COMP47350: Data Analytics (Conv)

Dr. Georgiana Ifrim
georgiana.ifrim@ucd.ie

Insight Centre for Data Analytics
School of Computer Science
University College Dublin

2018/19

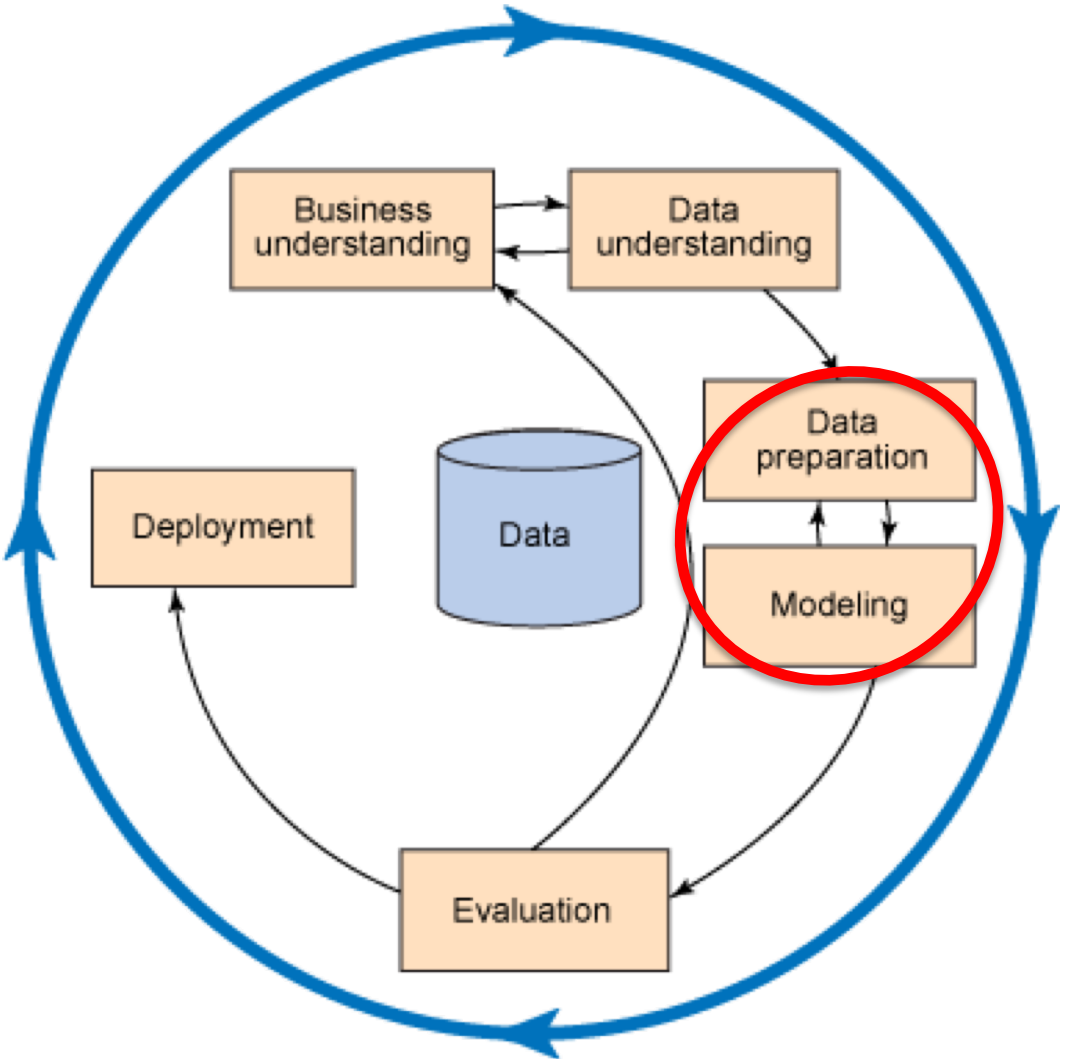
Module Topics

- **Python Environment** (Anaconda, Jupyter Notebook)
- **Getting Data** (Web scrapping, APIs, DBs)
- **Understanding Data** (slicing, visualisation)
- **Preparing Data** (cleaning, transformation)
- **Modeling & Evaluation** (machine learning)

Data Analytics Project Lifecycle:

CRISP-DM

CRISP-DM: **CR**oss-**I**ndustry **S**tandard **P**rocess for **D**ata **M**ining



Modeling Data

- **Modeling:**
  - How to build prediction models
  - How to evaluate prediction models

Supervised Learning Problems:

- **Regression:** predicting a numeric target feature
- **Classification:** predicting a categorical target feature

Modeling Data

Types of Learning	Continuous target feature	Categorical target feature
Supervised (target feature given)	**regression**	**classification**
Unsupervised (no explicit target feature)	dimension reduction	clustering

Modeling Data

- Modeling:
  - How to build prediction models (train, test)
  - How to evaluate prediction models

We study three supervised machine learning methods:

- **Linear Regression** (linear model: assumes linear relationship between features and target)
- **Logistic Regression** (linear model)
- **Random Forests** (non-linear model: no linearity assumption)

Supervised Machine Learning

- Learn to approximate (**model**) the relationship between a set of **descriptive features** and a **target feature**

- Different types of learning strategies (textbook):
  - **Error-based** (Linear Regression, Logistic Regression, Support Vector Machines, Neural Networks)
  - **Information-based** (Decision Trees, Random Forests)
  - **Similarity-based** (Nearest Neighbors)
  - **Probability-based** (Naïve Bayes)

Error-based Learning

- **Linear regression**
  - Widely used prediction model
  - Assumes:
    - numeric (descriptive and target) features
    - **linear relationship** between descriptive features and target feature
  - Predictive model described by a set of **parameters** also known as **weights** (e.g., w_0, w_1, ..., w_n, where n is the number of features)
  - Prediction function:

target_value = **w_0** + **w_1** * feature_1 + **w_2** * feature_2 + ….+ **w_n** * feature_n

Linear Regression

Training the model:

- Initialize model parameters, e.g., w_0 = 0, w_1 =0
- Error function to measure how good the model is (e.g., is prediction close to the actual target)
- Training set
- Iteratively adjust parameters using training set and error function, e.g., w_0 = 0.5, w_1 = 0.7

Testing the model:

- Use learned parameters (w_0, w_1) to make predictions on new samples

Linear Regression

- A **parameterized** prediction model: start with a set of random parameter values, e.g., w_i=0 (for all features)
- **Error function** measures how well this initial model performs when making predictions for examples in a **training dataset**
- Based on the value of the error function the **parameters are iteratively adjusted** to create a more and more accurate model

Linear Regression: Example

Table: The **office rentals dataset**: a dataset that includes office rental prices and a number of descriptive features for 10 Dublin city-centre offices.

ID	Size	Floor	Broadband Rate	Energy Rating	Rental Price
1	500	4	8	C	320
2	550	7	50	A	380
3	620	9	7	A	400
4	630	5	24	B	390
5	665	8	100	C	385
6	700	4	8	B	410
7	770	10	7	B	480
8	880	12	50	A	600
9	920	14	8	C	570
10	1,000	9	24	B	620

Can we predict the **RENTAL PRICE** (target outcome), given the descriptive features for an office? (10 training examples, 4 features)

Linear Regression: Simple Example

Table: The **office rentals dataset**: a dataset that includes office rental prices and a number of descriptive features for 10 Dublin city-centre offices.

ID	Size	Rental Price
1	500	320
2	550	380
3	620	400
4	630	390
5	665	385
6	700	410
7	770	480
8	880	600
9	920	570
10	1,000	620

Can we predict the **RENTAL PRICE** (target outcome), given the descriptive feature **SIZE**?

Linear Regression: Simple Example

ID	Size	Rental Price
1	500	320
2	550	380
3	620	400
4	630	390
5	665	385
6	700	410
7	770	480
8	880	600
9	920	570
10	1,000	620

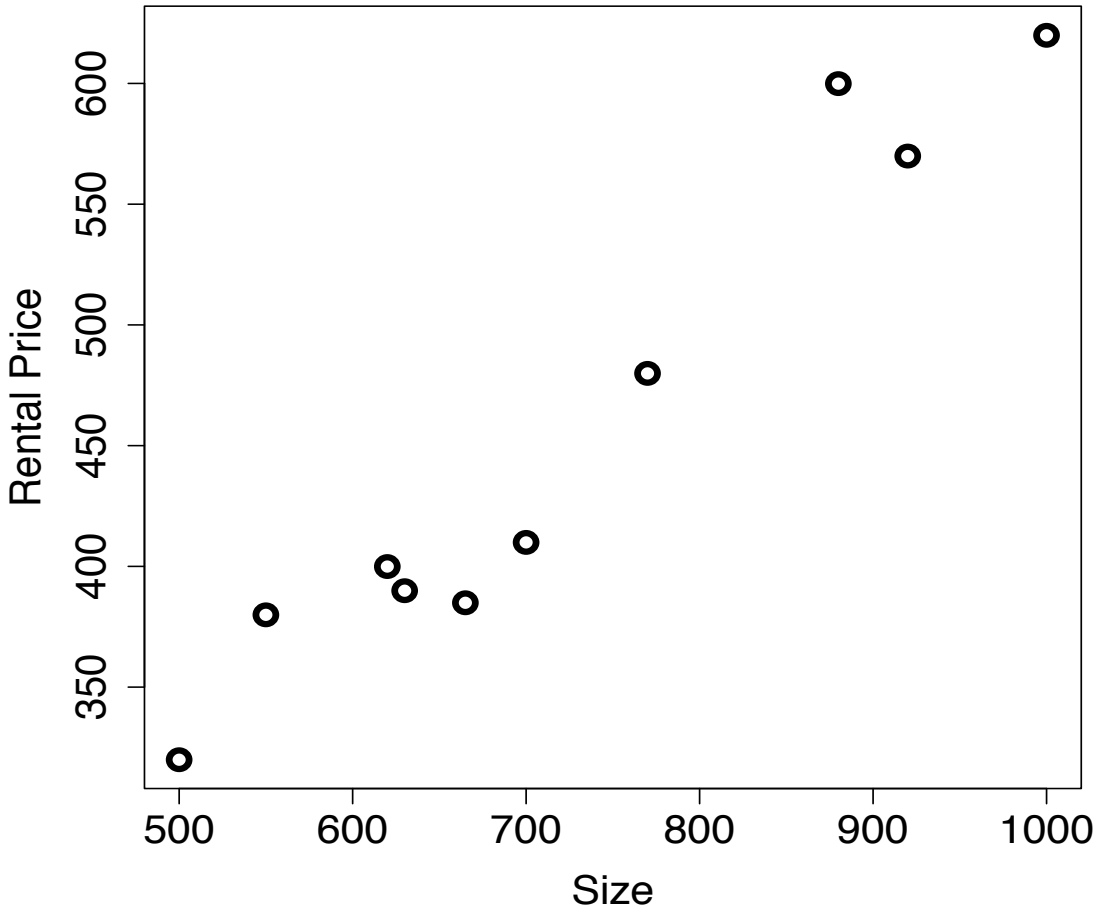


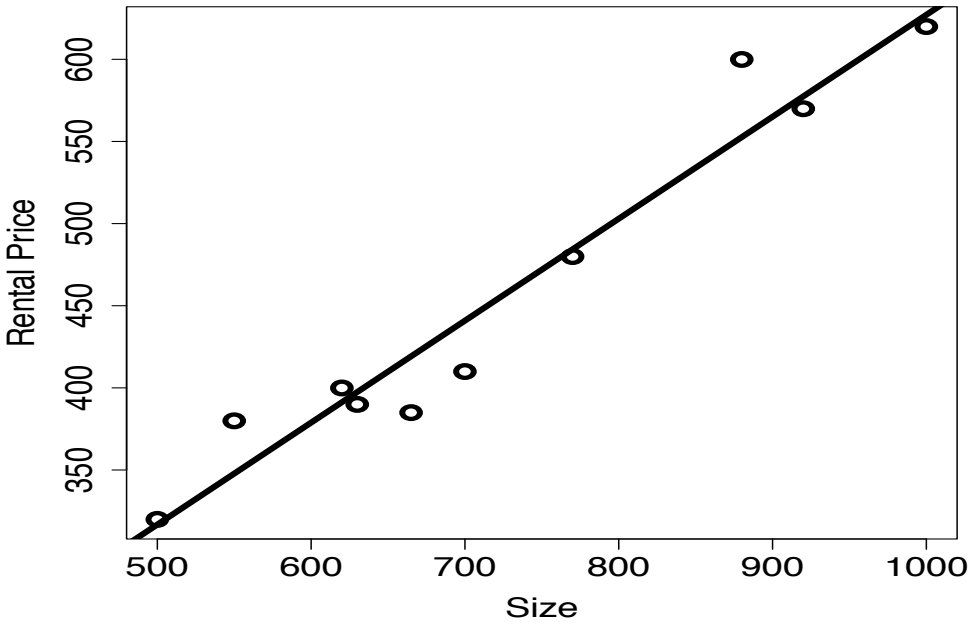
Figure: A scatter plot of the SIZE and RENTAL PRICE features from the office rentals dataset.

Linear Regression: Linear Model

- Scatter plot shows linear relationship between SIZE and RENTAL PRICE
- This relationship can be approximately captured via a parameterized line
- The equation of a line can be written as:
  $y = b + m*x$
- (x,y) are data points; **m** and **b** are parameters: **m** is the slope, **b** is the bias (aka y-intercept)

Linear Regression Model

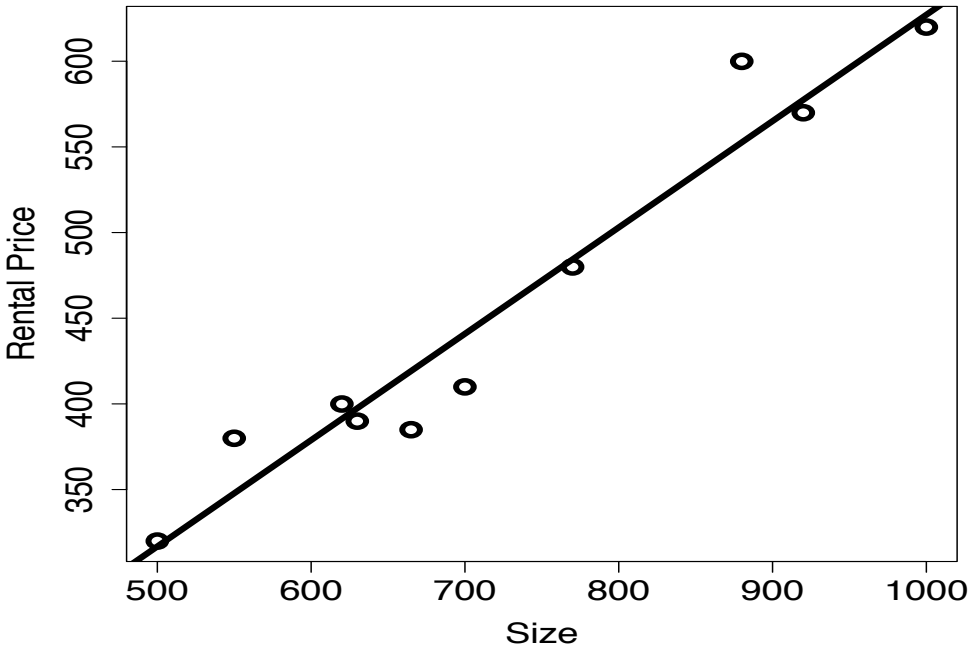
- Same scatter plot as before with a simple linear model added to capture the relationship between SIZE and RENTAL PRICE
- The model is: $\text{RENTAL PRICE} = 6.47 + 0.62 \times \text{SIZE}$



Linear Regression Model

- **Interpretation:** For an increase of a square foot in SIZE, the RENTAL PRICE increases by 0.62 euro.

$$\text{RENTAL PRICE} = 6.47 + 0.62 \times \text{SIZE}$$



Linear Regression Prediction

- Using the previous trained model, we can **predict** the rental price for a **new office** size

$$\text{RENTAL PRICE} = 6.47 + 0.62 \times \text{SIZE}$$

- Model parameters: b = 6.47; m= 0.62
- What is the expected rental price for a new example, e.g., for a 730 square foot office?

Linear Regression Prediction

- What is the expected rental price for a new/test 730 square foot office, using the previous model?

$$\text{RENTAL PRICE} = 6.47 + 0.62 \times \text{SIZE}$$

$$\begin{aligned} \text{RENTAL PRICE} &= 6.47 + 0.62 \times 730 \\ &= 459.07 \end{aligned}$$

- This model is known as **simple linear regression** (1 descriptive feature, 1 target feature)

Linear Regression: Simple vs Multivariable Regression

- Simple linear regression model can be written as:

predicted_value = **b** + **m** * feature_value

- We can refer to b and m as weights **w**:

predicted_value = **w_0** + **w_1** * feature_value

- This allows us to extend the model to use more features:

predicted_value = **w_0** + **w_1** * feature_1 + **w_2** * feature_2 + ….+ **w_n** * feature_n

Evaluation: Error Function

- We need an approach to search for the best parameters for the given training data, e.g., different choices of **w** lead to different predictions and thus prediction errors
- We use an error function: a way to capture the gap between the predictions made by the model and the actual values in the training dataset
- Most common error function: **sum of squared errors**

Linear Regression: Evaluation

- Error between prediction and actual value:

  predicted_value = w_0 + w_1 * feature_1

- Error on single example:

  error = actual_value – predicted_value

- Sum of squared errors: sum the error over all training samples

  Total error = (error_on_example_1)^2 + (error_on_example_2)^2 + …+ (error_on_example_m)^2

Linear Regression: Measuring Error

Table: Calculating the sum of squared errors for the candidate model (with $\mathbf{w}[0] = 6.47$ and $\mathbf{w}[1] = 0.62$) making predictions for the the office rentals dataset.

ID	Size	Rental Price
1	500	320
2	550	380
3	620	400
4	630	390
5	665	385
6	700	410
7	770	480
8	880	600
9	920	570
10	1,000	620

ID	Rental Price	Model Prediction	Error Error	Squared Error
1	320	316.79	3.21	10.32
2	380	347.82	32.18	1,035.62
3	400	391.26	8.74	76.32
4	390	397.47	-7.47	55.80
5	385	419.19	-34.19	1,169.13
6	410	440.91	-30.91	955.73
7	480	484.36	-4.36	19.01
8	600	552.63	47.37	2,243.90
9	570	577.46	-7.46	55.59
10	620	627.11	-7.11	50.51
			Sum	**5,671.64**
		Sum of squared errors (Sum/2)		**2,835.82**

Linear Regression Lines

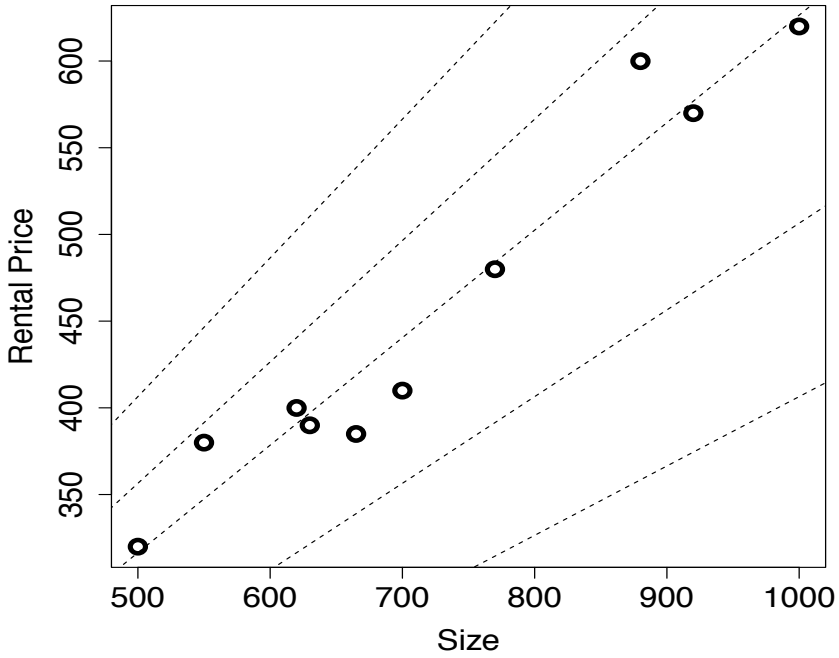


Figure: A scatter plot of the SIZE and RENTAL PRICE features from the office rentals dataset. A collection of possible simple linear regression models capturing the relationship between these two features are also shown. For all models **w**[0] is set to 6.47. From top to bottom the models use 0.4, 0.5, 0.62, 0.7 and 0.8 respectively for **w**[1].

Sum of Squared Errors

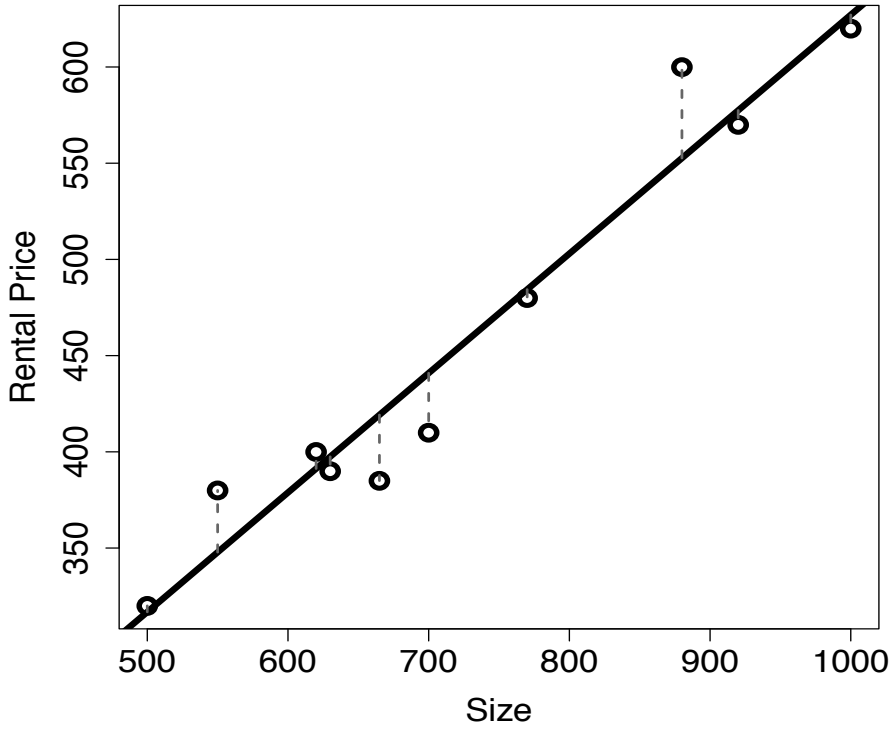
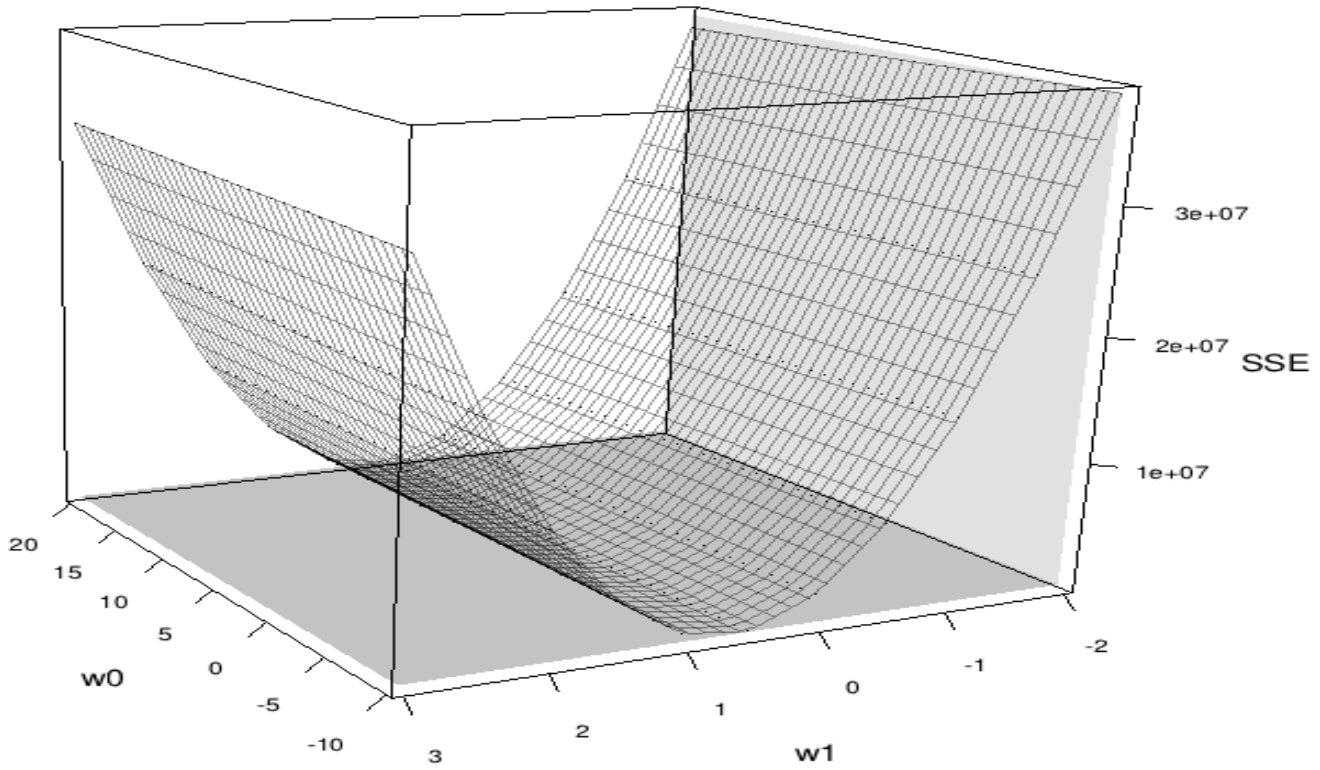


Figure: A scatter plot of the SIZE and RENTAL PRICE features from the office rentals dataset showing a candidate prediction model (with $\mathbf{w}[0] = 6.47$ and $\mathbf{w}[1] = 0.62$) and the resulting errors.

Visualizing Error

For every combination of weights, **w**[0] and **w**[1], there is a corresponding sum of squared errors value that can be plotted (**error surface**). The model that best fits the training data is the model corresponding to the lowest point on the error surface.



Visualizing Model Fitting

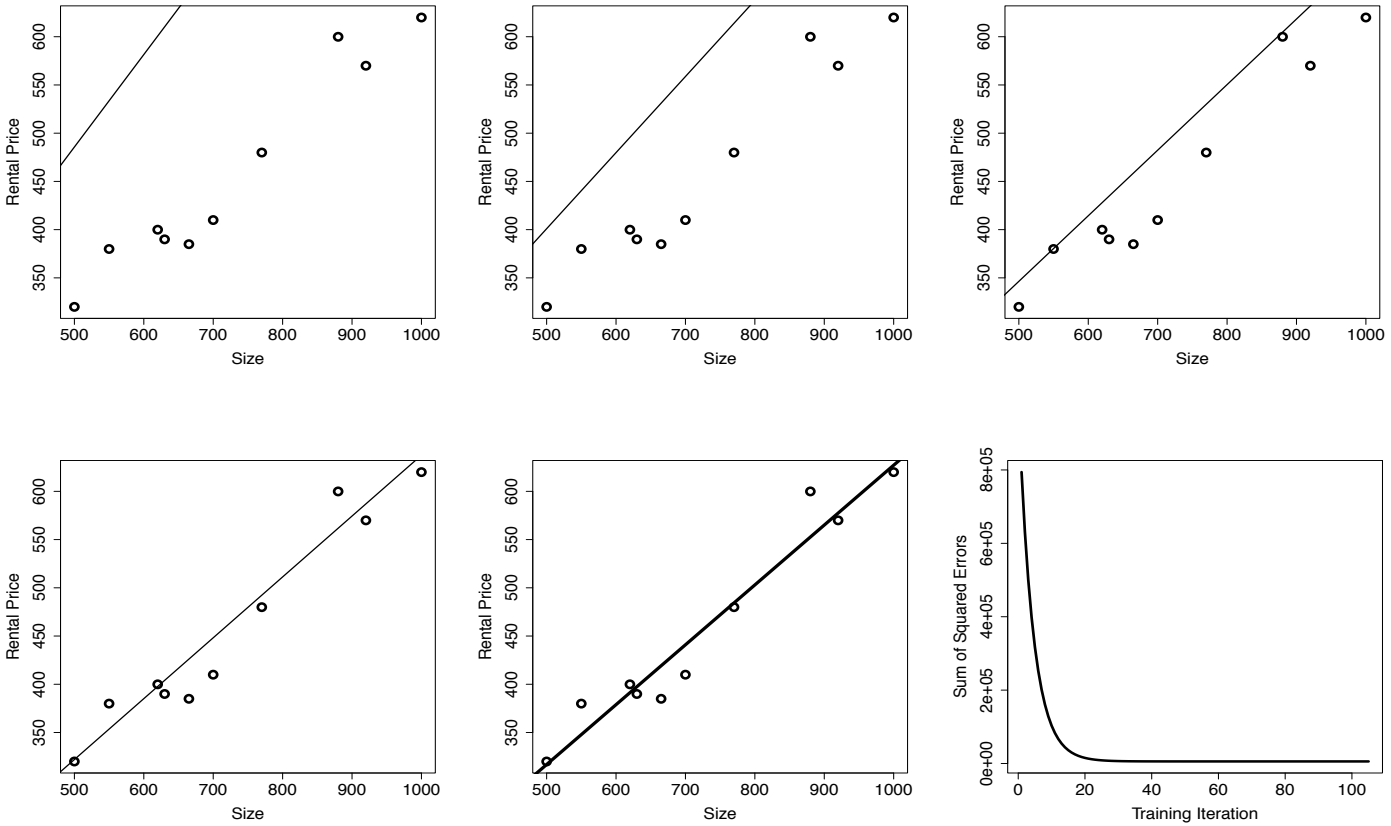


Figure: A selection of the simple linear regression models developed during the gradient descent process for the office rentals dataset. The final panel shows the sum of squared error values generated during the gradient descent process.

References

- **FMLPDA Book: Fundamentals of Machine Learning for Predictive Data Analytics**, John D. Kelleher, Brian Mac Namee, Aoife D'Arcy, MIT Press, 2015 (machinelearningbook.com)
- Chapter3 from **An Introduction to Statistical Learning**, by G. James, D. Witten, T. Hastie, R. Tibshirani, Springer, 2016 (**free book:** http://www-bcf.usc.edu/~gareth/ISL/)
- A friendly introduction to linear regression (using Python):

http://www.dataschool.io/linear-regression-in-python/