Ciência de Dados
(Big Data Processing and Analytics)

Big Data Analytics – Mineração e Análise de Dados

Professor curador
Prof. Dr. Rogério de Oliveira

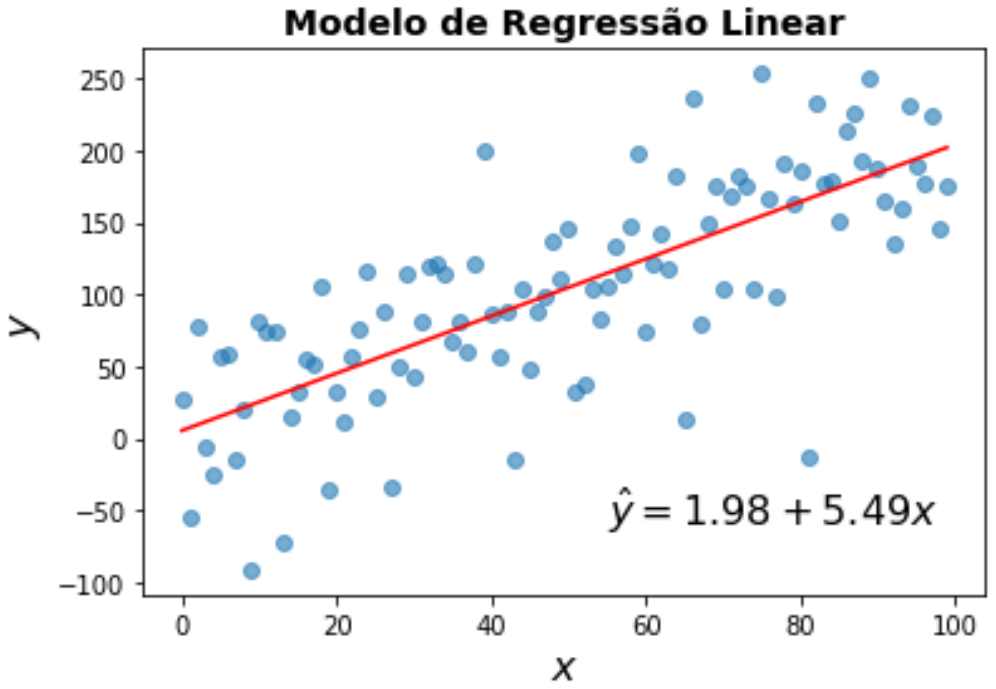
TRILHA 2
Regressão e Classificação: Regressão Linear e Logística

Parte A

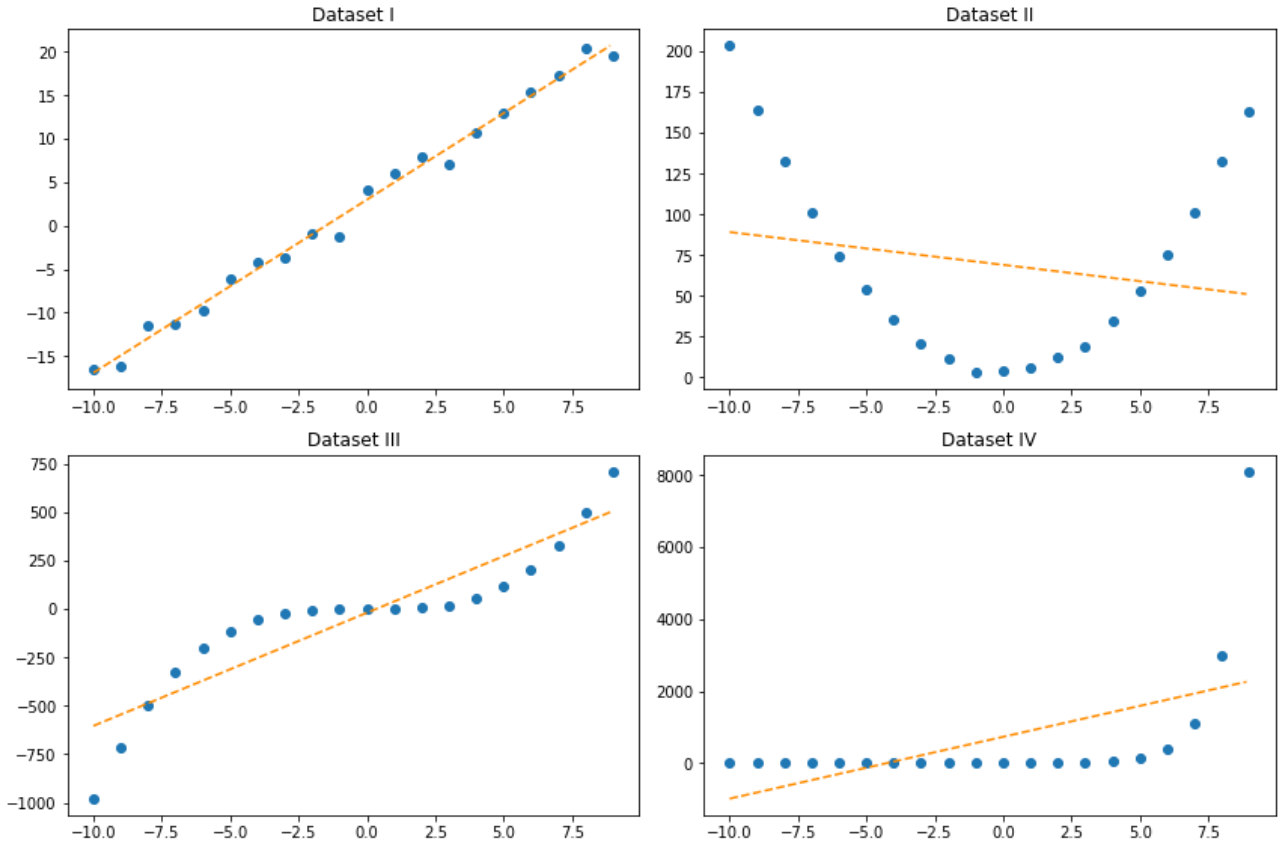
Regressão Linear Simples

$$b = \frac{cov(x, y)}{var(x)}$$

$$a = \bar{y} - b\bar{x}$$



Coeficiente de Determinação



$$R^2 = 1 - \frac{E_{res}}{E_{total}}$$

$$E_{res} = \sum (y_i - \hat{y}_i)^2$$

$$E_{total} = \sum (y_i - \bar{y})^2$$

Statsmodels ols

```
model = sm.ols(formula="Price ~ Passengers +    Length +
             Width     + Turncircle + Luggageroom + \
             Weight + Horsepower + EngineSize + \
             RPM + Wheelbase ", data=df)
result = model.fit()
print(result.summary())
```

OLS Regression Results

Dep. Variable:	Price	R-squared:	0.727
Model:	OLS	Adj. R-squared:	0.689
Method:	Least Squares	F-statistic:	18.94
Date:	Sat, 20 Nov 2021	Prob (F-statistic):	2.32e-16
Time:	17:22:57	Log-Likelihood:	-251.04
No. Observations:	82	AIC:	524.1
Df Residuals:	71	BIC:	550.6
Df Model:	10		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	53.1792	28.749	1.850	0.069	-4.146	110.504
Passengers	-0.3768	1.317	-0.286	0.776	-3.004	2.250
Length	0.0090	0.130	0.069	0.945	-0.251	0.269
Wheelbase	0.6436	0.280	2.301	0.024	0.086	1.201
Width	-1.5020	0.457	-3.289	0.002	-2.413	-0.591
Turncircle	-0.5811	0.374	-1.555	0.124	-1.326	0.164
Luggageroom	0.0801	0.349	0.230	0.819	-0.615	0.776
Weight	0.0066	0.005	1.366	0.176	-0.003	0.016
Horsepower	0.1430	0.046	3.123	0.003	0.052	0.234
EngineSize	-0.7457	2.409	-0.310	0.758	-5.549	4.057
RPM	-0.0025	0.002	-1.081	0.283	-0.007	0.002

Omnibus:	28.002	Durbin-Watson:	1.869
Prob(Omnibus):	0.000	Jarque-Bera (JB):	81.343
Skew:	1.048	Prob(JB):	2.17e-18
Kurtosis:	7.406	Cond. No.	2.88e+05

TRILHA 2
Regressão e Classificação: Regressão Linear e Logística

Parte B

Regressão Linear vs Logística

Tarefas de Aprendizado Supervisionado

Breast Cancer Data

diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean
M	15.340	14.26	102.50	704.4
B	12.880	28.92	82.50	514.3
M	17.080	27.15	111.20	930.9
B	16.140	14.86	104.30	800.0
M	13.480	20.82	88.40	559.2
B	14.470	24.99	95.81	656.4
B	12.490	16.85	79.19	481.6
M	23.210	26.97	153.50	1670.0
B	11.620	18.18	76.38	408.8
B	9.787	19.94	62.11	294.5
M	21.750	20.99	147.30	1491.0
B	10.800	21.98	68.79	359.9
M	25.730	17.46	174.20	2010.0
B	11.870	21.54	76.83	432.0
B	7.691	25.44	48.34	170.4

y

X

y

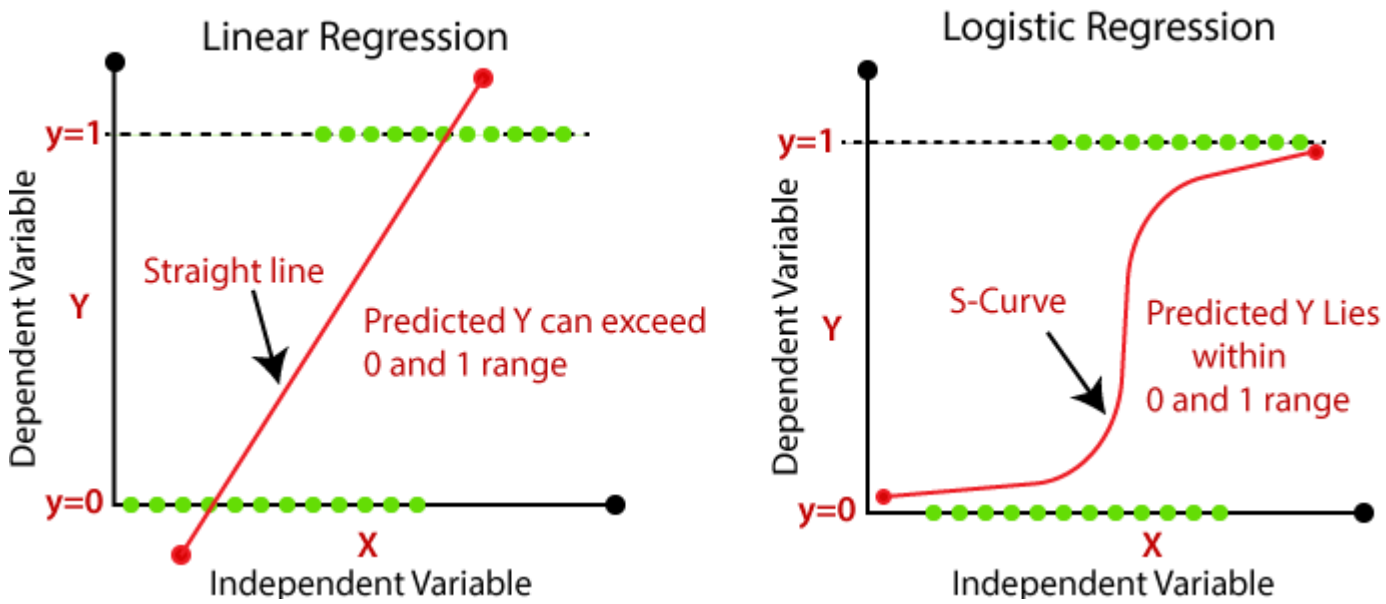
Classificação

Árvores de Decisão
Regressão Logística
K-Vizinhos mais Próximos
Support Vector Machines

Regressão

Regressão Linear
Regressão Polinomial
Modelos Neurais para Regressão

Regressão Linear vs Logística

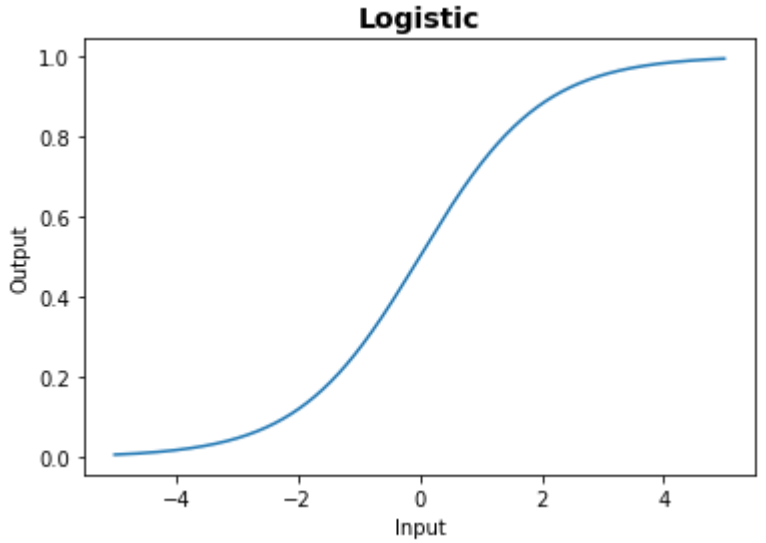


Regressão Logística

$$log(\frac{p}{1-p}) = a_0 + a_1x_1 + \ldots + a_nx_n$$

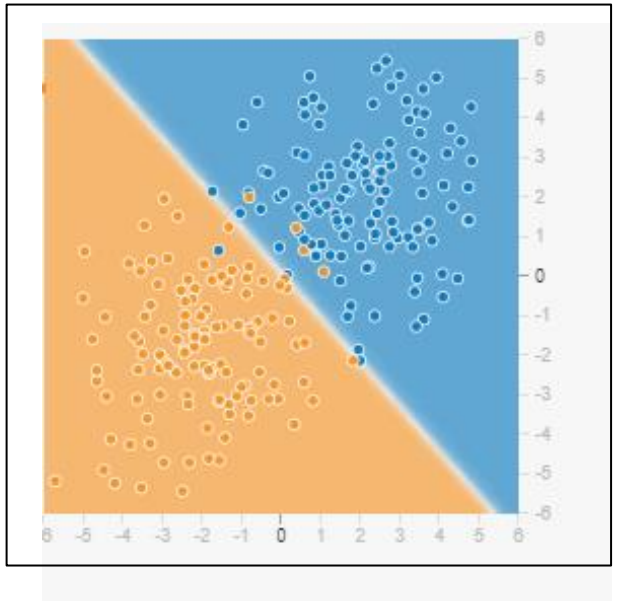
$$p = 1/(1 + e^{-(a_0 + a_1x_1 + \ldots + a_nx_n)})$$

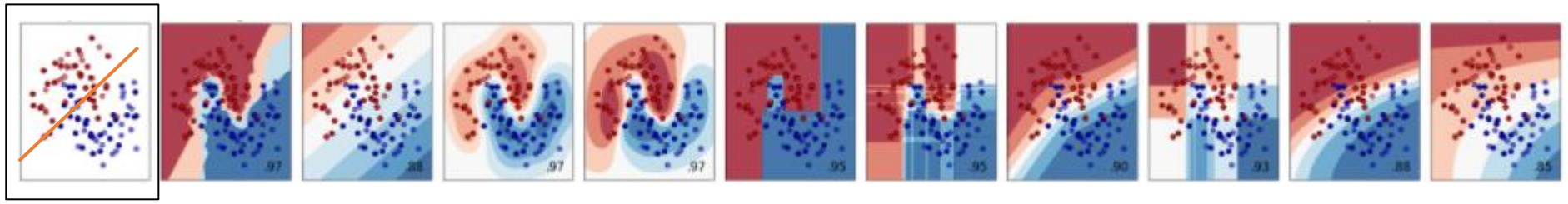
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



Regressão Logística

- Separador Linear
- Classificação Binária
  (classes dicotômicas)





Regressão Logística Scikit-Learn

```
X_train = df[['EngineSize', 'Horsepower', 'RPM', 'Price', 'Weight']]
y_train = df['Manual']
```

```
from sklearn.linear_model import LogisticRegression  # para configurar o modelo..

# criando o modelo
logreg = LogisticRegression()

# treinando o modelo
logreg.fit(X_train,y_train)
```

```
y_pred = logreg.predict(X_train)

df['Manual_predict'] = y_pred

df[['Manual','Manual_predict']]
```

	Manual	Manual_predict
1	1	1
2	1	1
3	1	1
4	1	1
5	1	1
...	...	...
89	1	0
90	1	1
91	1	1
92	1	1
93	1	1

Universidade Presbiteriana
Mackenzie