COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI ICADE CIHS

Machine Learning I

- Logistic Regression

comillas.edu

Multiple Linear Regression Statement

- The Multiple Linear Regression model allows estimate the continuous random variable Y from a set of $\boldsymbol{p}$ input variables

Deterministic component ($\boldsymbol{p}+\mathbf{1}$ coefficients)

Random component

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \epsilon$$

HYPERPLANE OF POPULATION REGRESSION
(unknown theoretical relation)

Sample

$(x_1, y_1) \ldots (x_N, y_N)$

$x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$

$$\epsilon \approx N(0, \sigma^2)$$

Random variable that considers the output variations with respect to the expected value of the deterministic component (error, noise)

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_p x_p$$

ESTIMATED REGRESSION LINE

Definition

- Logistic regression is a form of regression analysis in which the outcome variable is binary or dichotomous

- General theory: analysis of variance (ANOVA) and logistic regression all are special cases of *General Linear Model (GLM)*

- Two cases: *simple and multiple logistic regression*

What is Logistic Regression?

In a nutshell:

- A statistical method used typically to ***model dichotomous or binary outcomes*** using predictor variables.

- Used when the research method is focused ***on whether or not an event occurred***, rather than when it occurred (time course information is not used).

- *What is the "Logistic" component*?

  Instead of modeling the outcome Y directly, the method models the ***log odds(Y)*** using the logistic function.

Odd ratio is ratio de riesgo o razón de probabilidad in Spanish

What is Logistic Regression?

- *What is the "Regression" component?*

  Methods used to quantify association between an outcome and predictor variables. They could be used to build predictive models as a function of predictors.

Logistic Regression is based on Odds Ratios

- *Does not model the outcome directly*, which leads to effect estimates quantified by means (i.e., differences in means)

- Estimates of effect are instead quantified by "*Odds Ratios*"

- Relationship between *Odds & Probability*

$$\text{Odds}(\text{event}) = \frac{\text{Probability}(\text{event})}{1\text{-Probability}(\text{event})}$$

$$\text{Probability}(\text{event}) = \frac{\text{Odds}(\text{event})}{1+\text{Odds}(\text{event})}$$

The Odds Ratio

- Definition of Odds Ratio: *Ratio of two odds estimates*

- If Pr(response | A) = 0.40 and Pr(response | B) = 0.20

$$Odd(response|A\ \ group) = \frac{0.40}{1-0.40} = 0.667$$

$$Odd(response|\ B\ group) = \frac{0.20}{1-0.20} = 0.25$$

Then the *odd ratio (OR)* is:

$$OR(A\ vs.B) = \frac{0.667}{0.25} = 2.67$$

The Odds Ratio

- Outcome = response, odd ratio=2.67

  Then, the odds of a response respect A group was estimated *to be 2.67 times* the odds of having a response from the B group.

  Alternatively, the odds of having a response *were 167% higher* in the A group than in the B group.

- An Odds Ratio of 2.67 for A vs. B ***does NOT mean*** that the outcome is 2.67 times as ***LIKELY*** to occur.
- It ***DOES mean*** that the ***ODDS*** of the outcome occurring are 2.67 times as high for A. vs. B.

Logistic Regression

- Simple logistic regression = logistic regression with 1 predictor variable

- Multiple logistic regression = logistic regression with multiple predictor variables

- Multiple logistic regression =
  Multivariable logistic regression =
  Multivariate logistic regression

The Logistic Regression Model

Logistic Regression:

$$\ln\left(\frac{\mathrm{P(Y)}}{\mathrm{1\text{-}P(Y)}}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_K X_K$$

Linear Regression:

$$\mathrm{Y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_K X_K + \varepsilon$$

The Logistic Regression Model

predictor variables

$$\ln\left(\frac{P(Y)}{1-P(Y)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_K X_K$$

dichotomous outcome

$\ln\left(\frac{P(Y)}{1-P(Y)}\right)$ is the log(odds) of the outcome.

The Logistic Regression Model

$$\ln\left(\frac{P(Y)}{1-P(Y)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_K X_K$$

intercept

model coefficients

$$\ln\left(\frac{P(Y)}{1-P(Y)}\right)$$ is the log(odds) of the outcome.

The Logistic Regression Model

$$\ln\left(\frac{P(Y)}{1\text{-}P(Y)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_K X_K$$

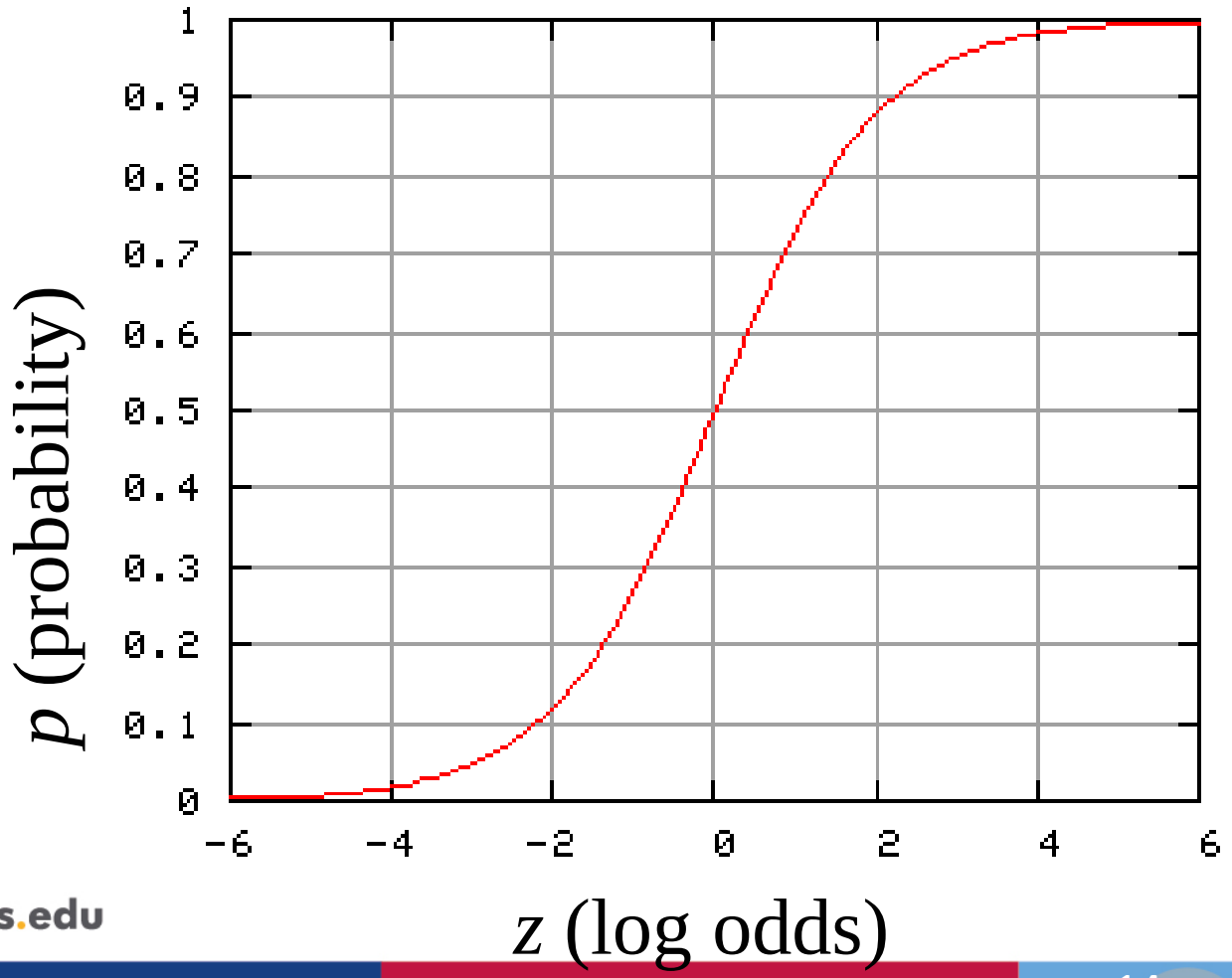
$$\Updownarrow$$

$$P(Y) = \frac{\exp\left(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_K X_K\right)}{1 + \exp\left(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_K X_K\right)}$$

In this latter form, the logistic regression model directly relates the probability of Y to the predictor variables.

The logistic curve

$$LOGIT(p) = \ln\left(\frac{p}{(1-p)}\right) = z \quad \Leftrightarrow \quad p = \frac{\exp(z)}{1+\exp(z)}$$



Logistic Regression

- Relationships among probability, odds and log

Measure	Min	Max	Name
Pr(Y=1)	0	1	prob
$\frac{\Pr(Y=1)}{1-\Pr(Y=1)}$	0	∞	odds
$\log\left(\frac{\Pr(Y=1)}{1-\Pr(Y=1)}\right)$	-∞	∞	log odds

Logistic Regression

- Hypothesis testing

Usually there are two types of tests:

- Likelihood Ratio test
- Wald test

Logistic Regression

Likelihood Ratio test

- Idea is to compare the (log) Likelihood of two models to test

$$H_0 : \beta_K = 0$$

- Two models:
  1. Full model = with predictor included
  2. Reduced model = without predictor

- Then

$$-2\ln\left(\frac{\hat{L}_{\text{Reduced}}}{\hat{L}_{\text{Full}}}\right) = -2\ln \hat{L}_{\text{Reduced}} - \left(-2\ln \hat{L}_{\text{Full}}\right)$$

$$\sim \chi^2 \text{ with df} = \#\text{ of extra parameters in full model}$$

Logistic Regression

Wald test

- Idea is to use large sample Z statistic from a single model to test:

$$H_0 : \beta_K = 0$$

$$\text{Here, } Z = \frac{\hat{\beta}_K}{SE_{\hat{\beta}_K}} \text{ where} Z \sim N(0,1)$$

- As the sample size gets larger and larger, the Wald test will approximate the Likelihood ratio test.

- The LR test is preferred but Wald test is common

The Logistic Regression Model. Cases

- LRM-1: Admission dataset
  - Problem description:

A researcher is interested in how variables, such as ***gre*** (Graduate Record Exam scores), ***gpa*** (grade point average) and ***prestige*** of the undergraduate institution, effect ***admission*** into graduate school. The response variable, admit/don't admit, is a binary variable

The variables ***gre*** and ***gpa*** are continuous. The variable ***rank (prestige)*** takes on the values 1 through 4. Institutions with a rank of 1 have the highest prestige, while those with a rank of 4 have the lowest.

The Logistic Regression Model. Cases

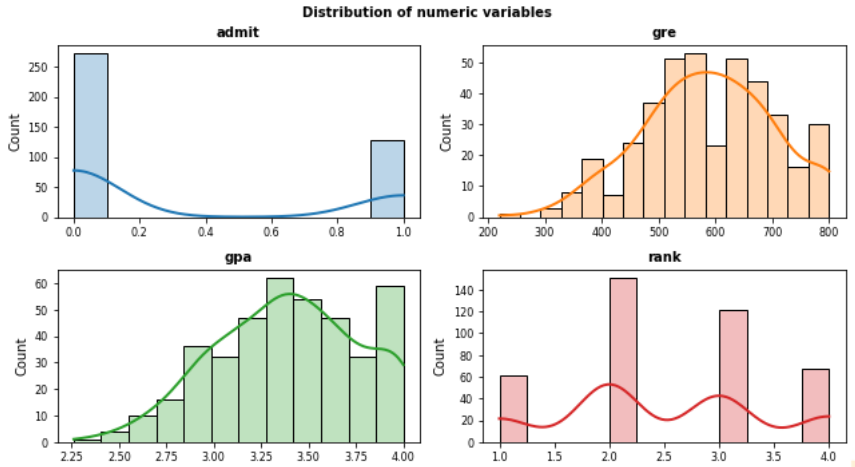
- LRM-1: Admission dataset

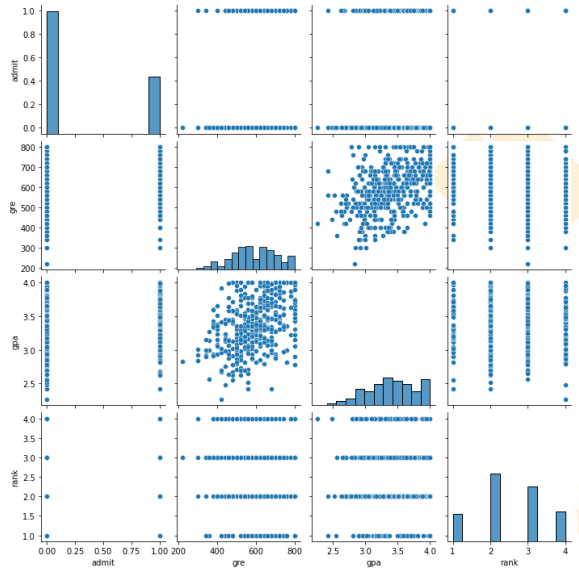
```
# Data Loading

admissions = pd.read_csv('Admissions.csv')
```

	admit	gre	gpa	rank
count	400.000000	400.000000	400.000000	400.00000
mean	0.317500	587.700000	3.389900	2.48500
std	0.466087	115.516536	0.380567	0.94446
min	0.000000	220.000000	2.260000	1.00000
25%	0.000000	520.000000	3.130000	2.00000
50%	0.000000	580.000000	3.395000	2.00000
75%	1.000000	660.000000	3.670000	3.00000
max	1.000000	800.000000	4.000000	4.00000

```
Rank categories
2    151
3    121
4     67
1     61
Name: rank, dtype: int64
```





comillas.edu

The Logistic Regression Model. Cases

- LRM-1: Admission dataset. MODEL

```
import statsmodels.formula.api as sm
# Fit the logistic resgression line using 'logit'
model = sm.logit("admit ~  gre + gpa + dmy_2 + dmy_3+ dmy_4", data=data).fit()
```

```
Optimization terminated successfully.
         Current function value: 0.573147
         Iterations 6
                           Logit Regression Results
==============================================================================
Dep. Variable:                  admit   No. Observations:                  400
Model:                          Logit   Df Residuals:                      394
Method:                           MLE   Df Model:                            5
Date:                Fri, 16 Jun 2023   Pseudo R-squ.:                 0.08292
Time:                        15:24:38   Log-Likelihood:                -229.26
converged:                       True   LL-Null:                       -249.99
Covariance Type:            nonrobust   LLR p-value:                 7.578e-08
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -3.9900      1.140     -3.500      0.000      -6.224      -1.756
gre            0.0023      0.001      2.070      0.038       0.000       0.004
gpa            0.8040      0.332      2.423      0.015       0.154       1.454
dmy_2         -0.6754      0.316     -2.134      0.033      -1.296      -0.055
dmy_3         -1.3402      0.345     -3.881      0.000      -2.017      -0.663
dmy_4         -1.5515      0.418     -3.713      0.000      -2.370      -0.733
==============================================================================
```

The Logistic Regression Model. Cases

- LRM-1: Admission dataset. MODEL Analysis

  - The previous output shows the coefficients of the model, their standard errors, the z-statistic (sometimes called a *Wald z-statistic*), and the associated p-values.
  - Both ***gre*** and ***gpa*** are ***statistically significant***, as are the three terms for ***rank***.
  - The *logistic regression coefficients give the change in the* ***log odds*** of the outcome for a one unit increase in the predictor variable.

    For every one unit change in gre, the ***log odds*** of admission (versus non-admission) increases by 0.0023.

    For a one unit increase in gpa, the ***log odds*** of being admitted to graduate school increases by 0.804.

    The indicator variables for rank have a slightly different interpretation. For example, having attended an undergraduate institution with rank of 2, versus an institution with a rank of 1, changes the log odds of admission by -0.675.

The Logistic Regression Model. Cases

- LRM-1: Admission dataset. MODEL Analysis
  - Residuals *do not have a direct interpretation* such in linear regression
  - *LL-Null (Null deviance)* considers the model without predictors. It is the value of log-likelihood of the model when no independent variable is included(only an intercept is included) LL-null = 249.99
  - *Log-Likelihood (Residual deviance)* considers the model with all the predictors. The natural logarithm of the Maximum Likelihood Estimation(MLE) function. MLE is the optimization process of finding the set of parameters that result in the best fit

    Log-Likelihood = 229.26

The Logistic Regression Model. Cases

- LRM-1: Admission dataset. MODEL Analysis
  - One measure of model fit is the significance of the overall model. This test asks whether the model with predictors fits significantly better than a model with just an intercept (i.e., a null model).
  - The test statistic is the difference between the residual deviance for the model with predictors and the null model. The test statistic is distributed chi-squared with degrees of freedom equal to the differences in degrees of freedom between the current and the null model (i.e., the number of predictor variables in the model).
  - The difference in deviance of the models without and with predictors:

    |LL-Null - Log-Likehood |= 20.73

The Logistic Regression Model. Cases

- LRM-1: Admission dataset. MODEL Analysis

  - The degrees of freedom for the difference between the two models is equal to the number of predictor variables in the mode, in our case 5 predictors.

  - Finally, the p-value of the difference with 5 degrees of freedom is obtained by a **chi-square** under the name LLR p-value . This value can be thought of as the substitute to the p-value for the overall F-value of a linear regression mode. Here it is very low: 7.578194e-08

  - This confirms that the model as a whole fits significantly better than an empty model

  - Finally ***Pseudo R-squ***. Is a substitute for the R-squared value in Least Squares linear regression. It is the ratio of the log-likelihood of the null model to that of the full model. *(It has to be multiplied by 1000)*

The Logistic Regression Model. Cases

- LRM-1: Admission dataset. MODEL Analysis
  - *AIC criteria*: convenient value should be in line with residual values. It is useful for comparing different models using the same output. Lower AIC value, better model

```
# Akaike criteria
model.aic
```

```
470.51749247589896
```

Logistic Regression Model. Cases

- LRM-1: Admission dataset. MODEL

It is possible to obtain the odds ratios and their confidence intervals by exponentiating the coefficients and confidence intervals obtained.

```
# Calculating Odd-ratios

# ... Define and fit model
odds_ratios = pd.DataFrame(
    {
        "Odd Ratio": model.params,
        "Lower CI": model.conf_int()[0],
        "Upper CI": model.conf_int()[1],
    }
)
odds_ratios = np.exp(odds_ratios)
print(odds_ratios)
```

```
           Odd Ratio  Lower CI  Upper CI
Intercept   0.018500  0.001981  0.172783
gre         1.002267  1.000120  1.004418
gpa         2.234545  1.166122  4.281877
dmy_2       0.508931  0.273692  0.946358
dmy_3       0.261792  0.133055  0.515089
dmy_4       0.211938  0.093443  0.480692
```

- As example of interpretation, for *a one unit increase in gpa, the odds* of being admitted to graduate school (versus not being admitted) increase by a factor of 2.23.

- Also, note that even it is produced, the odds ratio for the intercept is not generally interpreted.

Logistic Regression Model. Cases

- LRM-1: Admission dataset. MODEL - Prediction

Prediction of probabilities can be computed for both categorical and continuous predictor variables.

In order to create predicted probabilities, it is required to create a new data frame with the values of the independent variables to take on to create our predictions for the different rank values.

As an example, the probability of admission will be predicted at each value of rank, **holding** *gre* ***and*** *gpa* ***at their means***.

First the new data frame will be created and presented.

Logistic Regression Model. Cases

- LRM-1: Admission dataset. MODEL - Prediction

Creating the new data frame

```
# Creation of a new data frame with mean values for gre and gpa

gre_m=admissions['gre'].mean()  # mean value
gpa_m=admissions['gpa'].mean()  #mean value

print(" mean values gre and gpa", gre_m, gpa_m)

# Configuration of a new datasey
new_data = {
            'gre' :  [gre_m, gre_m, gre_m, gre_m],
            'gpa' : [gpa_m, gpa_m, gpa_m, gpa_m],
            'rank': [1, 2, 3, 4]
}

print(new_data)

#Conversion to dataframe
new_df = pd.DataFrame(new_data)
new_df

#Creation of dummies
test=pd.get_dummies(new_df, columns=['rank'], prefix='dmy',drop_first=True)
```

	gre	gpa	dmy_2	dmy_3	dmy_4
0	587.7	3.3899	0	0	0
1	587.7	3.3899	1	0	0
2	587.7	3.3899	0	1	0
3	587.7	3.3899	0	0	1

Logistic Regression Model. Cases

- LRM-1: Admission dataset. MODEL - Prediction

The new data frame will be used to calculate the predicted probabilities. First, two new variables are created: probability and Classification. *A threshold of 0.5* is used for classification.

```
# performing predictions on the test datdaset
yhat = model.predict(exog=test)

# Predicted classification
# ============================================================================
classification = np.where(yhat <0.5, 0, 1)

test['Probability']=yhat
test['Classification']=classification
test
```

	gre	gpa	dmy_2	dmy_3	dmy_4	Probability	Classification
0	587.7	3.3899	0	0	0	0.516602	1
1	587.7	3.3899	1	0	0	0.352285	0
2	587.7	3.3899	0	1	0	0.218612	0
3	587.7	3.3899	0	0	1	0.184668	0

- The predicted probability of being accepted into a graduate program is *0.52* for students from the highest prestige undergraduate institutions (rank=1), and *0.18* for students from the lowest ranked institutions (rank=4), *holding gre and gpa at their means*.

Logistic Regression Model. Cases

- LRM-1: Admission dataset. MODEL – Whole case

  We can split the dataset in training and test

```
# Complete Analysis using training and test datasets
# División de los datos en train y test
#
================================================================================
====
X = data.drop(columns = 'admit')
y = data['admit']


from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
                                        X,
                                        y.values.reshape(-1,1),
                                        train_size   = 0.8,
                                        random_state = 1234,
                                        shuffle      = True
                                    )
```

Logistic Regression Model. Cases

- LRM-1: Admission dataset. MODEL - Prediction

Resulting model after training

```
Optimization terminated successfully.
         Current function value: 0.601202
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:                      y   No. Observations:                  320
Model:                          Logit   Df Residuals:                      314
Method:                           MLE   Df Model:                            5
Date:                Mon, 19 Jun 2023   Pseudo R-squ.:                 0.07416
Time:                        11:49:58   Log-Likelihood:                -192.38
converged:                       True   LL-Null:                       -207.80
Covariance Type:            nonrobust   LLR p-value:                 1.016e-05
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -2.9074      1.213     -2.396      0.017      -5.285      -0.530
gre            0.0023      0.001      1.958      0.050   -2.47e-06       0.005
gpa            0.5231      0.356      1.469      0.142      -0.175       1.221
dmy_2         -0.7251      0.334     -2.170      0.030      -1.380      -0.070
dmy_3         -1.3675      0.364     -3.754      0.000      -2.081      -0.654
dmy_4         -1.4922      0.454     -3.287      0.001      -2.382      -0.602
==============================================================================
```

We remove gpa, gre is in the border to be eliminated

Logistic Regression Model. Cases

- LRM-1: Admission dataset. MODEL - Prediction

Resulting model after training

```
Optimization terminated successfully.
         Current function value: 0.604604
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:                      y   No. Observations:                  320
Model:                          Logit   Df Residuals:                      315
Method:                           MLE   Df Model:                            4
Date:                Mon, 19 Jun 2023   Pseudo R-squ.:                 0.06892
Time:                        11:59:29   Log-Likelihood:                -193.47
converged:                       True   LL-Null:                       -207.80
Covariance Type:            nonrobust   LLR p-value:                 9.235e-06
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -1.4921      0.723     -2.063      0.039      -2.909      -0.075
gre            0.0030      0.001      2.689      0.007       0.001       0.005
dmy_2         -0.7666      0.332     -2.311      0.021      -1.417      -0.116
dmy_3         -1.3481      0.362     -3.725      0.000      -2.057      -0.639
dmy_4         -1.5175      0.453     -3.352      0.001      -2.405      -0.630
==============================================================================
```

It seems that this model is better at keeping properties

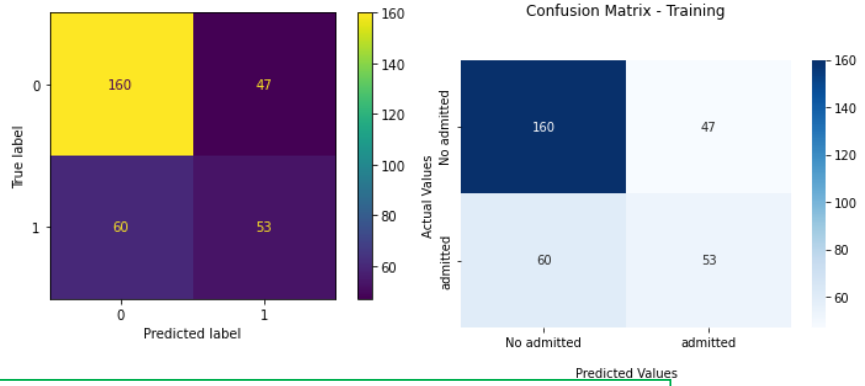
Logistic Regression Model. Cases

- LRM-1: Admission dataset. MODEL - Prediction

Resulting model after training. *Threshold =0.4*

```
Accuracy in training is: 66.5625%
```

Predicción	0	1
Real		
0	160	47
1	60	53

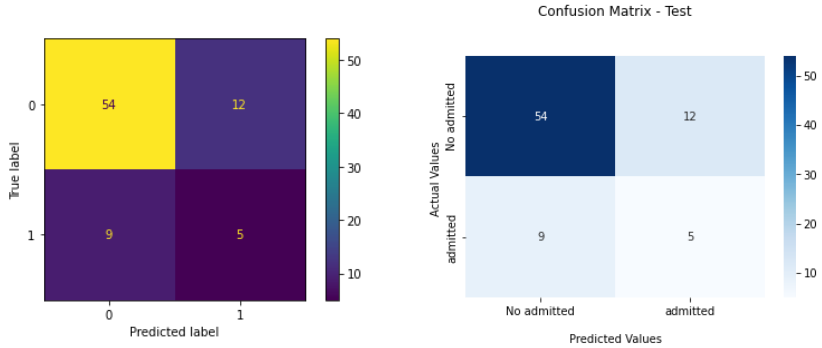


- Correctly classified = 160+53 = 212
- Incorrectly classified= 60+47 = 107
- Ratio of correct classification:

212/320= 0.7 ➔ *66,56%*

```
Accuracy in test is: 73.75%
```

Predicción	0	1
Real		
0	54	12
1	9	5



- Correctly classified = 54+5 = 59
- Incorrectly classified= 9+12 = 21
- Ratio of correct classification:

59/80= 0.7 ➔ 73.75*66,56%*

Logistic Regression Model. Cases

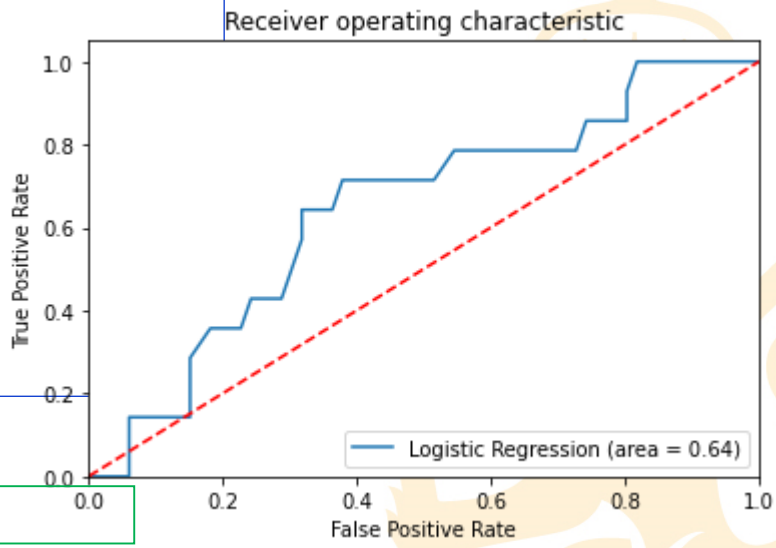
ROC Curve for the test dataset

- LRM-1: Admission dataset. MODEL - Prediction

```
#Libraries required for ROC curve
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
# ROC curve for the test dataset

logit_roc_auc = roc_auc_score(y_test, model.predict(exog = X_test)) # AUC estimatiion
fpr, tpr, thresholds = roc_curve(y_test, model.predict(exog = X_test))  #ROC curve
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.savefig('Log_ROC')
plt.show()

...
# calculate AUC
auc = roc_auc_score(y_test, model.predict(exog = X_test))
print('AUC: %.3f' % auc)
```




```
AUC: 0.643
```

Logistic Regression Model. Cases

- LM-1: Admission dataset. Construction of deciles

Suppose a Logistic Regression model based on the whole dataset, including all inputs. A probability is estimated per case,also, it is known if the student was admitted or not.

	actual	predicted_prob
0	0	0.172627
1	1	0.292175
2	1	0.738408
3	1	0.178385
4	0	0.118354
...	...	...
395	0	0.488670
396	0	0.165504
397	0	0.181062
398	0	0.463667
399	0	0.300731

400 rows × 2 columns

This can be classified in descent order of probability

	predicted_prob	actual
2	0.738408	1
293	0.733722	0
12	0.720539	1
150	0.696072	1
69	0.694368	0
...	...	...
17	0.078953	0
71	0.074860	0
48	0.072354	0
304	0.071985	0
289	0.058786	0

The number of cases per decile will be:
400/10 = 40 examples per decile

	predicted_prob	actual	decile
2	0.738408	1	1
293	0.733722	0	1
12	0.720539	1	1
150	0.696072	1	1
69	0.694368	0	1
...	...	...	...
17	0.078953	0	10
71	0.074860	0	10
48	0.072354	0	10
304	0.071985	0	10
289	0.058786	0	10

400 rows × 3 columns

First decile, first 40 cases with higher probability and the same for the other deciles

Logistic Regression Model. Cases

- LM-1: Admission dataset. Construction of deciles

  Estimation of gain per decile

	decile	gain
0	1	24
1	2	19
2	3	21
3	4	11
4	5	12
5	6	9
6	7	8
7	8	14
8	9	5
9	10	4

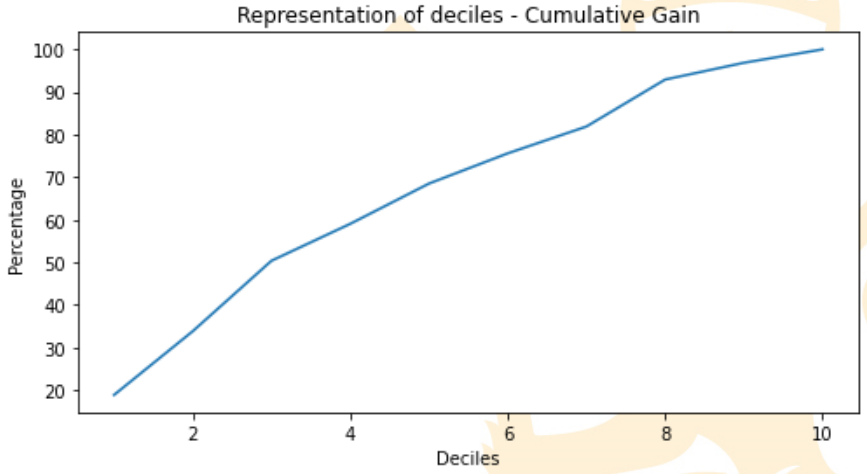
24 cases in this decile that were admitted

Total admitted 127

The right column is the cumulative gain

	decile	gain	gain_percentage
0	1	24	18.897638
1	2	19	33.858268
2	3	21	50.393701
3	4	11	59.055118
4	5	12	68.503937
5	6	9	75.590551
6	7	8	81.889764
7	8	14	92.913386
8	9	5	96.850394
9	10	4	100.000000

Graphic representation of cumulative gain per decile



Logistic Regression Model. Cases

- LM-1: Admission dataset. Construction of deciles

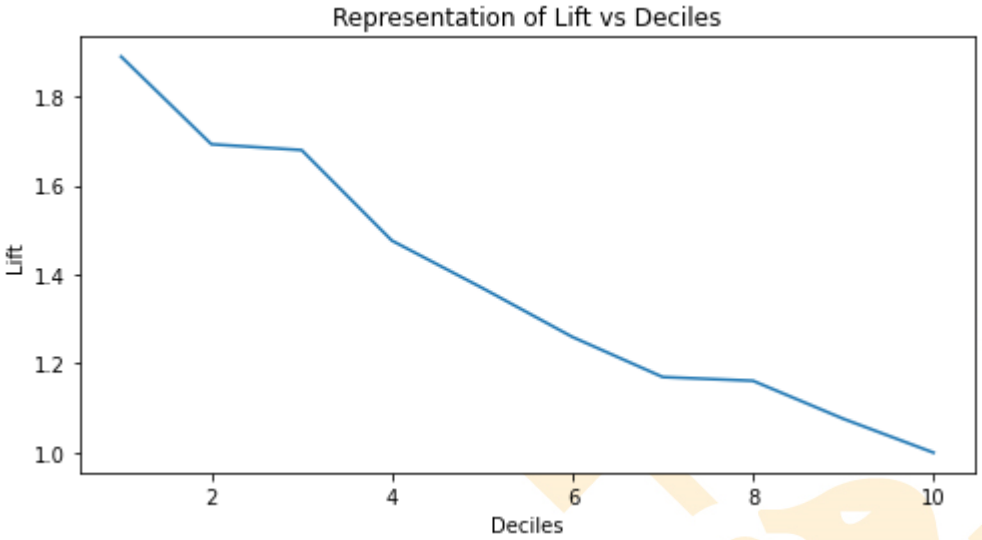
Estimation of lift per decile

Lift per decile

	decile	gain	gain_percentage	lift
0	1	24	18.897638	1.889764
1	2	19	33.858268	1.692913
2	3	21	50.393701	1.679790
3	4	11	59.055118	1.476378
4	5	12	68.503937	1.370079
5	6	9	75.590551	1.259843
6	7	8	81.889764	1.169854
7	8	14	92.913386	1.161417
8	9	5	96.850394	1.076115
9	10	4	100.000000	1.000000

Gain cumulative / percentage of sample
18.89/10%
33.85/20%
....

Graphic representation of lift vs decile



- The Lift of 1.889764 for the first decile, for example, means that when selecting the first 10% of the cases based on the model, one can expect 1.889764 times the total number of admitted found by randomly selecting 10%-of-records without a model.

Logistic Regression Model. Cases

- LM-1: Admission dataset. Use of LogisticRegression from sklearn.linear_model

```
from sklearn.linear_model import LogisticRegression
logit = LogisticRegression()

#Fitting model with X and Y values of dataset
logit.fit(X,Y)

pred_y = logit.predict_proba(X)
pred_y[0:6,]  # probabilty of each class
```

```
array([[0.82502055, 0.17497945],
       [0.69830844, 0.30169156],
       [0.30346652, 0.69653348],
       [0.80251107, 0.19748893],
       [0.86707922, 0.13292078],
       [0.61315912, 0.38684088]])
```

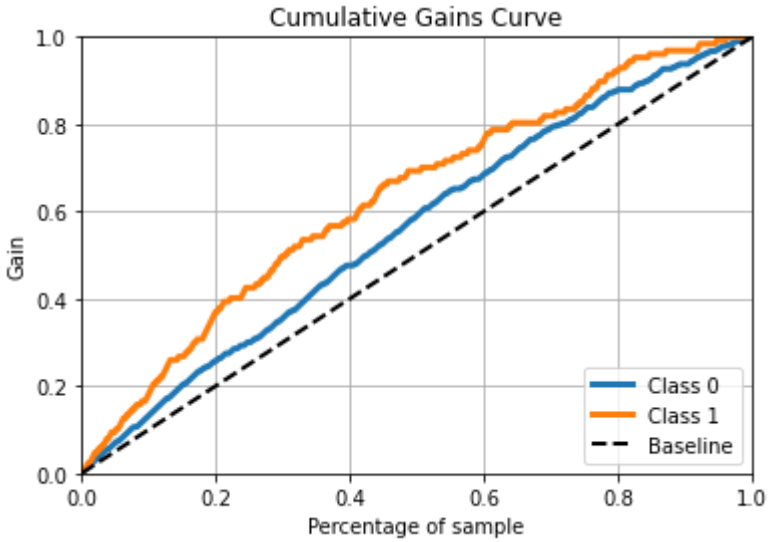
Probabilities for classes 0 and 1

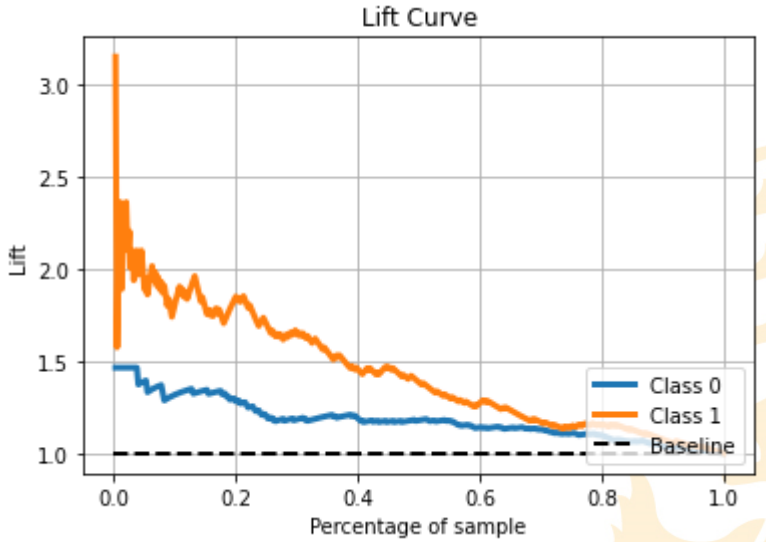
An alternative study without sorting by probabilities

Logistic Regression Model. Cases

- LM-1: Admission dataset. Use of LogisticRegression from sklearn.linear_model

```
import scikitplot as skplt
#following line is used to find Gains Curve
skplt.metrics.plot_cumulative_gain(Y,pred_y)
```

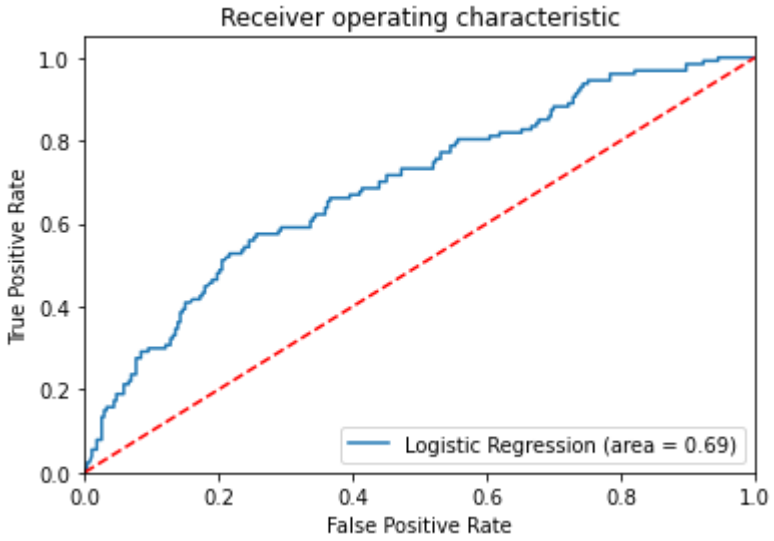




Logistic Regression Model. Cases

- LM-1: Admission dataset. Use of LogisticRegression from sklearn.linear_model

ROC curve for the whole dataset




```
AUC: 0.693
```

Another example is added to the material of this chapter analyzing data in bank.csv

comillas.edu