Linear Regression

Department of Computing and Mathematical Sciences
University of Greenwich, London

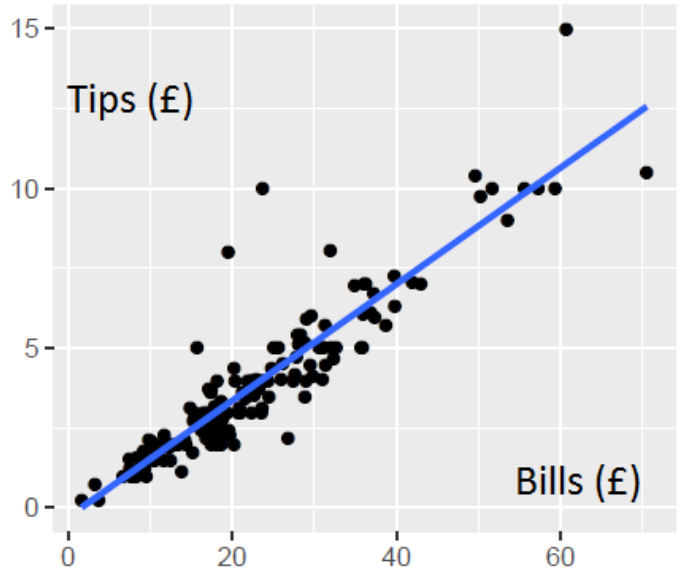
Feb 12, 2021

Outline

Linear Regression

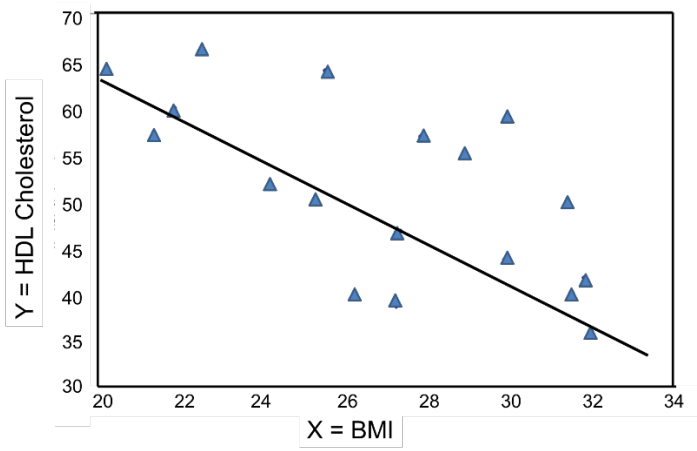
A restaurant may want to study the bill and tipping patterns of its customers.

Or more specifically wants to find on average, 1£ increase in the Bill is likely to increase Tips by how much?



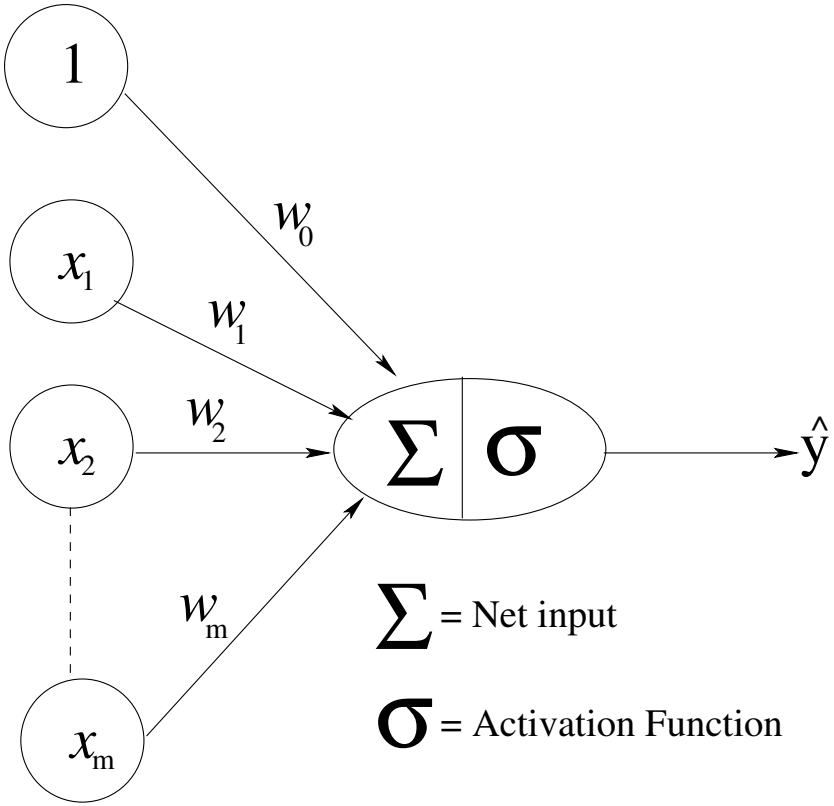
A study wants to determine whether higher BMI relates to having a lower HDL Cholesterol (good Cholesterol).

Or more specifically wants to find on average, a unit increase in the Body Mass index is likely to lower HDL by how much?



Linear Regression

Suppose, a sample, X contains n records - each with m features, and a target variable, y. A **supervised** regression task's goal is to predict the target variable (continuous) given the features.



- Perceptron:
  - Activation function (σ) is the threshold function, e.g.,

$$\sigma(z) = \begin{cases} 1, & z \geq 0 \\ -1, & \text{otherwise} \end{cases}$$

where, $z = \sum_{j=0}^{m} w_j x_j$, and $x_0 = 1$.

  - Output, $\hat{y} \in \{0, 1\}$ - computed with an additional unit.
- Linear Regression:
  - Activation function (σ) is the linear function, i.e., $\sigma(z) = z$.
  - Output, $\hat{y} \in \mathbb{R}$ (real numbers).

Linear Regression

Regression Task: Compute the weights w in a way that the computed output $\hat{y}$ approximates the target (y) well.

Naive approach:

- Step 1: Randomly select the weights, w, i.e., $w_0, w_1, \ldots, w_m$.
- Step 2: Compute the output, $\hat{y} = \sigma(z) = \sum_{j=0}^{m} w_j x_j$.
- Step 3: Check whether $\hat{y}$ closely matches the target, y, e.g., compute a distance measure, i.e., $(y - \hat{y})^2$.
- Step 4: Repeat Step 2 and 3 for all samples, $i = 1, 2, \ldots, n$ and take the average distance measure.
- Repeat the whole process [from Step 1] k times, and select the best set of weights (among k iterations) giving the minimum average distance.

(Least Squares) Linear Regression

The procedure of previous slide is not practical.
For example, say a weight can take only integers from 0, 1, . . . , 9.
For 5 features ($m = 5$), we end up comparing $k = 10^6$ (including w_0) set of weights for finding the best one.
In practice, each weight, $w_j \in \mathbb{R}$ (real numbers). Therefore, arbitrarily selecting a subset k, will not find the optimum solution.

It actually has a closed form solution:

$$w = (X^T X)^{-1} X^T Y \tag{1}$$

For detailed calculation of deriving Eq.(1), please refer to any **Statistics** book covering 'Linear Regression'.

(Least Squares) Linear Regression

The closed form solution, Eq.(1) requires a few matrix multiplications and also matrix inversion.

Computationally, it can be quite expensive especially for big data where we deal with large number of records n and features m.

There is a better way (better suited for big data scenarios):

- We change the weight, w a little, and observe the improvement/loss of the y's 'prediction' ($\hat{y}$) performance.
- We update the weight in the direction of the improvement.
- We stop when there is no more improvement [or sometimes, the maximum iteration step has been reached].

In the following slides, we discuss this algorithm (Gradient Descent).

(Least Squares) Linear Regression – Gradient Descent

The goal now is to find the set of weights that minimise,

$\mathcal{L}(w) = \frac{1}{2n}\sum_{i=1}^{n}(y^{[i]} - \hat{y}^{[i]})^2$, where $\hat{y}^{[i]} = \sum_{j=0}^{m} w_j{}^{[i]} x_j{}^{[i]}$.

$\mathcal{L}(w)$ is called the loss function.

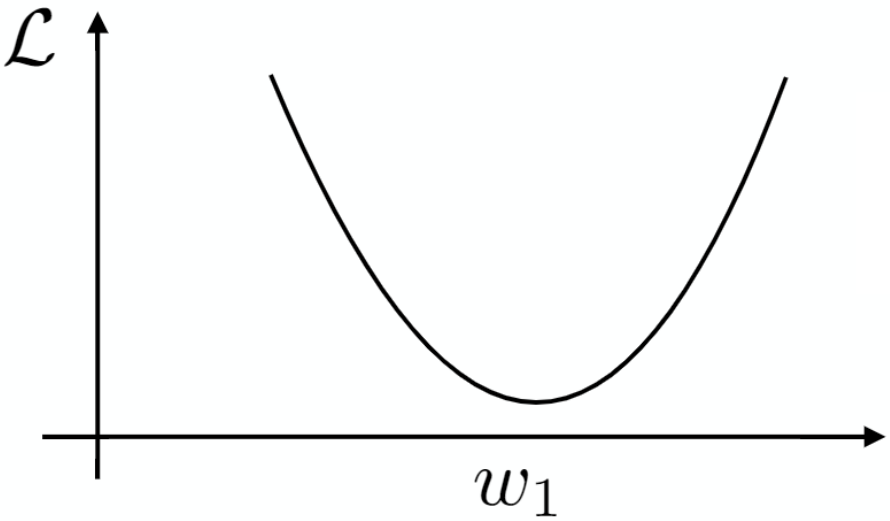


Figure: Quadratic Loss (Convex)

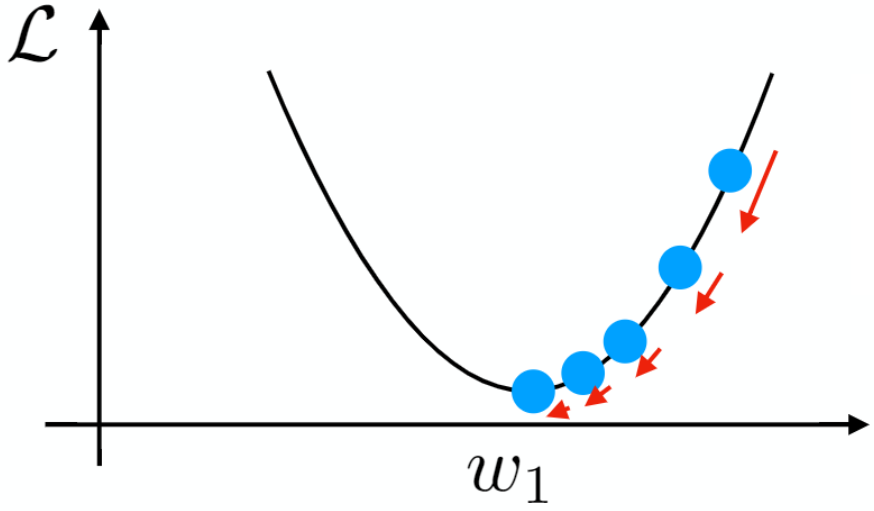


Figure: Updating Weight, w_1

(Least Squares) Linear Regression – Gradient Descent

Initialise the weights, w

For every training epoch:

For every record i, $< X^{[i]}, y^{[i]} >$:

(a) $\hat{y}^{[i]} = \sigma(\sum_{j=0}^{m} w_j x_j)$

For each weight index, $j \in \{0, 1, \ldots\ldots m\}$:

(b) $\frac{\partial \mathcal{L}}{\partial w_j} = -(y^{[i]} - \hat{y}^{[i]}) x_j{}^{[i]}$

(c) $w_j = w_j + \eta \left(-\frac{\partial \mathcal{L}}{\partial w_j} \right)$, where $\eta =$ learning rate

End of procedure when the weights from one epoch to the next do not change or maximum number of epochs are reached.

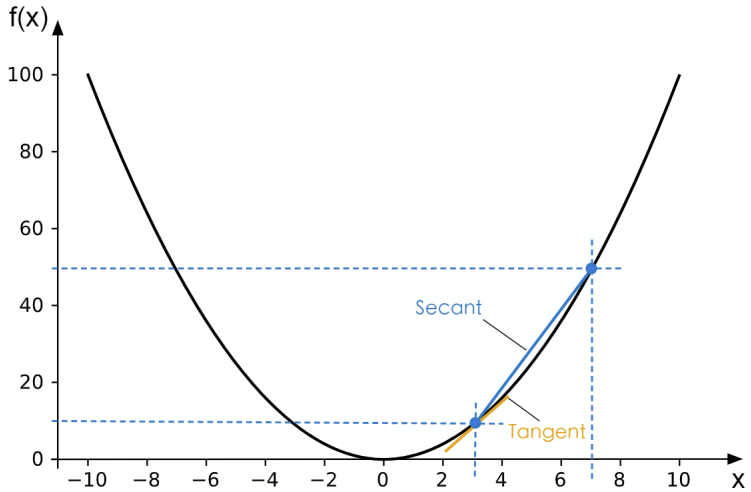


Figure: Graphical Representation of Derivatives

$$\frac{dy}{dx} = \lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

(Least Squares) Linear Regression – Gradient Descent

Choosing learning rate, η.

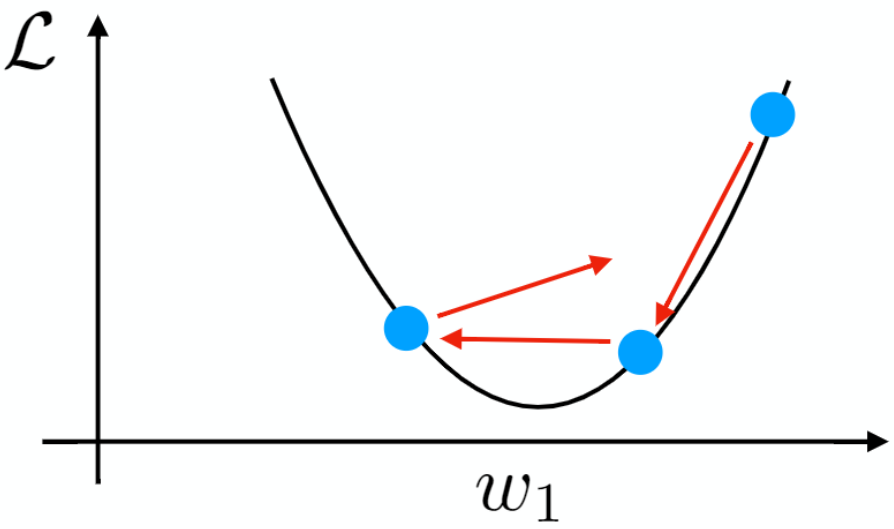


Figure: η = too large

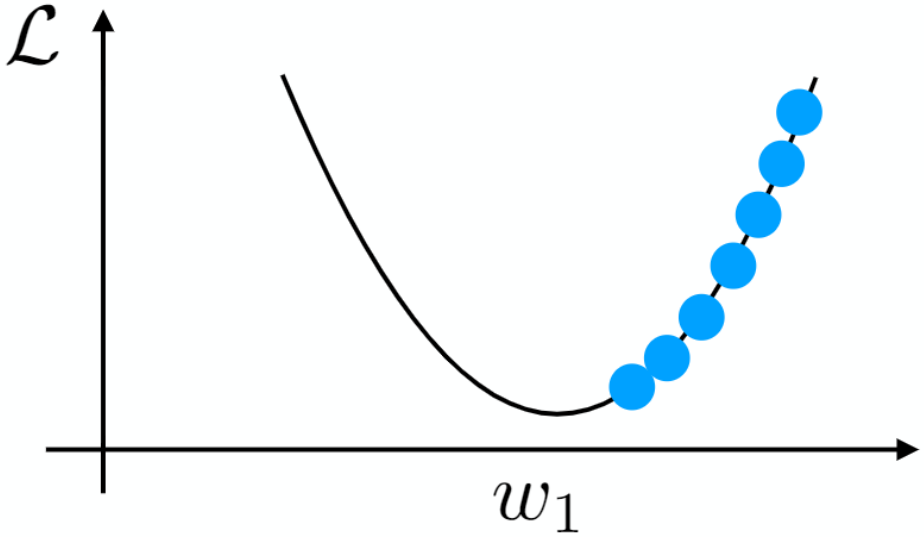


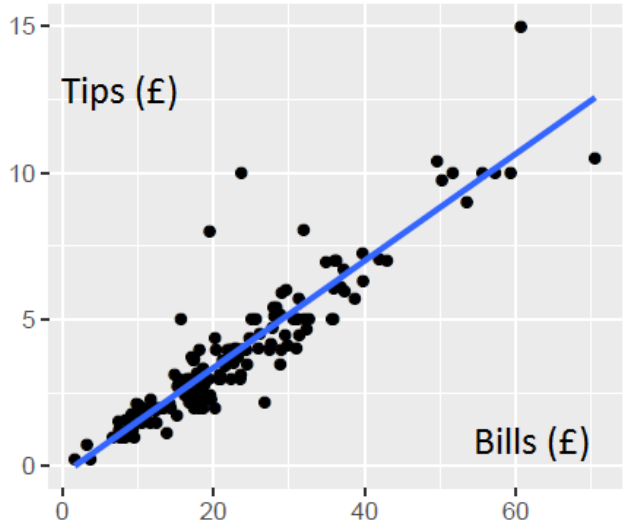
Figure: η = too small

Hyperparameters of Linear Regression? Difference with Model Parameters?
Scikit-Learn Linear Regression Documentation Link here. Its Implementation Details Link here.

Linear Regression – Interpretation

Remember the restaurant example used is Slide 3? For example, they wanted to find on average, 1£ increase in the Bill is likely to increase Tips by how much?

Suppose, the solution after using the Linear Regression algorithm:
Tips (y) = 0.18 * Bills (x) - 0.29.
Looking at the solution, we can see that on average, 1£ increase in Bill will likely to give an extra 18 pence in Tips.



Furthermore, there are many other elegant statistical analysis can be performed. We will see a few examples based on our notebook here.

Linear Regression – Evaluation Metrics

Measure of goodness of fit and how well the model performs.

- MAE (Mean Absolute Error):

  $\frac{1}{n_{train}} \sum_{i=1}^{n_{\text{train}}} |y^{[i]} - \hat{y}^{[i]}|$ (Training Set),

  and $\frac{1}{n_{test}} \sum_{i=1}^{n_{\text{test}}} |y^{[i]} - \hat{y}^{[i]}|$ (Testing Set).

- RMSE (Root Mean Square Error): Compute both for training and testing as above using the formula, $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y^{[i]} - \hat{y}^{[i]})^2}$

- R2 score (r^2): $r^2 \leq 1$

  Explained variance, $r^2 = 1 - \frac{\sum_{i=1}^{n}(y^{[i]} - \hat{y}^{[i]})^2}{\sum_{i=1}^{n}(y^{[i]} - \bar{y})^2}$.

  In scikit-learn library's Linear Regression implementation, **score** indicates this metric. More details in our notebook here.

Linear Regression – Regularization

- L2 Regularization (Ridge) – loss function becomes:

$$\mathcal{L}(w) = \frac{1}{2n}\sum_{i=1}^{n}(y^{[i]} - \hat{y}^{[i]})^2 + \lambda \sum_{j=0}^{m} {w_j}^2,$$

where $\hat{y}^{[i]} = \sum_{j=0}^{m} w_j{}^{[i]} x_j{}^{[i]}$.

- L1 Regularization (Lasso) – loss function becomes:

$$\mathcal{L}(w) = \frac{1}{2n}\sum_{i=1}^{n}(y^{[i]} - \hat{y}^{[i]})^2 + \lambda \sum_{j=0}^{m} |w_j|.$$

λ is the penalty parameter. $\lambda = 0$ yields usual Linear Regression.
In scikit-learn library's implementation, λ is denoted by α (alpha) - more details here.

Linear Regression – Regularization

The goal remains the same: find optimum set of weights, w but with an additional loss term (+hyperparameter, λ).

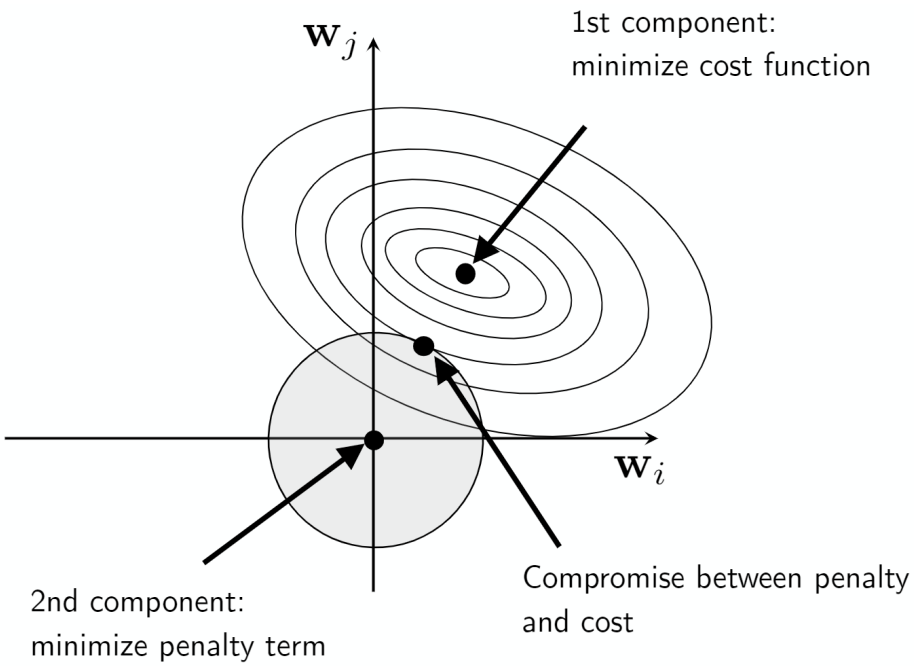


Figure: Ridge [RM19]

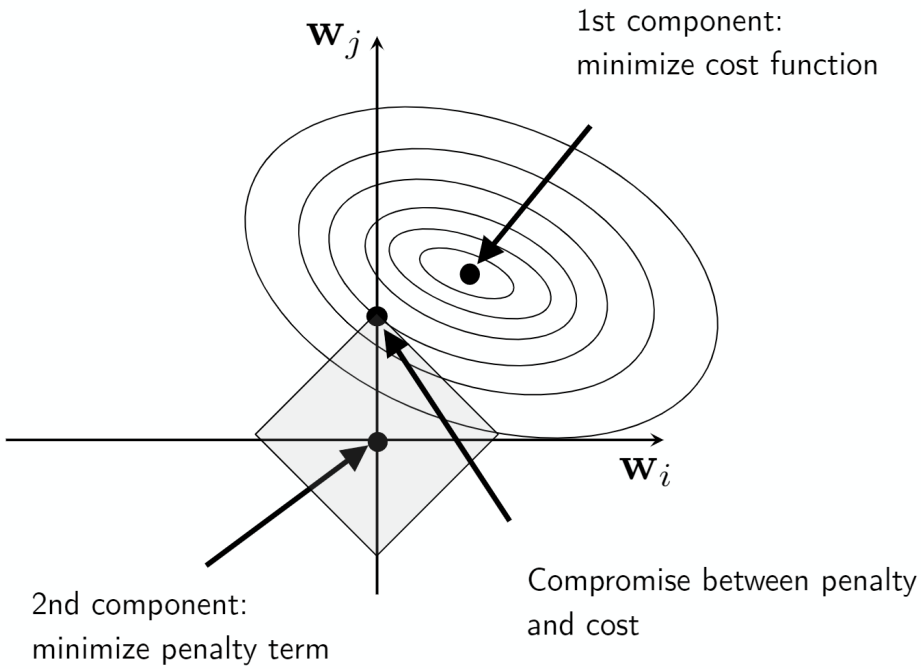


Figure: Lasso [RM19]

$\lambda \uparrow =$ Regularization strength/Penalty against complexity/Weight Shrinkage.

Linear Regression – Regularization

- Suppose, a solution with 3 features are of the form, $y = 0.5x_1 + 20x_2 + 0.5x_3 - 5$. w_2 is quite high which makes y to be quite dependent on x_2 compared to x_1 and x_3. Using ridge, this [collinearity/overfitting] can be restricted. For example, $y = 5.5x_1 + 10x_2 + 5.5x_3 - 5$ may be better?
- On the other hand, $\lambda \uparrow\uparrow$ can give rise to 'under-fitting'.
- Lasso encourages sparsity - some weights associated with features can be zero (feature selection?).
- Lasso is rarely useful and can easily give rise to under-fitting.
- In both regularization scenarios, you may need to find the optimum λ. GridSearch can be useful - we will demonstrate it in the next lecture using the notebook here.

Linear Regression – More Interpretation

At this point, please go through the notebook (here) based on Boston housing dataset.
We will address some use-case analysis here that might have motivated creating this notebook.

Use case 1: A real estate agency is looking to open a new office in a state of USA. They boast of a wealthy set of clients who are only interested in expensive houses, e.g., price $> \$400,000$. They do not mind moving to different states if they can find what they want (i.e., luxurious house). Having this in mind, the agency is exploring if Boston can be attractive to their customers. Then, they may consider opening an office in Boston. You (data scientist) are contacted and asked to find the probability of finding a house $> \$400,000$ in Boston. How do you go about it?

Linear Regression – More Interpretation

Many options. You may want to collect relevant data to come up with the conclusion (e.g., collecting house prices in Boston).

Or, you may already have the Boston dataset as used in our notebook. You can utilise statistical analysis on that sample alone.

Alternatively, you could approximate the Boston house price 'population' using an ML algorithm for the sample. This is a useful approach even for houses (those not yet in market/sample).

Using our prediction's distribution of mean=22.49 and std=8.11 - see the notebook's results (using Z-table: link here):

$$\begin{aligned} \text{Probability}, Pr(Price > 40) &= Pr(Z > (Price - 22.49)/8.11) \\ &= Pr(Z > (40 - 22.49)/8.11) \\ &= Pr(Z > 2.16) = 1 - 0.9846 = 1.54\% \end{aligned}$$

Only 1.54% probability of finding a house with price $> \$400,000$.

Linear Regression – More Interpretation

Use case 2 (Hypothesis Testing): Suppose you approximated the house price population mean (22.49) using ML as in our notebook. You declared (may be arrogantly) that this is the mean for house price overall in USA [claim taking all the house prices of different states as well - those that were not in the sample dataset]. Next, others may want to validate this claim. They may hire a data analyst whose job would be to collect more data and create a new sample to test this hypothesis (your claim). Since we do not have that luxury at this moment, we will test it against the same sample that we have for Boston (not ideal though! - the analyst should have used a different dataset preferably from a different state).

We will perform one sample two tailed Z-test. Refer to the statistics resources given in the end slide [Lan03, TtsmrdFP] in order to understand how this test might be suitable given our task. A brief tutorial on Z-test is here.

Linear Regression – More Interpretation

Our sample has 506 records with mean ($\bar{x} = 22.53$, and standard deviation, $\sigma = 9.2$).

Null Hypothesis (H_0): The two means – the claimed one (μ_0) for all house prices in USA) and the sample one are the same.
Alternative Hypothesis (H_1): They are not the same.
Compute the z-statistic:

$$z = \frac{(\bar{x} - \mu_0)}{\frac{\sigma}{\sqrt{n}}} = \frac{(22.53 - 22.49)}{\frac{9.2}{\sqrt{506}}} = 0.098.$$

Inspecting the z-statistic value across different significance levels [summarised table can be found here], we cannot reject the Null Hypothesis. In other words, the sample's (Boston) house price follows the claimed average [not different].
Repeat the same exercise again if the claimed average is 40, and use the Boston dataset sample to test this hypothesis.

Conclusion

- We discussed 'Linear Regression' following ML landscape.
- We identified the 'learning' part regarding this algorithm. For example, clear concept of loss/cost function, and what we were trying to optimise/learn/evaluate?
- We also introduced the concept of regularization from Linear Regression perspective. Its motivation is generally the same when used in other algorithms.
- We interpreted Linear Regression's application in a few ways – decision making, hypothesis testing or just interpreting the developed model and its parameters' significance.
- Perform the lab tasks related to the two Linear Regression notebooks which will help understanding the concepts better.

References

David Lane.
Introduction to Statistics.
Open Textbook Library - free eBook link here, 2003.

Sebastian Raschka and Vahid Mirjalili.
Python Machine Learning, 3rd Ed.
Packt Publishing, Birmingham, UK, 2019.

Stats Tutor (this site may require deprecated Flash Plugin).
Statistics support for students.
`https://www.statstutor.ac.uk/`.
Accessed on 05.02.2021.