CSE 634 - Data Mining Concepts and Techniques

STATISTICAL METHODS (REGRESSION)

Professor **Anita Wasilewska**

Contents

- Linear Regression

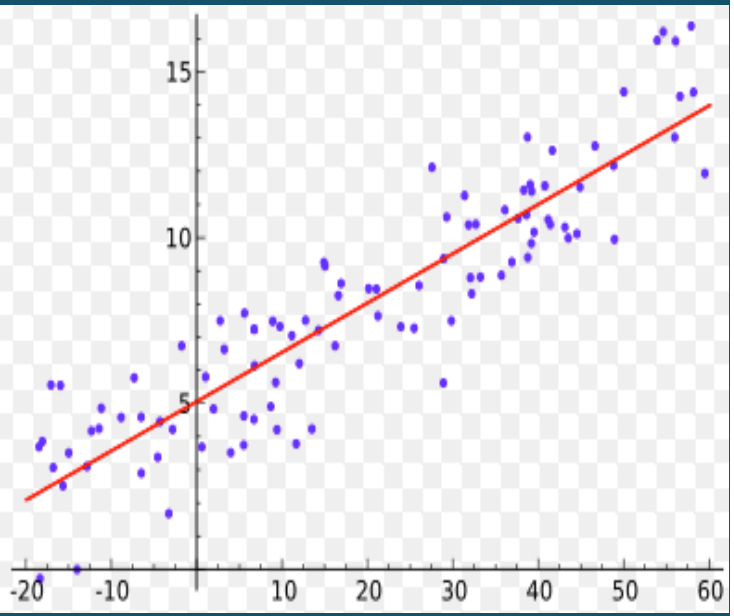
- Logistic Regression
- Bias and Variance in Regression
- Model Fit
- Methods to prevent Overfitting
- Regularization Methods (Lasso and Ridge regression)
- Research Paper

References

- **http://www3.cs.stonybrook.edu/~cse634/L2ch2preprocess.pdf - Lecture Slides**
- **http://www3.cs.stonybrook.edu/~cse634/L4ch6testing.pdf - Lecture Slides**
- www.en.wikipedia.org/wiki/Linear_regression
- www.stat.wmich.edu/s216/book/node127.html
- www.theanalysisfactor.com/assessing-the-fit-of-regression-models/
- www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/
- www.deeplearning4j.org/earlystopping
- www.cc.gatech.edu/~bboots3/CS4641-Fall2016/Lectures/Lecture3_1.pdf
- www.gerardnico.com/data_mining/shrinkage
- www3.cs.stonybrook.edu/~has/CSE545/Slides/6.11-11.pdf
- www.is.uni-freiburg.de/ressourcen/business-analytics/xx_regularization.pdf
- www.quora.com/How-would-you-describe-LASSO-regularization-in-laymens-terms
- www.mcser.org/journal/index.php/jesr/article/download/7722/7403

1

Regression Techniques



Linear Regression

Method of Prediction

What is Linear Regression?

- Commonly used **Predictive analysis** technique
- Linear approach for modelling the relationship between a dependent variable y and one or more independent variables denoted by X

Simple Linear Regression - 1 dependent variable, 1 independent variable

Multiple Linear Regression - 1 dependent variable, 2 or more independent variables.

https://en.wikipedia.org/wiki/Linear_regression

Simple Regression- Calculating the regression line

x	y	$x - \bar{x}$	$y - \bar{y}$	$(x - \bar{x})^2$	$(x - \bar{x})(y - \bar{y})$
1	2	-2	-2	4	4
2	4	-1	0	1	0
3	5	0	1	0	0
4	4	1	0	1	0
5	5	2	1	4	2
$\bar{x} = 3$	$\bar{y} = 4$			10	6

$$b_1 = \frac{\sum(x - \bar{x})(y - \bar{y})}{(x - \bar{x})^2} = \frac{6}{10} = 0.6$$

PREDICTED $y = b_0 + b_1 x$

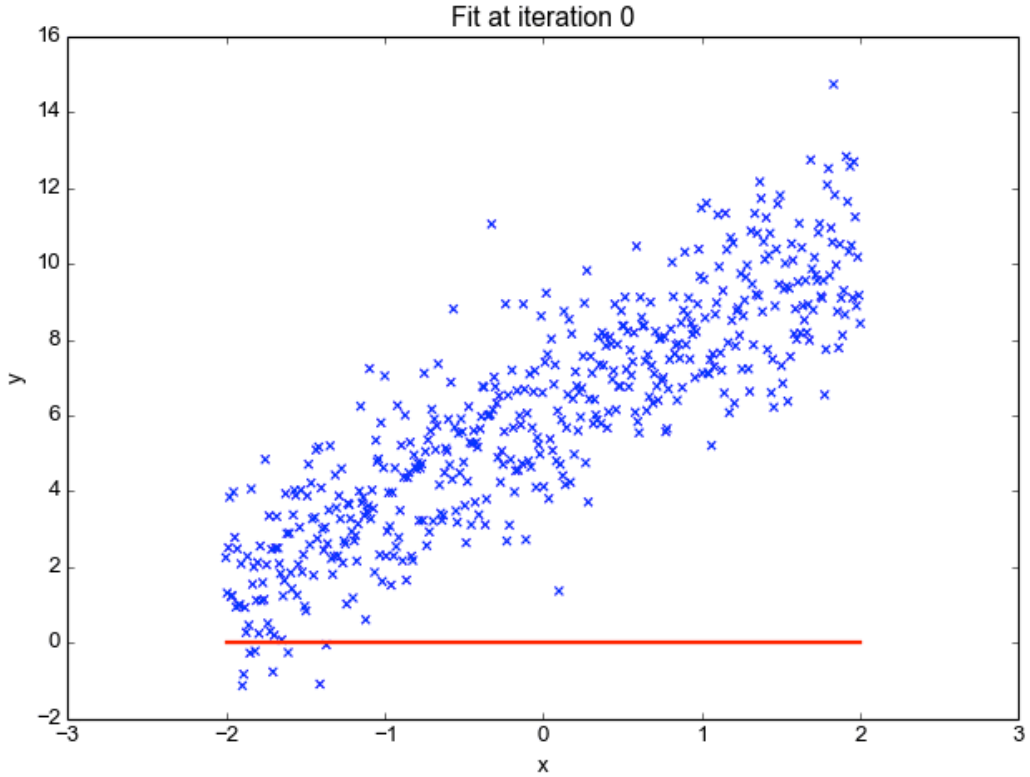
$b_1 = 6/10 = 0.6$

$b_0 = y - b_1 x \qquad b_0 = 2.2$

Therefore **$y = 2.2 + 0.6x$**

https://www.youtube.com/watch?v=JvS2triCgOY

Finding the Line of Best Fit



https://eli.thegreenplace.net/images/2016/regressionfit.gif

Assessing the fit of Regression Models

- Residual
  Y - Predicted Y
- SSE (The sum of the squared residuals)

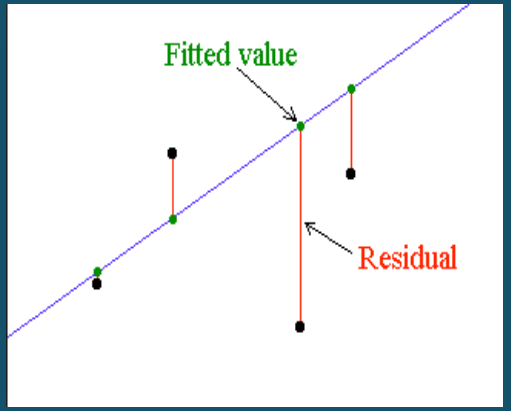
$$\text{SSE} = (Y_1 - \text{PREDICTED } Y_1)^2 + \cdots + (Y_n - \text{PREDICTED } Y_n)^2$$

- SST (Total sum of squares)

$$\text{SSTo} = (Y_1 - \overline{Y})^2 + \cdots + (Y_n - \overline{Y})^2$$

- SSR(Sum of Squares regression)

$$\text{SSR} = \text{SSTo} - \text{SSE}$$



http://www.stat.wmich.edu/s216/book/node127.html

Model Evaluation Error Metrics

1. R squared (Coefficient of Determination)

$$R^2 = \frac{\mathrm{SSR}}{\mathrm{SSTo}}$$

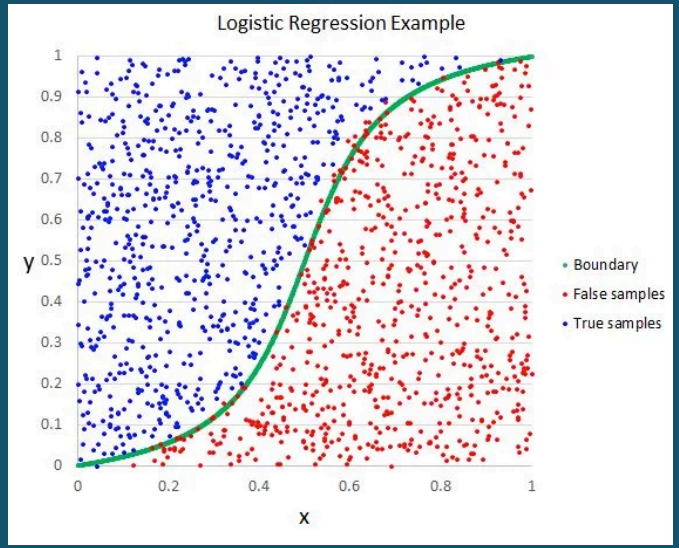
When R^2 value is 1, model perfectly fits the data.

1. Mean Squared Error

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

When MSE is 0, model perfectly fits the data.

https://www.theanalysisfactor.com/assessing-the-fit-of-regression-models/



Logistic Regression

Method of Classification

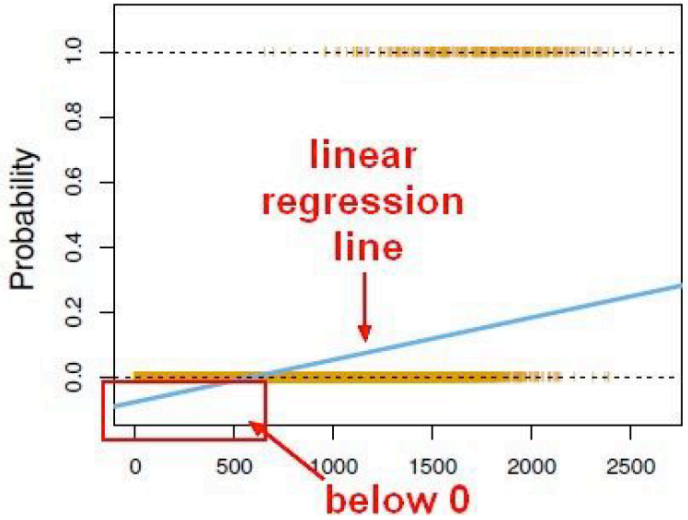
Background

We’re more often interested in making **categorical assignments**.

- Does this email belong in the **spam folder or the inbox**? **(Spam/Inbox)**
- How **likely is this customer to sign up** for subscription service? **(Yes/No)**
- Does a **patient have a particular disease**? **(Yes/No)**

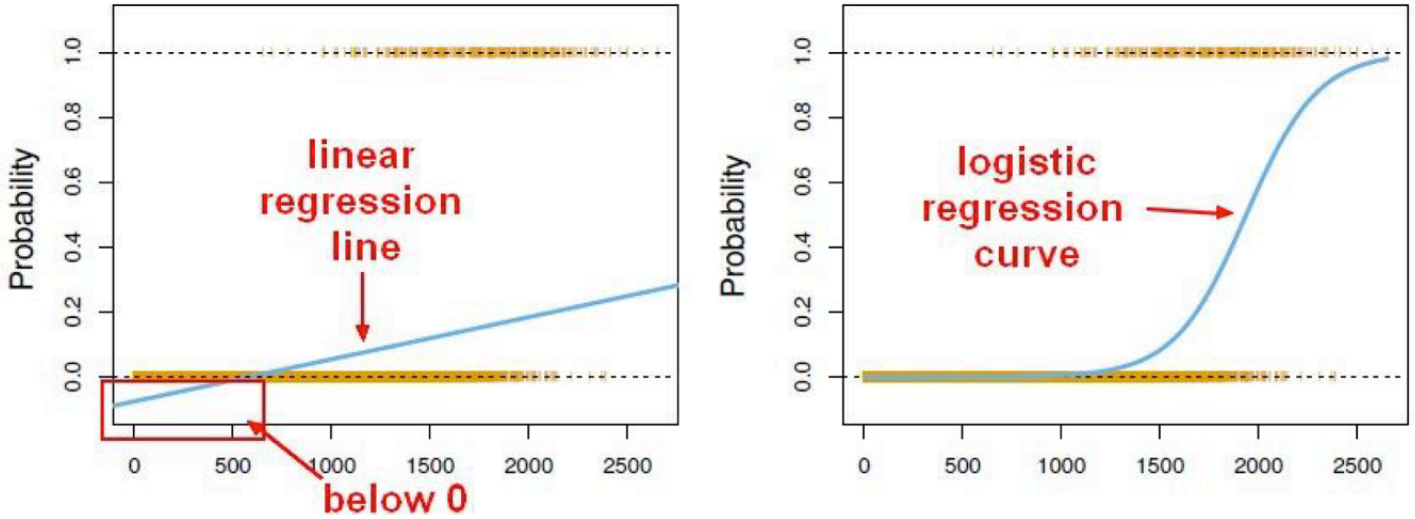
When we are interested in either assigning data-points to categories we call this task **classification**. The simplest kind of classification problem is **binary classification**, when there are only two classes.

The convention for binary classification is to have two classes 0 and 1. We can't use a normal linear regression model on binary groups. It won't lead to a good fit.



https://www.udemy.com/python-for-data-science-and-machine-learning-bootcamp/learn/v4/t/lecture/5733426

A regular linear model is a poor choice here because it can output values greater than 1 or less than 0. Therefore, we can transform our linear regression to a **logistic regression curve**.



https://www.udemy.com/python-for-data-science-and-machine-learning-bootcamp/learn/v4/t/lecture/5733426

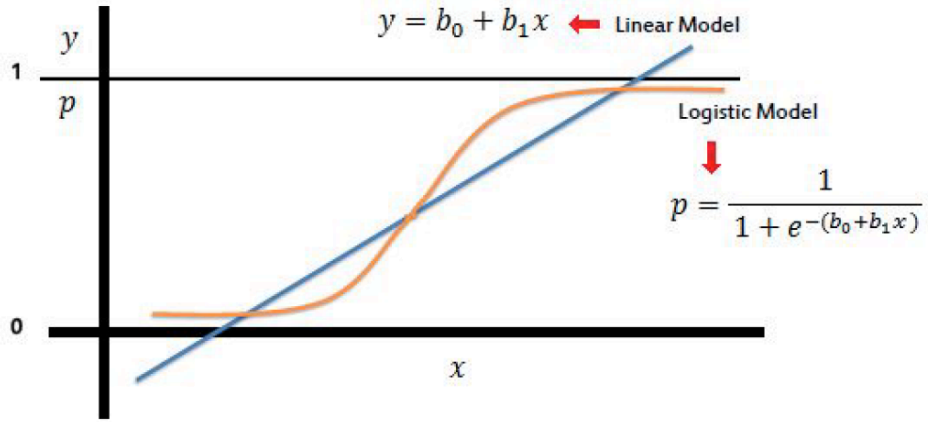
To build a correct classification model, we modify it slightly, by running the linear regression function through a **sigmoid activation function σ**.

$$\hat{y} = \sigma(\beta_0 + \beta_1 x)$$

The **sigmoid function** σ, sometimes called a squashing function or a logistic function - thus the name logistic regression - maps a real-valued input to the range 0 to 1. Specifically, it has the functional form:

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

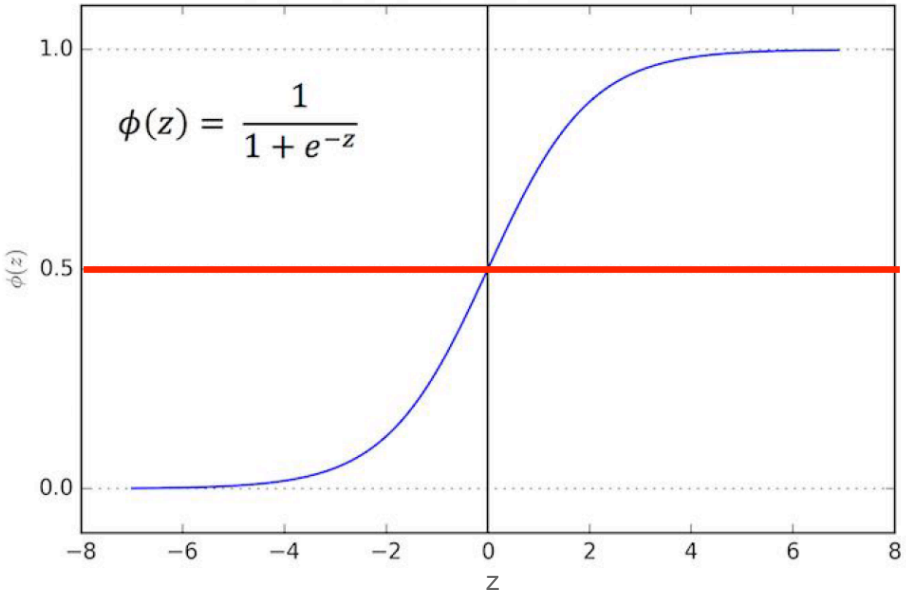
This means we can take our Linear Regression Solution and place it into the Sigmoid Function.



This results in a probability from 0 to 1 of belonging in a class.

https://www.udemy.com/python-for-data-science-and-machine-learning-bootcamp/learn/v4/t/lecture/5733426

We can set a cutoff point at 0.5, anything below it results in class 0, anything above is class 1.



We use the logistic function to output a value ranging from 0 to 1. Based off of this probability we assign a class.

https://www.udemy.com/python-for-data-science-and-machine-learning-bootcamp/learn/v4/t/lecture/5733426

Model Evaluation

The following steps should be taken to create a Logistic Regression model to classify a feature in a dataset:

1. Divide the data into training and testing datasets. We can do this manually by keeping a certain amount (let's say 30%) of the data as test data. One good way of running the model is by using **k-fold cross validation**.
2. After you train a logistic regression model on some training data, we evaluate our model's performance on test data.
3. We can use a **confusion matrix** and some metrics like **Precision, Recall, F-Score** to evaluate classification models.

Confusion Matrix

A confusion matrix is a table that is often used to **describe the performance of a classification model** (or "classifier") on a set of test data for which the true values are known.

We can infer that:

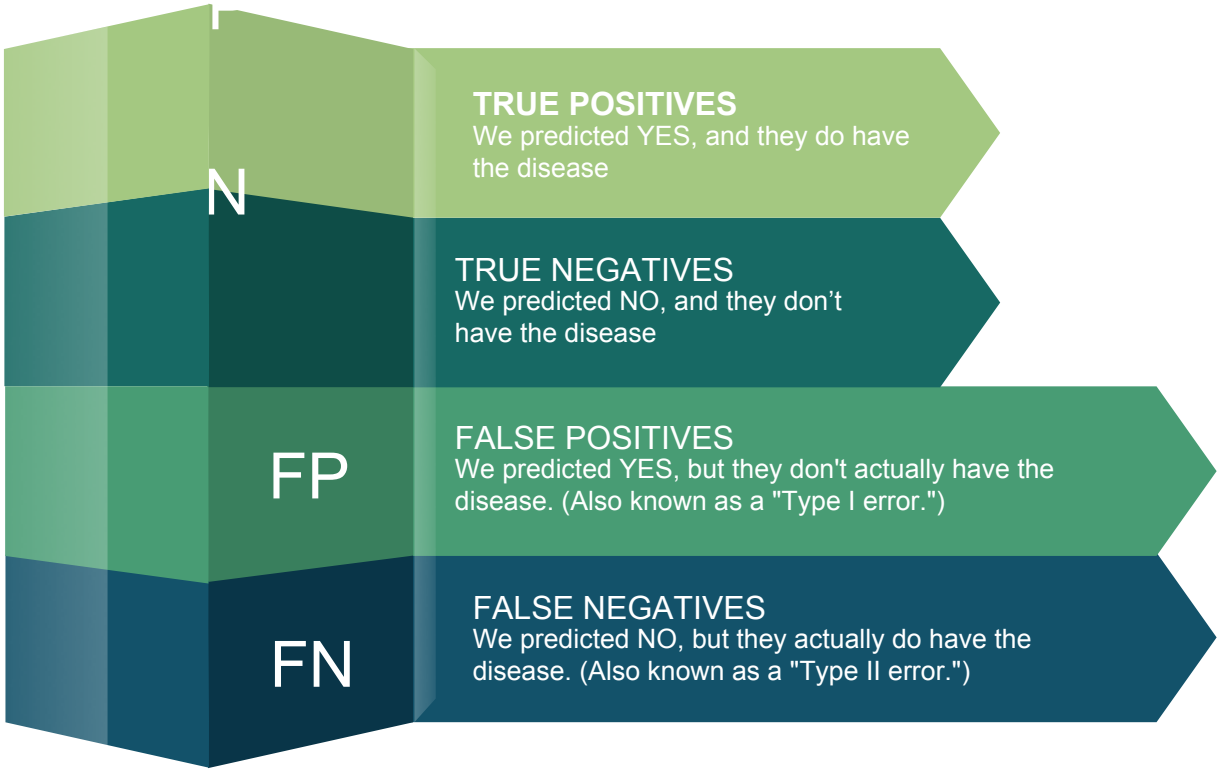
1. There are two possible predicted classes: "yes" and "no". (having disease or not having disease)
2. The classifier made a total of 165 predictions
3. Out of those 165 cases, the classifier predicted "yes" 110 times, and "no" 55 times.
4. In reality, 105 patients in the sample have the disease, and 60 patients do not.

Example:

n=165	**Predicted: NO**	**Predicted: YES**
Actual: NO	50	10
Actual: YES	5	100

http://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/

Let’s define the most basic terms:



Therefore, we can modify the confusion matrix as:

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

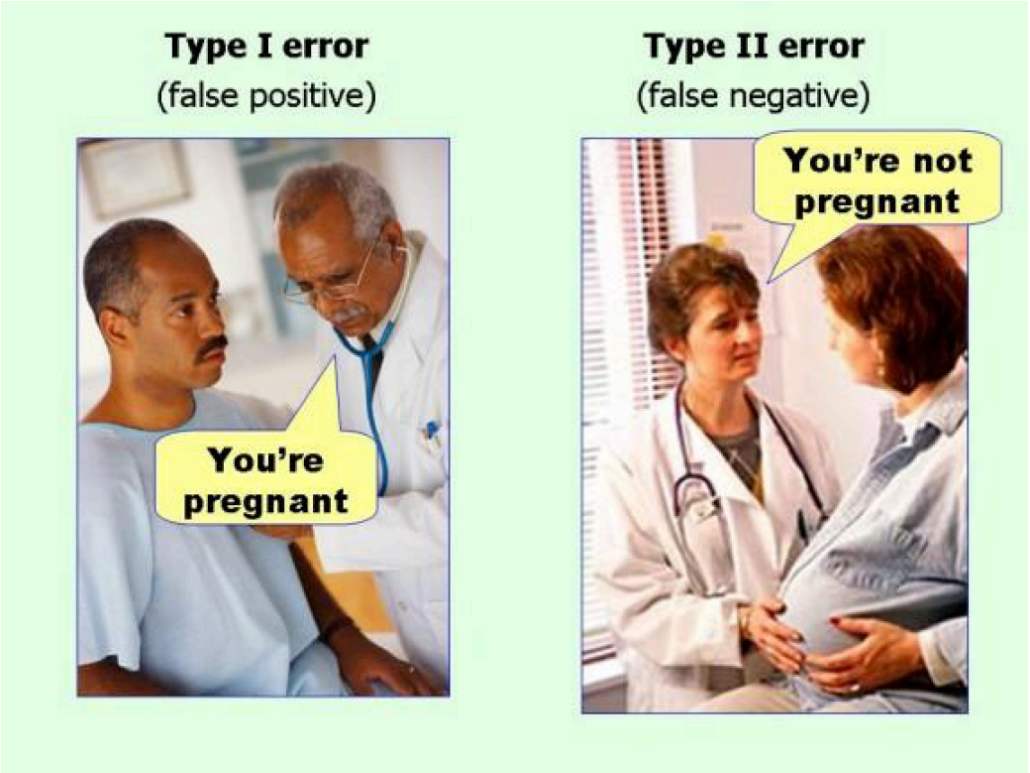
- **False Positive Rate:** When it's actually NO, how often does it predict yes?
    - FP/actual NO = 10/60 = 0.17
- **Precision:** When it predicts yes, how often is it correct?
    - TP/predicted YES = 100/110 = 0.91

Some important metrics can be calculated from the confusion matrix.

- **Accuracy:** Overall, how often is the classifier correct?
    - (TP+TN)/total = (100+50)/165 = 0.91
- **Misclassification Rate:** Overall, how often is it wrong?
    - (FP+FN)/total = (10+5)/165 = 0.09
    - equivalent to (1 - Accuracy)
    - also known as "**Error Rate**"
- **True Positive Rate:** When it's actually YES, how often does it predict YES?
    - TP/actual YES = 100/105 = 0.95
    - Also known as "**Sensitivity**" or "**Recall**"

http://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/

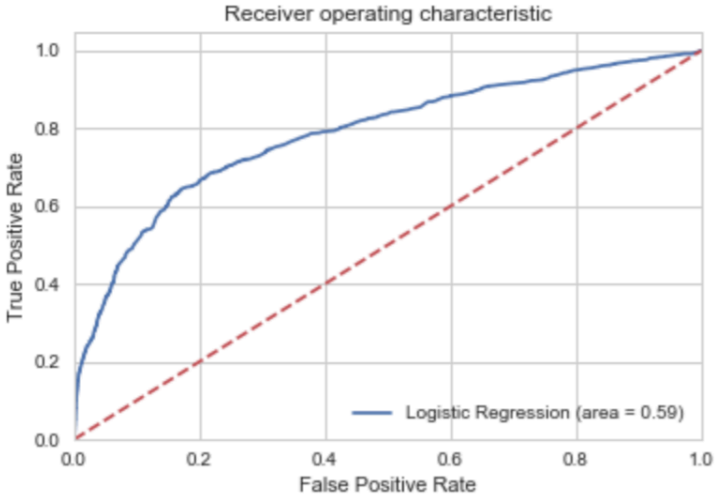
Type I and Type II error in a nutshell...



https://whatsthebigdata.files.wordpress.com/2017/01/errors.jpg

ROC Curve

The **receiver operating characteristic (ROC)** curve is another common tool used with binary classifiers. The dotted line represents the ROC curve of a purely random classifier; a good classifier stays as far away from that line as possible (toward the top-left corner).



https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8

Logistic Regression Example

We have an advertising data set, indicating whether or not a particular internet user clicked on an Advertisement on a company website. We will try to create a model that will predict **whether or not they will click on an ad (Class attribute)** based off the features of that user.

	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	Ad Topic Line	City	Male	Country	Timestamp	Clicked on Ad
0	68.95	35	61833.90	256.09	Cloned 5thgeneration orchestration	Wrightburgh	0	Tunisia	2016-03-27 00:53:11	0
1	80.23	31	68441.85	193.77	Monitored national standardization	West Jodi	1	Nauru	2016-04-04 01:39:02	0
2	69.47	26	59785.94	236.50	Organic bottom-line service-desk	Davidton	0	San Marino	2016-03-13 20:35:42	0
3	74.15	29	54806.18	245.89	Triple-buffered reciprocal time-frame	West Terrifurt	1	Italy	2016-01-10 02:31:19	0
4	68.37	35	73889.99	225.58	Robust logistical utilization	South Manuel	0	Iceland	2016-06-03 03:36:18	0

Classification Model:

```
from sklearn.model_selection import train_test_split
X = ad_data[['Daily Time Spent on Site', 'Age',
             'Area Income','Daily Internet Usage', 'Male']]
y = ad_data['Clicked on Ad']
from sklearn.linear_model import LogisticRegression
logmodel = LogisticRegression()
logmodel.fit(X_train,y_train)
predictions = logmodel.predict(X_test)
from sklearn.metrics import classification_report,confusion_matrix
print(classification_report(y_test,predictions))
print(confusion_matrix(y_test,predictions))
```

Confusion Matrix:

```
[[156   6]
 [ 24 144]]
```

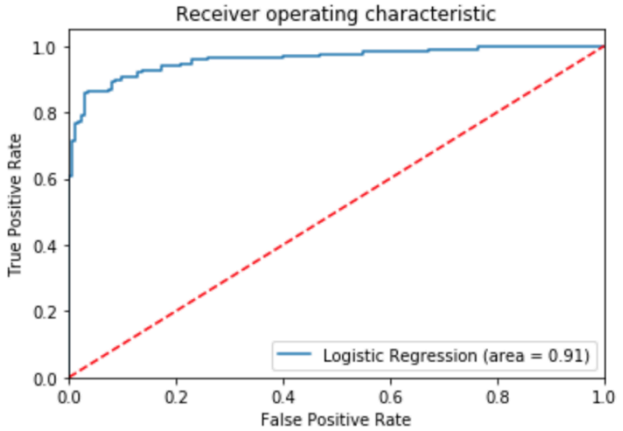
Classification Report

Class	Precision	Recall	F1-Score	Support
0	0.87	0.96	0.91	162
1	0.96	0.86	0.91	168
Average/Total	0.91	0.91	0.91	330

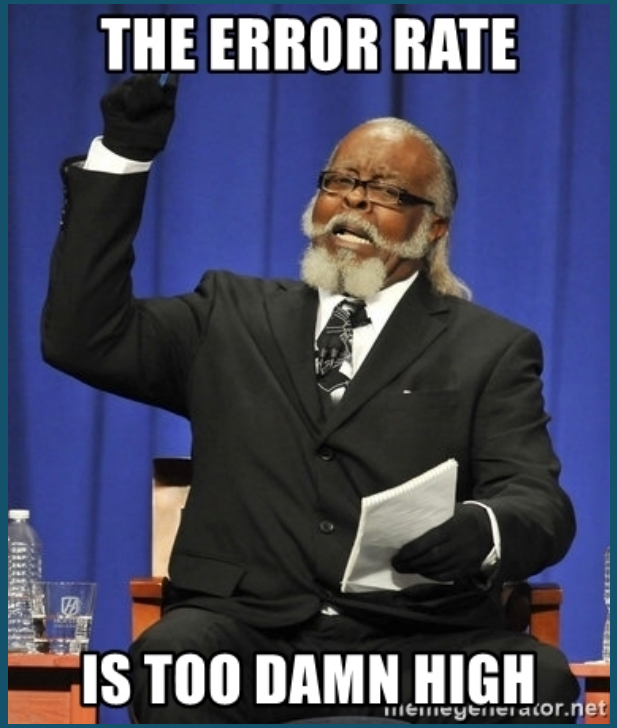
ROC Curve for the model:

```
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
logit_roc_auc = roc_auc_score(y_test, logmodel.predict(X_test))
fpr, tpr, thresholds = roc_curve(y_test, logmodel.predict_proba(X_test)[:,1])
plt.figure()
plt.plot(fpr, tpr, label='Logistic Regression (area = %0.2f)' % logit_roc_auc)
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic')
plt.legend(loc="lower right")
plt.show()
```

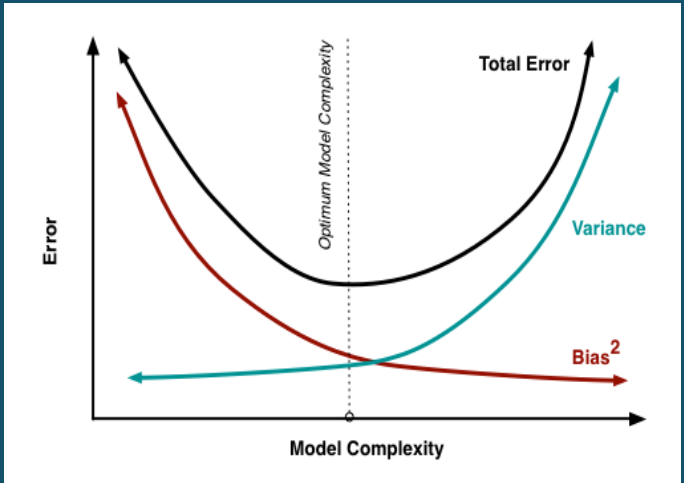
The ROC model is much far away from the dotted line. Thus, we've built a pretty good classifier.



What about the Error?



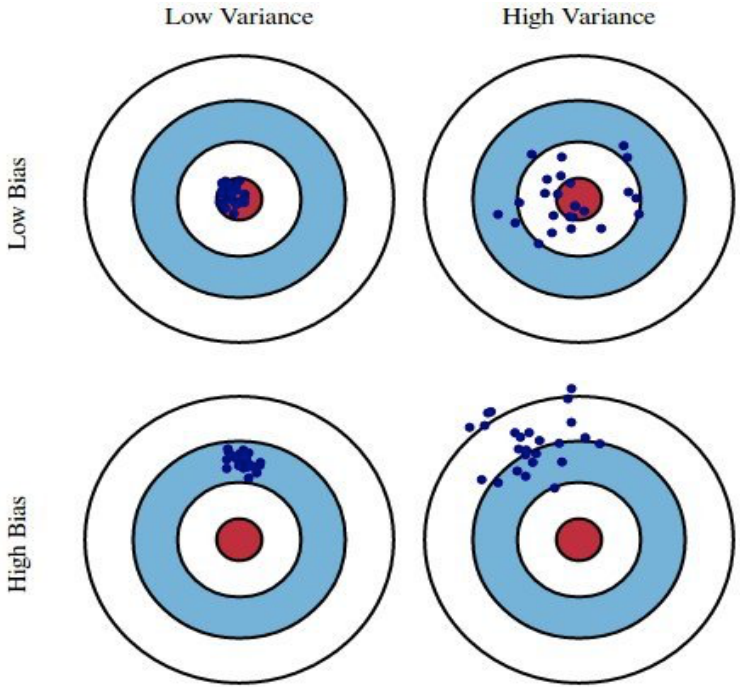
http://memecaptain.com/gend_image_pages/-1PsXg



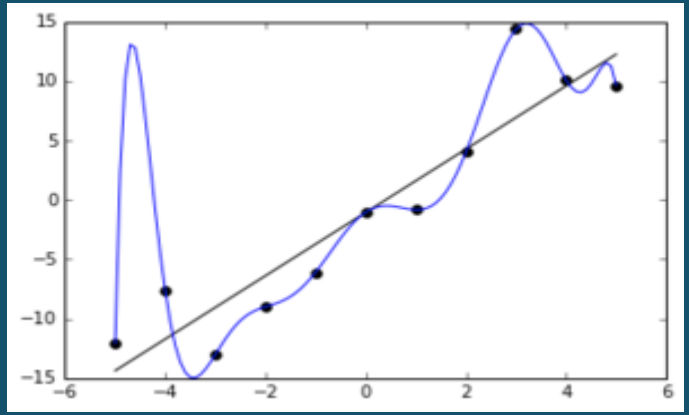
Bias & Variance

https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff

- **Error due to bias:** Difference between the expected prediction value of the model and the actual value
- **Error due to variance:** Variability of a model prediction from a given point
- **Bias** is how displaced a model's predictions are from correct values, while **Variance** is the degree to which these predictions vary between model iterations
- **$Err(x)=Bias^2+Variance+Irreducible\ Error$**



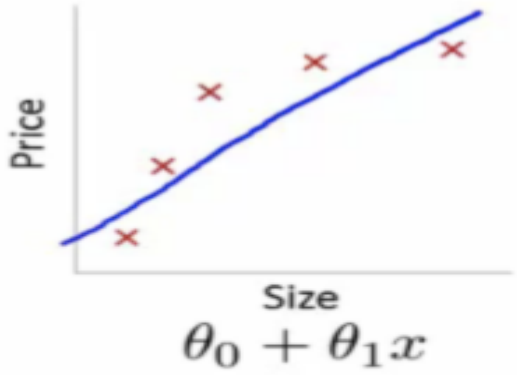
https://www.kdnuggets.com/2016/08/bias-variance-tradeoff-overview.html



Model Fit

https://en.wikipedia.org/wiki/Overfitting

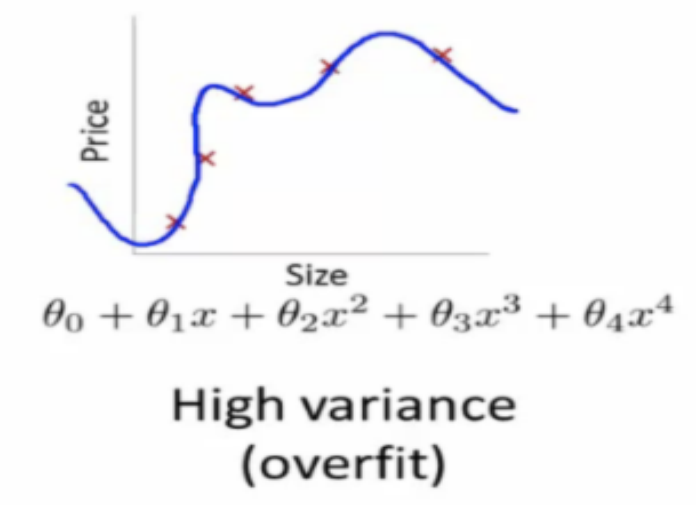
Underfitting



- A statistical model or a machine learning algorithm is said to be **underfitting** when it cannot capture the underlying trend of the data.
- It usually happens when we have less data to build an accurate model and also when we try to build a linear model with a non-linear data.
- Underfitting can be avoided by using more data and also reducing the features by feature selection

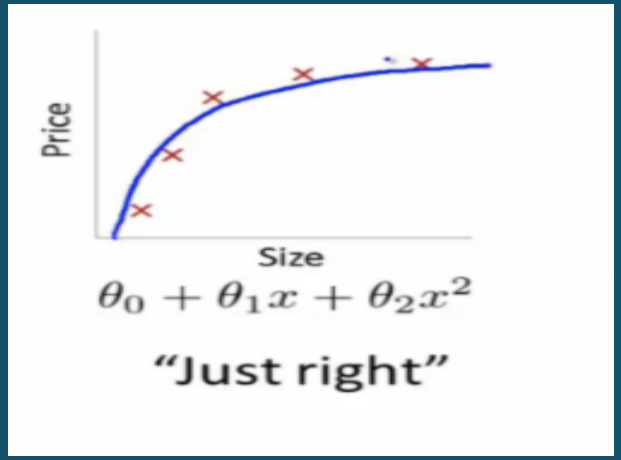
https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/

Overfitting



- **Overfitting** occurs when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data
- Model is subject to low bias, but high variance
- Too much data and complex models result in overfitting

https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/



Methods to prevent Overfitting

https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/

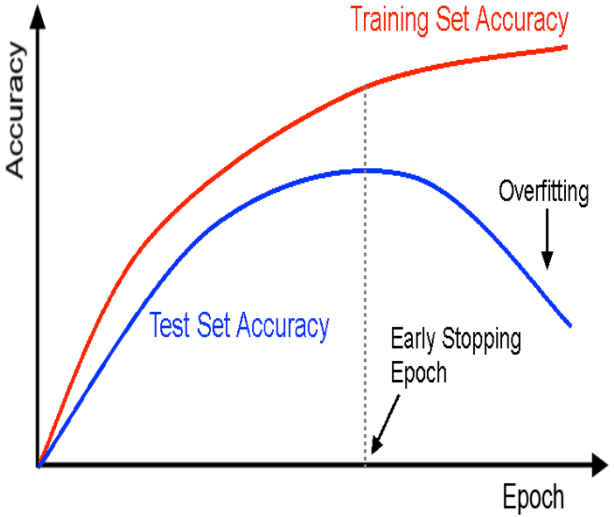
Cross Validation

- This is a cross-validation used to prevent the overlap of the test sets
- **First step:** split data into k disjoint subsets : D1, … Dk, of equal size, called folds
- **Second step:** use each subset in turn for testing, the remainder for training
- **Training and testing is performed k times**

https://imada.sdu.dk/~marco/Teaching/AY2010-2011/DM825/animation.gif

Early Stopping

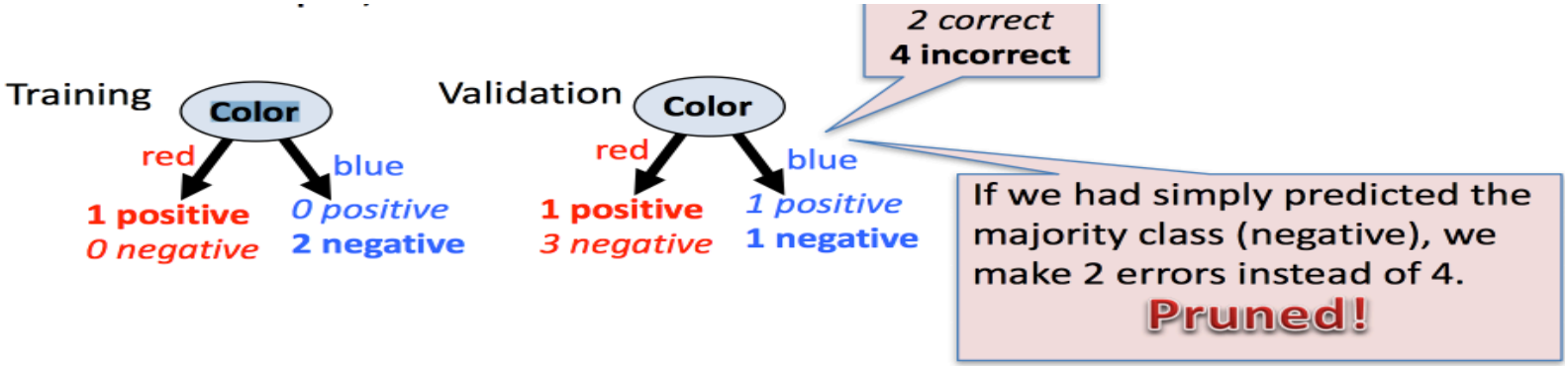
- Split the training data into a training set and a validation set, e.g. in a 2-to-1 proportion.
- Train only on the training set and evaluate the per-example error on the validation set once in a while, e.g. after every fifth epoch.
- Stop training as soon as the error on the validation set is higher than it was the last time it was checked.
- Use the weights the network had in that previous step as the result of the training run.



https://deeplearning4j.org/earlystopping

Pruning

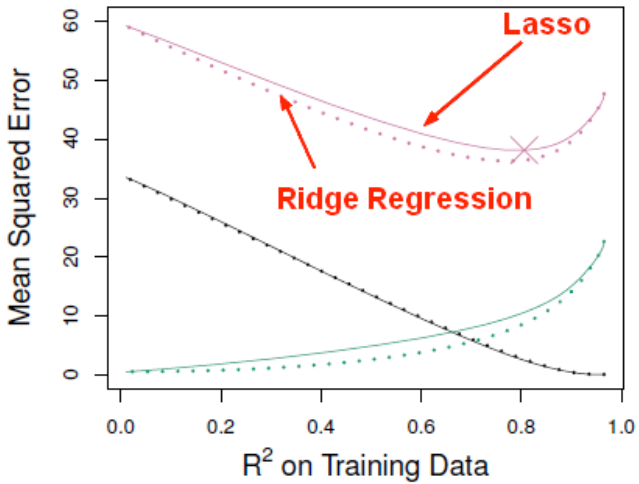
- The process of adjusting decision trees to minimize **Misclassification Error.**
- The replacement takes place if a decision rule establishes that the expected error rate in the subtree is greater than in the single leaf.



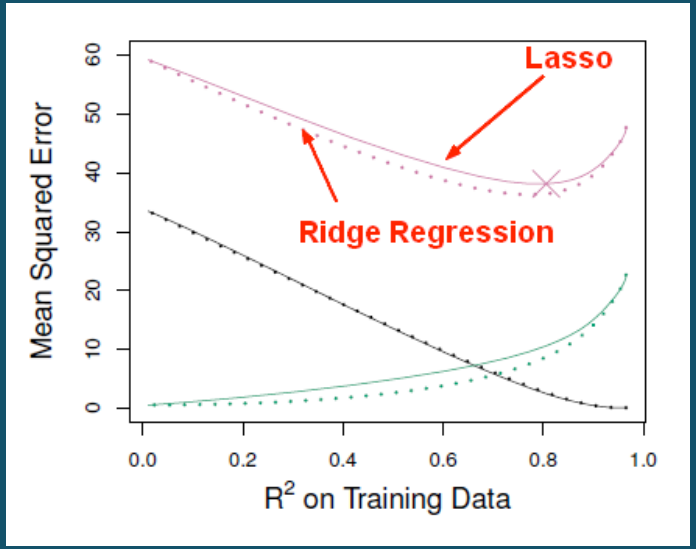
https://www.cc.gatech.edu/~bboots3/CS4641-Fall2016/Lectures/Lecture3_1.pdf

Shrinkage/Regularization

- These methods **constrain** or **shrink** the coefficient estimates towards zero
- If there is noise in the training data, then the estimated coefficients won't generalize well to the future data
- Help to avoid overfitting and will perform at the same time feature selection for certain regularization norms



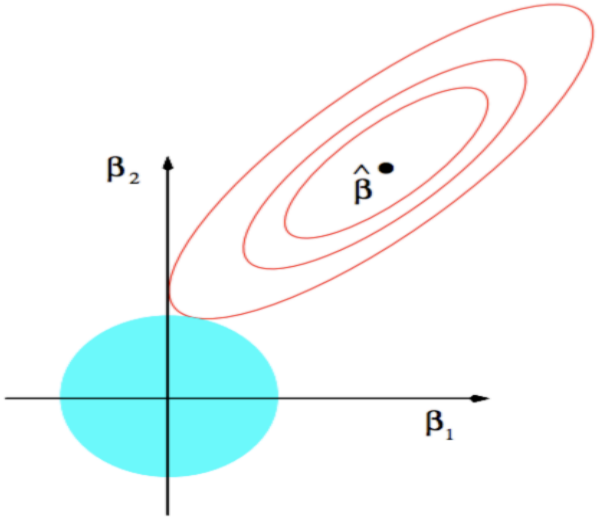
https://gerardnico.com/data_mining/shrinkage



Ridge & LASSO Regression

https://gerardnico.com/data_mining/shrinkage

Ridge Regression



- If all the weights are unconstrained, they are susceptible to high variance
- The **Ridge Penalization** will force the parameters to be relatively small
- The bigger the penalization, the smaller the coefficients are

https://stats.stackexchange.com/questions/123205/relation-between-the-tuning-parameter-lambda-parameter-estimates-beta-i-a

- **Ordinary Least Squares objective**:

$$\hat{\beta} = argmin_{\beta}\{\sum_{i=1}^{N}(y_i - \sum_{j=1}^{m} x_{ij}\beta_j)^2\}$$

- **Ridge Regression**:

$$\hat{\beta}^{ridge} = argmin_{\beta}\{\sum_{i=1}^{N}(y_i - \sum_{j=1}^{m} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{m} \beta_j^2\}$$

where **λ** is the tuning parameter which controls the strength of the **penalty term**

https://www3.cs.stonybrook.edu/~has/CSE545/Slides/6.11-11.pdf

Pros & Cons

PROS

- Ridge regression can reduce the variance
- Can improve predictive performance
- Mathematically simple computations

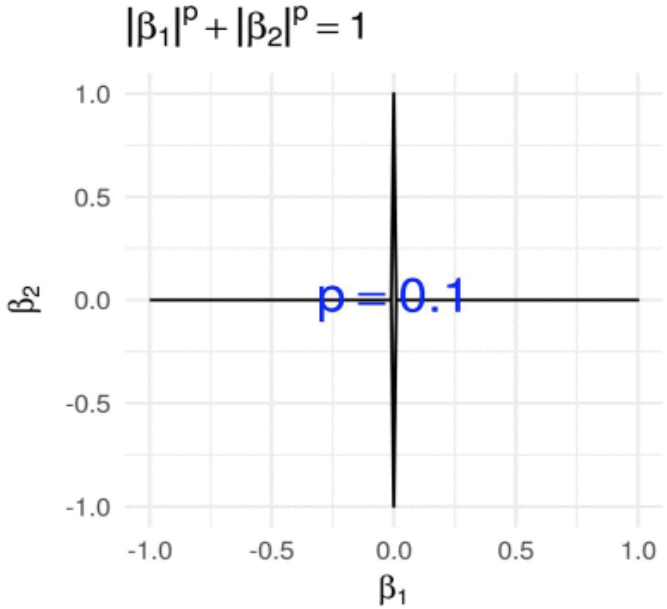
CONS

- Ridge regression is not able to shrink coefficients to exactly zero
- As a result, it cannot perform attribute selection

⇒ **Alternative**: LASSO Regression

https://www.is.uni-freiburg.de/ressourcen/business-analytics/xx_regularization.pdf

Lasso Regression

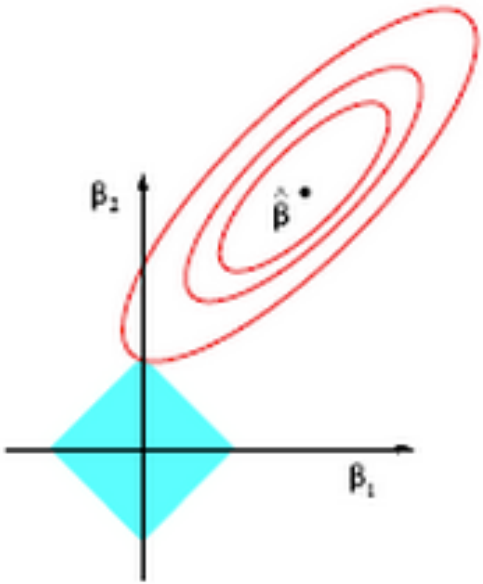


L **Least**
A **Attribute**
S **Selection and**
S **Shrinkage**
O **Operator**

- **Supervised Technique** useful for **attribute selection** and to **prevent overfitting** training data
- Penalized regression method

http://www.alexejgossmann.com/contours_of_statistical_penalty_functions_as_GIFs/

How it works



Penalizes the sum of absolute value of weights found by the regression.

There are two methods to add constraints to add penalty:

$$Min \underbrace{\sum_{i=1}^{N}(y_i - \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i})^2}_{\text{sum of square error term}} + \underbrace{\lambda(\sum_{j=1}^{2}|\beta_j|)}_{\text{Penalty term}}$$

$$\underbrace{\lambda(\sum_{j=1}^{2}|\beta_j|)}_{\text{Penalty term}} \leq \eta$$

- shrink some weights to zero
- λ is tuning parameter chosen by **cross validation**

https://www.quora.com/How-would-you-describe-LASSO-regularization-in-laymens-terms

Properties & Advantages of LASSO

Properties

- Useful when number of classes is very less than number of attributes
- Makes the model sparser and reduces the possibility of overfitting

Advantages

- Improves the **prediction accuracy** and interpretability of regression models
- For feature selection, **LASSO** uses **convex optimization** to find best features. So, it converges faster.
- Convenient when we want some automatic feature/variable selection

2

Application of Logistic Regression in the Study of Students' Performance Level

Miftar Ramosacaj, Dr. Vjollca Hasani, Dr. Alba Dumi

Journal of Educational and Social Research MCSER Publishing, Rome-Italy, Vol. 5 No. 3
September 2015

Paper Link

Introduction

- **Aim**- Predict students' performance level in first semester of studies
- **Classification Method**- "**Logistic Regression Analysis**"
- **Assumption Made** -
  1. The number of credits gained after first semester signify the performance of the student
  2. Class Attribute is based on the number of credits gained after first semester:
     a. Y=1 that is > 30 credits **Good Performer**
     b. Y=0 that is < 30 credits **Low Performer**

Identified Attributes

Dependent/Class Attribute

Independent/Classifier Attribute

Attributes with insignificant impact (ignored in regression)

Credits Gained (Binary)

1. > 30 Credits
0. < 30 Credits

Social Environment (Non-Binary)

1. Stressful
2. Not Stressful
3. Very Stressful

High School Points (Non-Binary)

1. 4000-5000
2. 3000-4000
3. 2000-3000

Gender (Binary)

1. Female
0. Male

Student Location (Binary)

1. Urban Area
0. Village

Type of School (Binary)

1. Private
0. Public

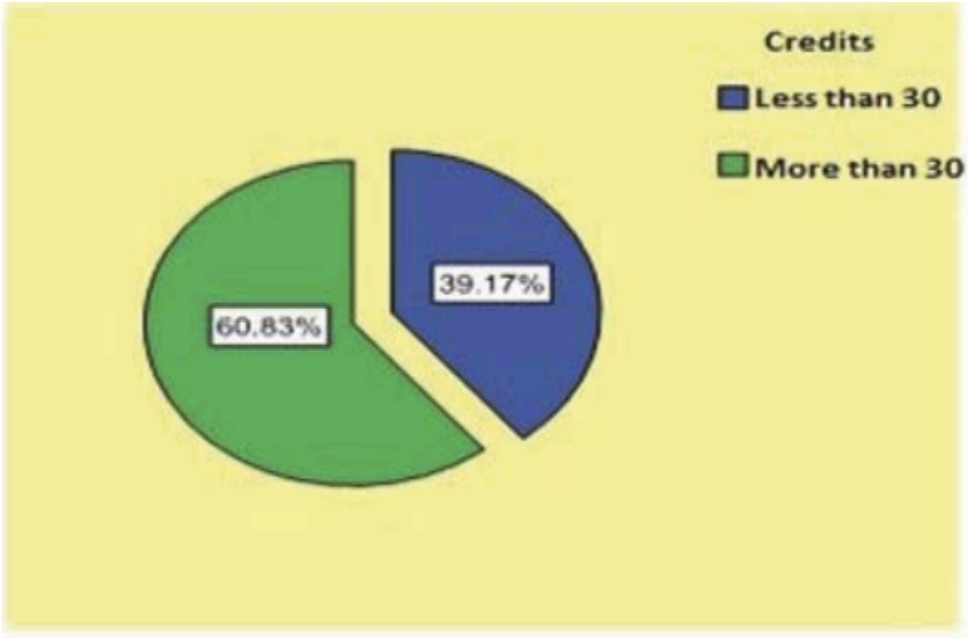
School Location (Binary)

1. Urban Area
0. Village

Class/Dependent Attribute Distribution

Data collected via Questionnaires filled by 240 Freshmen

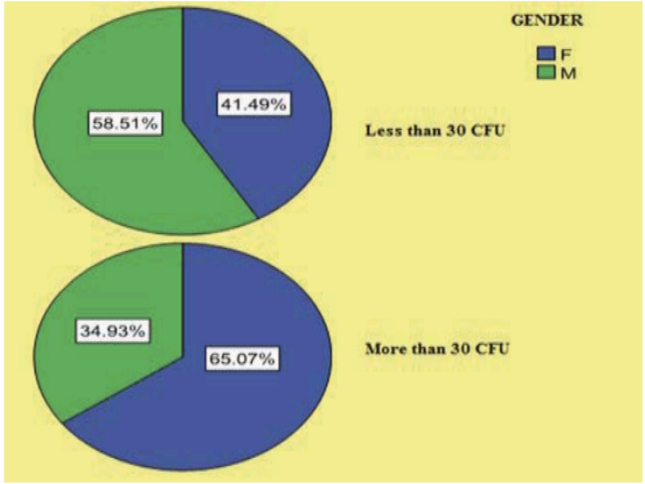
Two Categories of Students Based on First Semester Results (Class/Dependent Attribute)

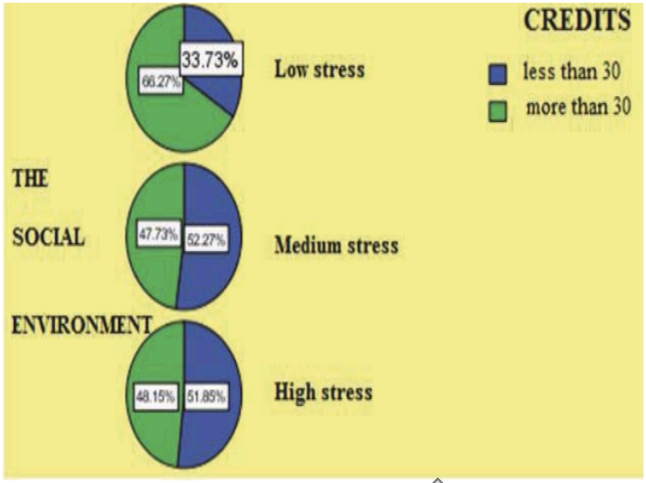


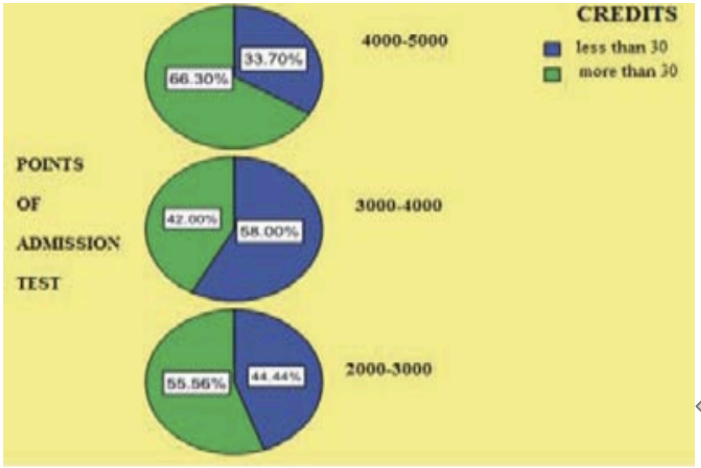
http://www.mcser.org/journal/index.php/jesr/article/download/7722/7403

Few observations made on the Classifier Attributes

Data collected via Questionnaires filled by 240 Freshmen







Gender

Stress Level

High School Points

http://www.mcser.org/journal/index.php/jesr/article/download/7722/7403

Logistic Regression Process

- **General form of Logistic Regression Equation:**

$$\theta = x = \frac{e^{(\alpha+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_n x_n)}}{1+e^{(\alpha+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_n x_n)}}$$

- **Logistic Regression equation for our problem after removal of insignificant attributes (using significant score):**

$$P(Y=1) = \theta = 1-P(Y=0) \quad \& \quad \theta = \frac{e^{(0.54-0.91 x_1-0.47 x_5+0.62 x_6-0.45 x_7)}}{1+e^{(0.54-0.91 x_1-0.47 x_5+0.62 x_6-0.45 x_7)}} \quad \alpha = 0.54$$

http://www.mcser.org/journal/index.php/jesr/article/download/7722/7403

Logistic Regression Process

$$P(Y=1) = \theta = 1 - P(Y=0) \quad \& \quad \theta = \frac{e^{(0.54 - 0.91x_1 - 0.47x_5 + 0.62x_6 - 0.45x_7)}}{1 + e^{(0.54 - 0.91x_1 - 0.47x_5 + 0.62x_6 - 0.45x_7)}} \qquad \alpha = 0.54$$

- Y=1 when θ > 0.5 => that is Passed credits > 30 (Student is a good performer)
- Y=0 when θ < 0.5 => that is Passed credits < 30 (Student is a bad performer)
- x_1= Gender (β_i= -0.91)
- x_5= High School Points (β_5= -0.47)
- x_6= Student Location (β_6= 0.62)
- x_7= Stress Level (Social Environment) (β_7= 0.45)

http://www.mcser.org/journal/index.php/jesr/article/download/7722/7403

...What we want is a **Machine** that can **Learn** from experience…

-Alan Turing

(1947)

What was their conclusion?

Was it good enough?

Let's look at some of their findings!

https://giphy.com/gifs/reaction-a5vil92PAF89q

LOGISTIC REGRESSION ANALYSIS

01 Female Students perform better than male counterparts

02 Environment where they live in is far from being appropriate for the research

03 High school results positively affect the student performance in first semester

04 Creating non stressful conditions positively contributes to increasing student performance