Linear Models for Classification and Regression

Leonardo Maglanoc

TUM

June 6, 2021

Table of Contents

Why use Linear Models?

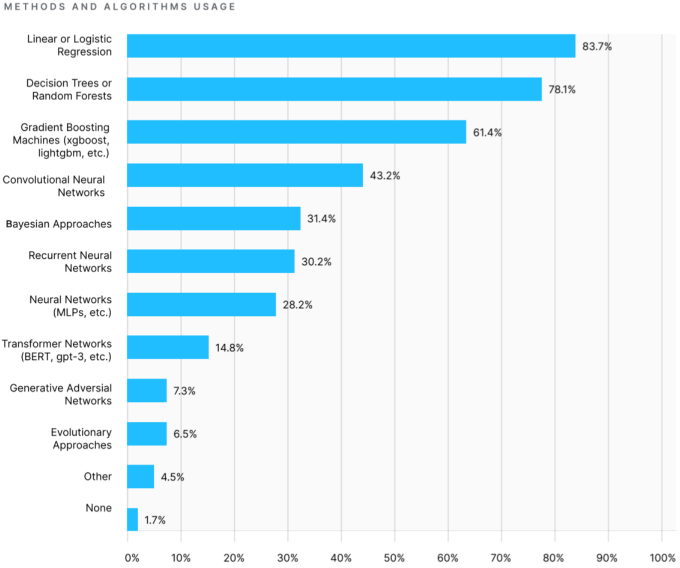


Figure: adapted from https://www.kaggle.com/kaggle-survey-2020

Regression vs Classification

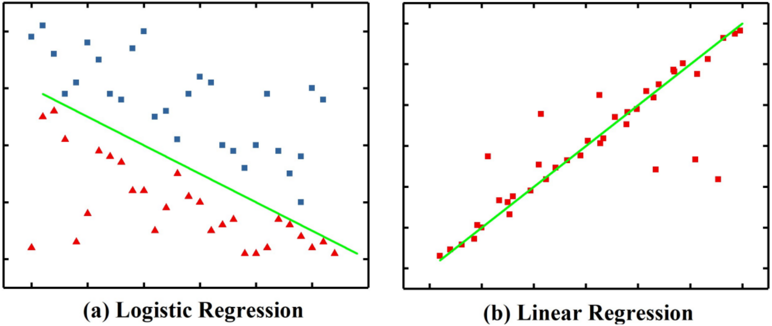


Figure: left: Classification, right: Regression (adapted from [1])

Idea of Linear Models:

$$a_1 v_1 + a_2 v_2 + \cdots + a_n v_n$$

Table of Contents

Table of Contents

Model

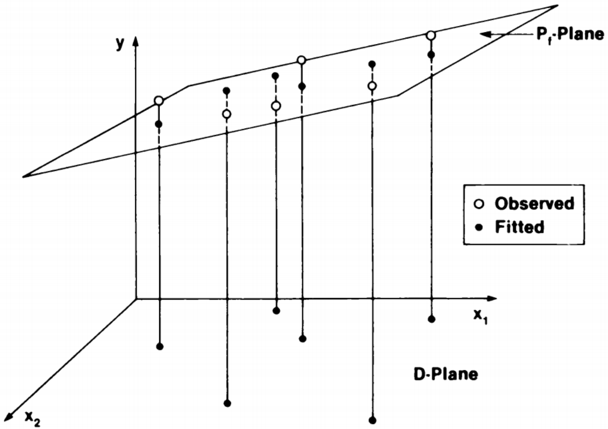


Figure: geometric representation of a regression plane (adapted from [2])

Mathematical Description

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix} + \begin{pmatrix} \epsilon_0 \\ \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix}$$

minimize error via Least Squares

$$\arg\min_{\boldsymbol{\beta}} \left[(\boldsymbol{y} - X\boldsymbol{\beta})^T (\boldsymbol{y} - X\boldsymbol{\beta}) \right]$$

Assumptions and Goodness of Fit

Assumption:

- input parameters are (linearly) **independent** from each other [3]
- errors ϵ_i have **Normal Distribution** with zero mean and constant variance

Goodness of Fit:

- Mean Squared Error
- R^2
- F-Test (Hypothesis Test)

Case Study with Linear Regression

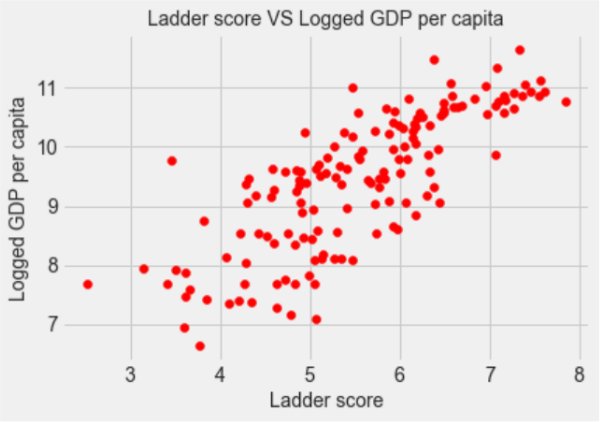
show Python-Notebook!
Library: Statsmodels with Python
`https://worldhappiness.report/faq/`

Case Study with Linear Regression

	Country name	Ladder score	Logged GDP per capita	Social support	Healthy life expectancy
0	Finland	7.842	10.775	0.954	72.0
1	Denmark	7.620	10.933	0.954	72.7
2	Switzerland	7.571	11.117	0.942	74.4
3	Iceland	7.554	10.878	0.983	73.0
4	Netherlands	7.464	10.932	0.942	72.4

Figure: excerpt of happines-dataset

Case Study with Linear Regression



Case Study with Linear Regression

```
                            OLS Regression Results
==============================================================================
Dep. Variable:           Ladder score   R-squared:                       0.756
Model:                            OLS   Adj. R-squared:                  0.746
Method:                 Least Squares   F-statistic:                     73.27
Date:                Thu, 03 Jun 2021   Prob (F-statistic):           5.06e-41
Time:                        12:31:21   Log-Likelihood:                -116.50
No. Observations:                 149   AIC:                             247.0
Df Residuals:                     142   BIC:                             268.0
Df Model:                           6
Covariance Type:            nonrobust
================================================================================================
                                   coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
const                           -2.2372      0.630     -3.548      0.001      -3.484      -0.991
Logged GDP per capita            0.2795      0.087      3.219      0.002       0.108       0.451
Social support                   2.4762      0.668      3.706      0.000       1.155       3.797
Healthy life expectancy          0.0303      0.013      2.274      0.024       0.004       0.057
Freedom to make life choices     2.0105      0.495      4.063      0.000       1.032       2.989
Generosity                       0.3644      0.321      1.134      0.259      -0.271       0.999
Perceptions of corruption       -0.6051      0.291     -2.083      0.039      -1.179      -0.031
==============================================================================
Omnibus:                       12.908   Durbin-Watson:                   1.614
Prob(Omnibus):                  0.002   Jarque-Bera (JB):               13.688
Skew:                          -0.667   Prob(JB):                      0.00107
Kurtosis:                       3.650   Cond. No.                     1.15e+03
==============================================================================
```

Table of Contents

Reuse

- Can we somehow reuse/reframe previously presented Linear Regression Methods to solve a Classification problem?
- Reason why it's called Logistic **Regression**

Model

$$\log\left(\frac{1}{1-p}\right) = X\beta$$

$$\Rightarrow P(y=1|\boldsymbol{x}) = \frac{1}{1+e^{-(\boldsymbol{\beta}^T\boldsymbol{x})}}$$

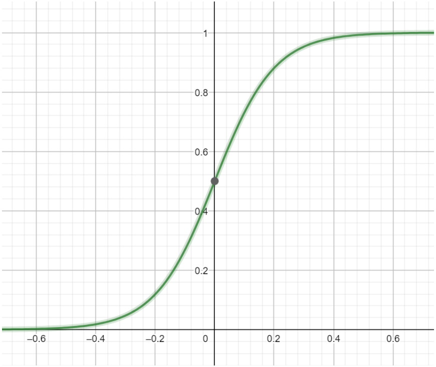


Figure: logistic function

Case Study with Logistic Regression

show Python-Notebook!

Library: Scikit-Learn with Python

`https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29`

Case Study with Logistic Regression

	diagnosis	texture_mean	perimeter_mean	smoothness_mean	compactness_mean	symmetry_mean
564	M	22.39	142.00	0.11100	0.11590	0.1726
565	M	28.25	131.20	0.09780	0.10340	0.1752
566	M	28.08	108.30	0.08455	0.10230	0.1590
567	M	29.33	140.10	0.11780	0.27700	0.2397
568	B	24.54	47.92	0.05263	0.04362	0.1587

Figure: excerpt of cancer-dataset

Case Study with Logistic Regression

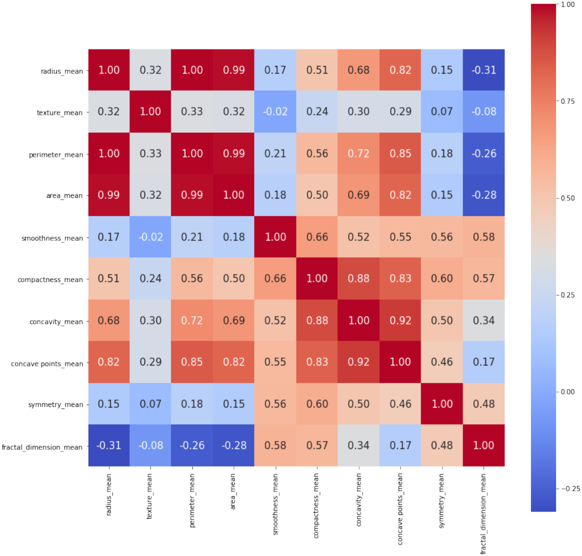


Figure: correlation-matrix

Case Study with Logistic Regression

Metrics	Values
Classification Accuracy	91.2%
MSE	0.0877

Table: Metrics for Logistic Regression Model

Table of Contents

Drawback of Linear Models

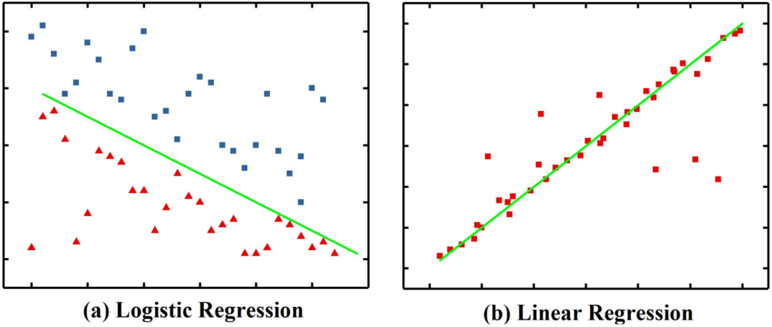


Figure: left: Classification, right: Regression (adapted from [1])

Summary

- Linear Models are the most often used Data Analysis Algorithms (in Kaggle)
- should be used before applying more complex Algorithms

Questions

Any Questions? :)

G. Y. Y. Q. Xianjin Fang, Fangchao Yu, "Regression analysis with differential privacy preserving," *IEEE Access*, vol. 7, pp. 129 353–129 361, 2019.

J. Mandel, "Use of the singular value decomposition in regression analysis," *The American Statistician*, vol. 36 No.1, pp. 15–24, 1982.

X. G. S. Xin Yan, *Linear Regression Analysis*. World Scientific, 2009.

J. F. Trevor Hastie, Robert Tibshirani, *The Elements of Statistical Learning*. Springer, 2008.

K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0893608089900208

- "Neural net blogpost," https://sebastianraschka.com/faq/docs/logisticregr-neuralnet.html, accessed: 2021-22-05.
- C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- S. Seabold and J. Perktold, "statsmodels: Econometric and statistical modeling with python," in *9th Python in Science Conference*, 2010.
- J. H. M. Daniel Jurafsky, *Speech and Language Processing*. Third Edition Draft, 2020.

Everything below here are just extra slides

[4] [5] [3] [6] [7] [1] [8] [9] [10] [2]

Computation via Singular Value Decomposition

$$\boldsymbol{y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon} = U\Sigma V^T\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

$$\Rightarrow \boldsymbol{\beta} = V\Sigma^{-1}U^T y$$

$O(n_{\text{samples}} \cdot n_{\text{features}}^2)$ Scikit-Learn [8]

- analogon of Eigendecomposition
- U ($n \times n$) and V ($p \times p$) as orthogonal matrices
- Σ ($p \times p$) consists only of diagonal entries known as the singular values of X
- check colinearities with singular values

Problem Setting

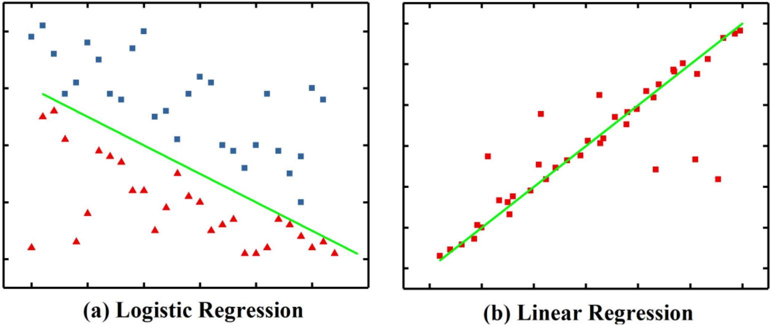


Figure: left: Classification, right: Regression (adapted from [1])

Linear Regression

$$y = X\boldsymbol{\beta}$$

Transformation Function

$$y = \log\left(\frac{1}{1-p}\right)$$

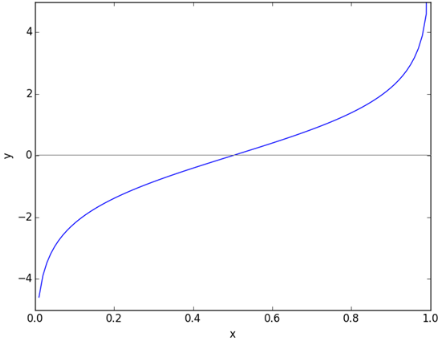


Figure: transformation function from $(0, 1)$ to $\mathbb{R}$

Computation

minimization problem:

$$\arg\min_{\boldsymbol{\beta}} \frac{1}{m} \sum_{i=1}^{m} e(f(\boldsymbol{x}^{(i)}; \boldsymbol{\beta})),$$

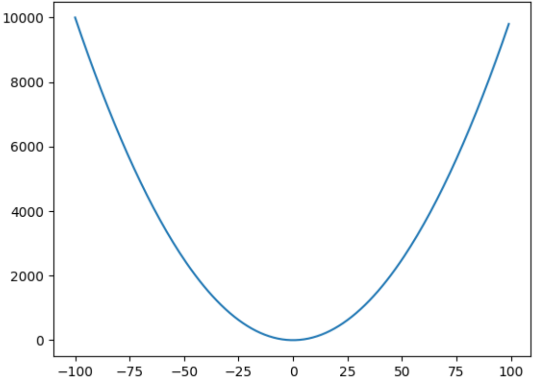


Figure: convex error function

Goodness of Fit

- split dataset into training and testing dataset
- **Classification Accuracy**

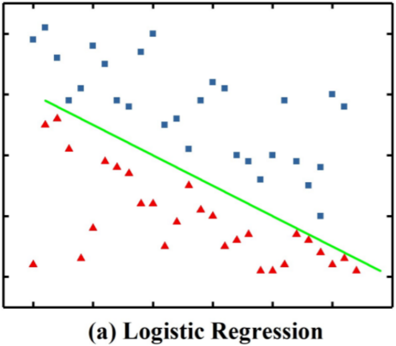


Figure: Logistic Regression

Polynomial Regression

$$X = \begin{pmatrix} 1 & x_1 & \cdots & x_1^p \\ 1 & x_2 & \cdots & x_2^p \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \cdots & x_n^p \end{pmatrix}$$

Logistic Regression as "Mini"-Neural-Network

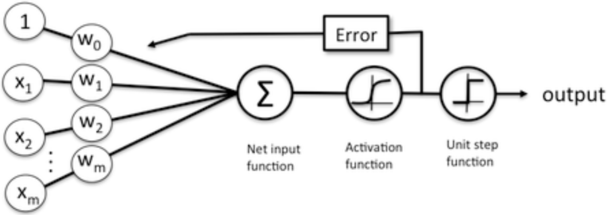


Figure: Logistic Regression as a Neural Net (adapted from [6])