3.1 - Linear and Logistic Regression

Giovanni Della Lunga
giovanni.dellalunga@unibo.it

Introduction to Machine Learning for Finance

Bologna - February, 2022

Back to Linear Regression

Linear Regression

A linear model makes a prediction by simply computing a weighted sum of the input features, plus a constant called the **bias** term (also called the **intercept** term):

$$y = b + w_1 X_1 + w_2 X_2 + \cdots + w_m X_m + \epsilon \tag{1}$$

where:

- y is the predicted value (the value of the target);
- m is the number of features;
- X_i is the i^{th} feature value that are used to predict y;
- b and w_j are the j^{th} model parameters (b being the **bias** term and w_i the **weights**)
- ϵ is the predicton error.

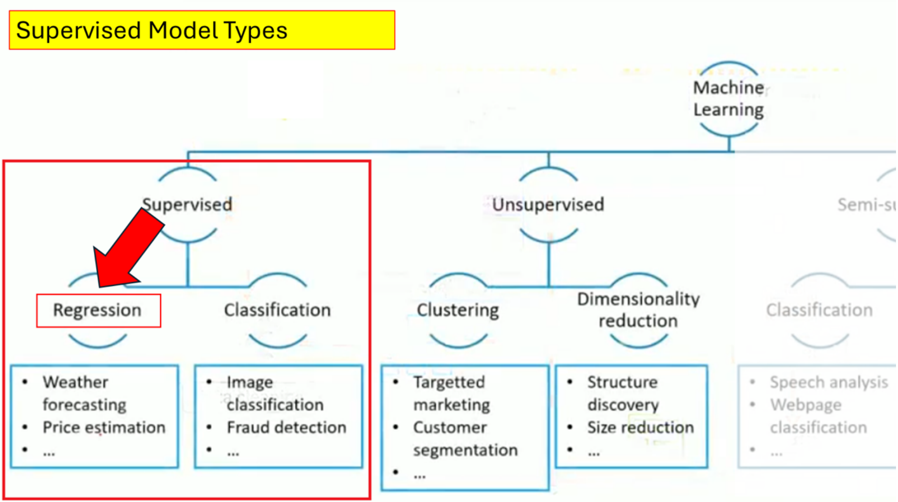
Linear Regression

- The parameters b and w_i are chosen to minimize the mean squared error over the training data set.
- This means that the task in linear regression is to find values for b and w_i that minimize

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y - b - w_1 X_{i1} - w_2 X_{i2} - \cdots - w_m X_{im})^2 \tag{2}$$

where n is the size of the training set.

Linear Regression



Iowa House Price Case Study

Linear Regression

The **Iowa House Pricing Dataset** from Kaggle is a popular dataset used for regression problems, particularly in the context of machine learning. It contains comprehensive information on houses in Ames, Iowa, and their sale prices, and it was designed to serve as an alternative to the Boston Housing Dataset. Here's a brief overview: **Dataset Overview**

- **Target Variable**: 'SalePrice' – the final price of the house in USD.
- **Number of Rows (Observations)**: 2,930 (combined training and test datasets).
- **Number of Features (Columns)**: 79 explanatory variables plus the target variable.

Iowa House Price Case Study

Linear Regression

- The objective is to predict the prices of house in Iowa from features
- 800 observations in training set, 600 in validation set, and 508 in test set
- Here the original competition description: **https://www.kaggle.com/c/house-prices-advanced-regression-techniques**

Iowa House Price Case Study

Linear Regression

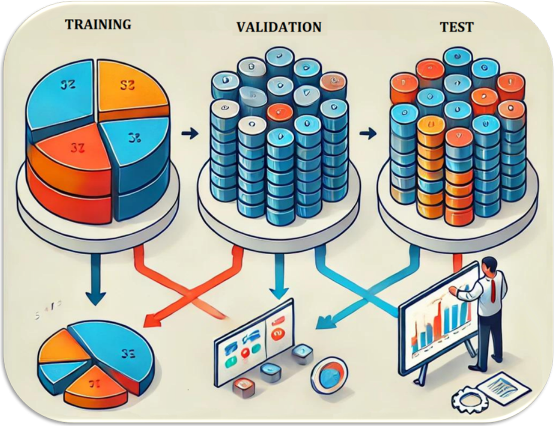
Types of Features The dataset includes a mix of:

- 1. **Numerical Features**:
  - Continuous (e.g., ‘LotArea‘, ‘GrLivArea‘, ‘SalePrice‘)
  - Discrete (e.g., ‘GarageCars‘, ‘TotRmsAbvGrd‘)
- 2. **Categorical Features**:
  - Nominal (e.g., ‘Neighborhood‘, ‘HouseStyle‘)
  - Ordinal (e.g., ‘ExterQual‘, ‘KitchenQual‘)
- 3. **Temporal Features**:
  - Year-based (e.g., ‘YearBuilt‘, ‘YrSold‘)
- 4. **Location Features**:
  - Specific location details (e.g., ‘Neighborhood‘, ‘MSSubClass‘).

Iowa House Price Case Study

Linear Regression

Dataset splitting: Training, Validation and Test



Iowa House Price Results

Linear Regression

INTERPOLATION RESULTS WITHOUT REGULARIZATION

intercept	-0.008295	Fireplaces	0.028258	MeadowV	-0.142466
LotArea	0.079	GarageCars	0.037997	Mitchel	-0.145749
OverallQual	0.214395	GarageArea	0.051809	Names	-0.093044
OverallCond	0.096479	WoodDeckSF	0.020834	NoRidge	0.333643
YearBuilt	0.160799	OpenPorchSF	0.034098	NPkVill	-0.216508
YearRemodAdd	0.025352	EnclosedPorch	0.006822	NriddgHt	0.534612
BsmtFinSF1	0.091466	Blmngtn	-0.169907	NWAmes	-0.225795
BsmtUnfSF	-0.03308	Blueste	-0.263946	OLDTown	-0.089516
TotalBsmtSF	0.138199	BrDale	-0.224482	SWISU	-0.020487
1stFlrSF	0.152786	BrkSide	0.120029	Sawyer	-0.074143
2ndFlrSF	0.132765	ClearCr	-0.045433	SawyerW	-0.127606
GrLivArea	0.161303	CollgCr	-0.013473	Somerst	0.120203
FullBath	-0.020808	Crawfor	0.221376	StoneBr	0.511099
HalfBath	0.017194	Edwards	0.007507	Timber	-0.008119
BedroomAbvGr	-0.08352	Gilbert	-0.024571	Veenker	0.036815
TotRmsAbvGrd	0.08322	IDOTRR	-0.000036	Bsmt Qual	0.011311

Ridge Regression

Linear Regression

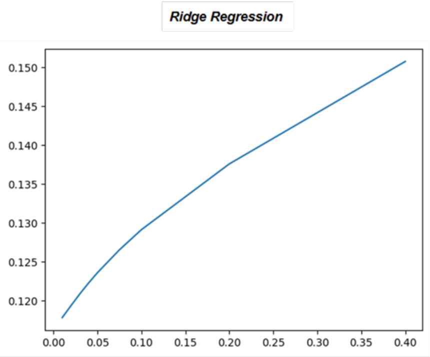
We try using Ridge regression with different values of the hyperparameter λ. The following code shows the effect of this parameter on the prediction error.

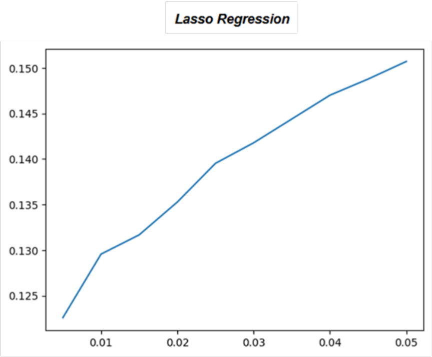
```
from sklearn.linear_model import Ridge  # Import the Ridge regression model from scikit-learn

# Define a list of regularization parameters (alphas) to test
# These values are scaled multiples of 1800 (e.g., 0.01 * 1800, 0.02 * 1800, etc.)
alphas = [0.01 * 1800, 0.02 * 1800, 0.03 * 1800, 0.04 * 1800,
          0.05 * 1800, 0.075 * 1800, 0.1 * 1800, 0.2 * 1800, 0.4 * 1800]
mses = []  # List to store the Mean Squared Error (MSE) for each alpha
# Iterate over each alpha value
for alpha in alphas:
    # Initialize the Ridge regression model with the current alpha value
    ridge = Ridge(alpha=alpha)
    # Train the Ridge regression model on the training dataset
    ridge.fit(X_train, y_train)
    # Predict the target values for the validation dataset
    pred = ridge.predict(X_val)
    # Compute the Mean Squared Error (MSE) for the validation predictions
    mse_val = mse(y_val, pred)  # mse function is assumed to be defined elsewhere
    # Append the computed MSE to the mses list
    mses.append(mse_val)
    # Print the MSE for the current model
    print(mse_val)
```

Ridge and Lasso Regression

Linear Regression

Error analysis using Ridge and Lasso regression with different values of the hyperparameter λ





As expected the prediction error increases as λ increases. Values of λ in the range 0 to 0.1 might be reasonably be considered because prediction errors increases only slightly when λ is in this range. However it turns out that the improvement in the model is quite small for these values of λ.

Iowa House Price Results

Linear Regression

Non-zero weights for Lasso when $\lambda = 0.1$ (overall quality and total living area were most important)

Feature	Weight
Lot Area (square feet)	0.04
Overall quality (Scale from 1 to 10)	0.30
Year built	0.05
Year remodeled	0.06
Finished basement (square feet)	0.12
Total basement (square feet)	0.10
First floor (square feet)	0.03
Living area (square feet)	0.30
Number of fireplaces	0.02
Parking spaces in garage	0.03
Garage area (square feet)	0.07
Neighborhoods (3 out of 25 non-zero)	0.01, 0.02, and 0.08
Basement quality	0.02

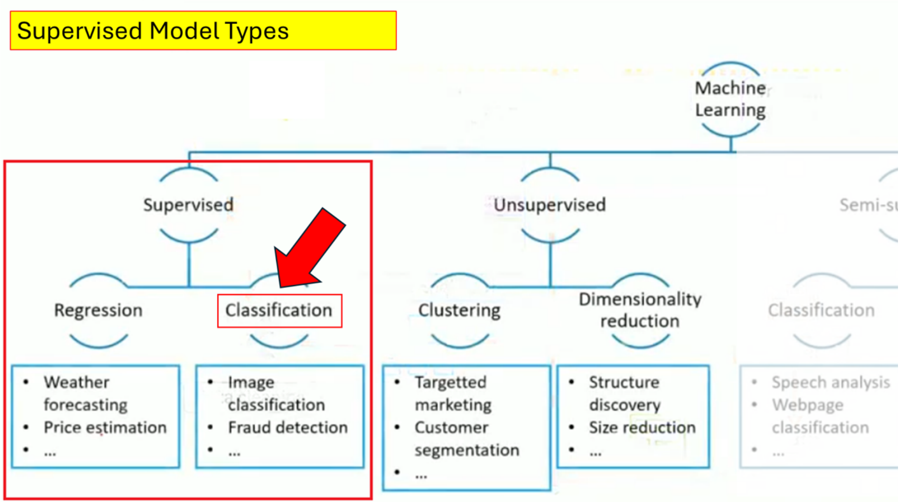
Summary of Iowa House Price Results

Linear Regression

- With no regularization correlation between features leads to some negative weights which we would expect to be positive
- Improvements from Ridge is modest
- Lasso leads to a much bigger improvement in this case
- Elastic net similar to Lasso in this case
- Mean squared error for test set for Lasso with $\lambda = 0.1$ is 14.7

Logistic Regression

Logistic Regression



Logistic Regression

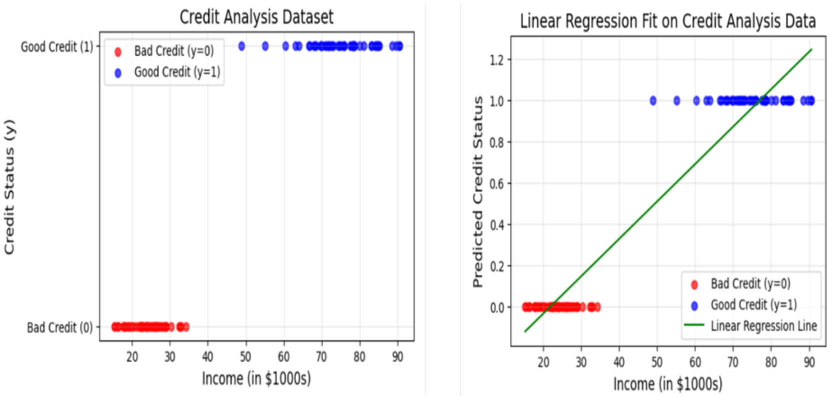
- The objective is to classify observations into a **positive outcome** and **negative outcome** using data on features
- Probability of a positive outcome is assumed to be a sigmoid function:

$$Q = \frac{1}{1+e^{-Y}} \tag{3}$$

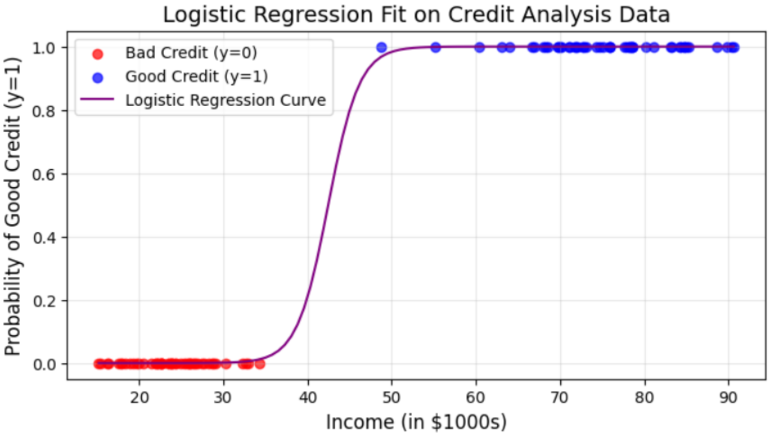
- where Y is related linearly to the values of the features:

$$Y = a + b_1X_1 + b_2X_2 + \cdots + b_mX_m \tag{4}$$

Logistic Regression



The Sigmoid Function



Maximum Likelihood Estimation

- We use the training set to maximize
- $$L = \sum_{\text{POS OUT}} \ln(Q) + \sum_{\text{NEG OUT}} \ln(1-Q) \tag{5}$$
- This cannot be maximized analytically but we can use a gradient ascent algorithm

Confusion Matrix

	p' (Predicted)	n' (Predicted)
p (Actual)	True Positive	False Negative
n (Actual)	False Positive	True Negative

Lending Club Case Study

- Data consists of loans made and whether they proved to be good or defaulted. (A restriction is that you do not have data for loans that were never made.)
- We use only four features
- Home ownership (rent vs. own)
- Income
- Debt to income
- Credit score
- Training set has 8,695 observations (7,196 good loans and 1,499 defaulting loans). Test set has 5,196 observations (4,858 good loans and 1,058 defaulting loans)

The Data

Home Ownership 1=owns, 0 =rents	Income ($'000)	Debt to Income (%)	Credit score	1=Good, 0=Default
1	44.304	18.47	690	0
1	136.000	20.63	670	1
0	38.500	33.73	660	0
1	88.000	5.32	660	1
	....			....
	.....			....

Results for Lending Club Training Set

- X_1= Home Ownership
- X_2= Income
- X_3= Debt to income ratio
- X_4 = Credit score
- $$Y = -6.5645 + 0.1395 \cdot X_1 + 0.0041 \cdot X_2 - 0.0011 \cdot X_3 + 0.0113 \cdot X_4$$

Decision Criterion

- The data set is imbalanced with more good loans than defaulting loans
- There are procedures for creating a balanced data set
- With a balanced data set we could classify an observation as positive if $Q > 0.5$ and negative otherwise
- However this does not consider the cost of misclassifying a bad loan and the lost profit from misclassifying a good loan
- A better approach is to investigate different thresholds, Z
- If $Q > Z$ we accept a loan
- If $Q \leq Z$ we reject the loan

Test Results

See Hull, Tables 3.10, 3.11, and 3.12

Z = 0.75:

	Predict no default	Predict default
Outcome positive (no default)	77.59%	4.53%
Outcome negative (default)	16.26%	1.62%

Z=0.80:

	Predict no default	Predict default
Outcome positive (no default)	55.34%	26.77%
Outcome negative (default)	9.75%	8.13%

Z=0.85:

	Predict no default	Predict default
Outcome positive (no default)	28.65%	53.47%
Outcome negative (default)	3.74%	14.15%

The Confusion matrix and common ratios

	Predict positive outcome	Predict negative outcome
Outcome positive	TP	FN
Outcome negative	FP	TN

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}}$$

$$\text{True Positive Rate (TPR also called sensitivity or recall)} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{The True Negative rate(also called specificity)} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

$$\text{The False Positive Rate} = \frac{\text{FP}}{\text{TN} + \text{FP}}$$

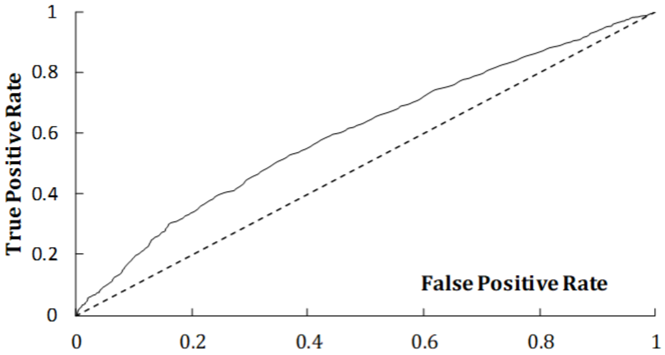
$$\text{Precision, P} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{F score} = 2 \times \frac{\text{P} \times \text{TPR}}{\text{P} + \text{TPR}}$$

Test Set Ratios for different Z values

	Z = 0.75	**Z = 0.80**	**Z = 0.85**
Accuracy	79.21%	63.47%	42.80%
True Positive Rate	94.48%	67.39%	34.89%
True Negative Rate	9.07%	45.46%	79.11%
False Positive Rate	90.93%	54.54%	20.89%
Precision	82.67%	85.02%	88.47%
F-score	88.18%	75.19%	50.04%

As we change the Z criterion we get an ROC



Area Under Curve (AUC)

- The area under the curve is a popular way of summarizing the predictive ability of a model to estimate a binary variable
- When $AUC = 1$ the model is perfect.
- When $AUC = 0.5$ the model has no predictive ability
- When $AUC < 0.5$ the model is worse than random
- In this case $AUC = 0.6020$

Choosing Z

- The value of Z can be based on
- The expected profit from a loan that is good, P
- The expected loss from a loan that defaults, L
- We need to maximize: $P \times \text{TP} - L \times \text{FP}$