graphic of a dark suit jacket and a dark tie against a red background. The tie is centered and extends down the middle of the frame.

# MAD MEN

Intuición · Creatividad · Appeal

A stylized graphic of a red shirt collar and a red tie against a dark purple background. The tie is centered and extends down the middle of the frame.

# MATH MEN

Analytics · Resultados · Lógica



## ATIVIDADE: Reg Logística

- Escolha uma base de dados no <https://www.kaggle.com/datasets>, e se familiarize com sua base
- Procure realizar a previsão (inferência) de uma variável categórica através de um Reg Logística. Se certifique de medir seus níveis de assertividade. Esteja a vontade a realizar mais um hiperparâmetro (número de vizinhos) de um e compará-los
- Não esqueça de junto com seus códigos realizar suas análises/conclusões (use o botão de +Texto).