Applied Analytics in Finance

Matilda Bosire

Strathmore University

September 21, 2023

Outline

Table of Contents

Supervised Learning Techniques

ML Algorithm Classes	Algorithm Names
Regression	Linear, Polynomial, Logistic, Stepwise, OLSR (Ordinary Least Squares Regression), LOESS (Locally Estimated Scatterplot Smoothing), MARS (Multivariate Adaptive Regression Splines)
Classification	KNN (k-nearest Neighbor), Trees, Naïve Bayesian, SVM (Support Vector Machine), LVQ (Learning Vector Quantization), SOM (Self-Organizing Map), LWL (Locally Weighted Learning)
Decision Trees	Decision trees, Random Forests, CART (Classification and Regression Tree), ID3 (Iterative Dichotomiser 3), CHAID (Chi-squared Automatic Interaction Detection), ID3 (Iterative Dichotomiser 3), CHAID (Chi-squared Automatic Interaction Detection)
Bayesian Networks	Naïve Bayesian, Gaussian, Multinomial, AODE (Averaged One-Dependence Estimators), BBN (Bayesian Belief Network), BN (Bayesian Network)

Figure: Supervised Machine Learning Algorithms

Unsupervised Learning Techniques

ML Algorithm Classes	Algorithm Names
Association Analysis	A priori, Association Rules, Eclat, FP-Growth
Clustering	Clustering analysis, k-means, Hierarchical Clustering, Expectation Maximization (EM), Density-based Clustering
Dimensionality Reduction	PCA (principal Component Analysis), Discriminant Analysis, MDS (Multi-Dimensional Scaling)
Artificial Neural Networks (ANNs)	Perception, Back propagation, RBFN (Radial Basis Function Network)

Figure: Unsupervised Machine Learning Algorithms

Table of Contents

Why not Linear Regression?

- The linear regression model assumes that the response variable Y is quantitative. However, in many situations, the response variable is qualitative.
- In the credit exercise, the probability of default is qualitative, as it categorizes debtors into default or non-default events (often, qualitative variables are referred to as categorical).
- Note that in a binary case it is not hard to show that even if we flip the coding (if $\hat{Y} > 0.5$, we have a default event and non-default otherwise), linear regression will produce the same final predictions.

Classification

- Predicting a qualitative response for an observation can be referred to as classifying that observation, since it involves assigning the observation to a category or class.
- Widely-used classifiers include logistic regression, linear discriminant analysis, quadratic discriminant analysis, naive Bayes, and K-nearest neighbors.

Table of Contents

Logistic Regression

- Models the probability that the response variable belongs to a particular category.
- For the Credit data, logistic regression models the probability of default, i.e.:

$$Pr(Default = Yes|age, savings.balance, checking.balance, past.debt, etc.$$

- The probabilities range from 0 to 1, and one might predict a default event for any individual for whom $p(age, savings.balance, checking.balance, past.debt, etc.) > 0.5$.

Logistic Regression

- To avoid negative or very large probabilities for predictor values that are close to zero or very large respectively, we must model $p(X)$ using a function that gives outputs between 0 and 1 for all values of X. In logistic regression, we use the logistic function:

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}},$$

  which translates to:

$$\frac{p(X)}{1 - p(X)} = e^{\beta_0+\beta_1 X}.$$

- The outcome of this model is always bounded between 0 and 1 and can therefore be interpreted as a probability.
- The quantity $\frac{p(X)}{1-p(X)}$ is called the *odds ratio*, and takes on values between 0 and infinity. Values close to 0 indicate very low probabilities, and vice versa.

Logistic Regression

- By taking the logarithm on both ends:

$$log(\frac{p(X)}{1-p(X)}) = \beta_0 + \beta_1 X,$$

the LHS is called the *log odds* or *logit*, that is linear in X, and can be generalized to p predictors as:

$$log(\frac{p(X)}{1-p(X)}) = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p.$$

- The linear regression, logistic regression, and Poisson regression are three examples of Generalized Linear models (`https://cran.r-project.org/web/packages/glm2/glm2.pdf`).
- MLE is preferred in parameter estimation, with p-values (¡ 0.05) being utilized to test for significance of predictor variables (reject the null hypothesis, $H_0 : \beta = 0$).
- A unit increase in a predictor is associated with an increase in the log odds of default by it's β coefficient.

Logistic Regression

- **Lab Exercise on course textbook: Chapter 4, subsection 7 (pg. 171)**

Logistic Regression

Example

Suppose a lender wants to predict the probability of default for a new loan applicant based on their credit history and demographic information. The lender has historical data on 1,000 borrowers, including their credit score (continuous variable ranging from 300 to 850), age (continuous variable in years), and employment status (categorical variable: employed, self-employed, or unemployed), as well as whether or not they defaulted on their loan (binary variable: 1 for default, 0 for non-default). We want to build a logistic regression model to predict the probability of default based on the input variables.

Logistic Regression - Solution

- The model has the representation

$$logit(p) = b_0 + b_1 * credit\_score + b_2 * age + b_3 * employment\_status$$

where p is the probability of default, $logit(p)$ is the log of the odds of default, $b_0 = -2.52$, $b_1 = 0.008$, $b_2 = 0.05$, and $b_3 = 0.77$ (if employment status is self-employed) and $b_3 = 1.09$ (if unemployed) are the coefficients estimated via ML. We note:

1. b_0 is the intercept, represents the log odds of default when all input variables are equal to zero (i.e., a borrower with a 0 credit score, age of 0, and employment status of employed). We have a negative intercept indicating that the baseline probability of default is low
2. b_1 is the coefficient for credit score, which represents the change in log odds of default for a one-unit increase in credit score. The estimated coefficient is positive, indicating that higher credit scores are associated with lower default probabilities

Logistic Regression - Solution

3. b_2 is the coefficient for age, which represents the change in log odds of default for a one-year increase in age. The coefficient is positive, indicating that older borrowers are associated with lower default probabilities
4. b_3 is the coefficient for employment status, which represents the difference in log odds of default for borrowers who are self-employed or unemployed compared to those who are employed. The coefficient is positive and larger for unemployed borrowers, indicating that they are associated with higher default probabilities

- To predict the probability of default for a new loan applicant, we use the logistic regression model to compute the log odds of default, and then transform this into a probability using the logistic function:

$$p = 1/(1 + exp(-logit(p)))$$

Logistic Regression - Solution

- Suppose a new loan applicant has a credit score of 700, age of 35, and is unemployed, then

$$logit(p) = -2.52 + 0.008700 + 0.0535 + 1.09 = 2.98$$

and

$$p = 1/(1 + exp(-2.98)) = 0.95. \quad \text{Interpretation?}$$

* In practice, the logistic regression model is validated using various statistical measures such as the **accuracy rate** and the **area under the receiver operating characteristic curve** to ensure that it performs well on new data
* $\beta_i > 0$ implies $e^{\beta_i} > 1$ and the odds and probability increase whereas $\beta_i < 0$ implies $e^{\beta_i} < 1$ and the odds and probability decrease with x_i

Table of Contents

K Means Clustering

- Non-parametric, un-supervised learning; measures the distance between different data points.
- Given a value for K, and a prediction point x_0, KNN first identifies the training observations closest to x_0, represented by N_0.
- It then estimates $f(x_0)$ (response), using the average of all training responses in N_0, i.e., the prediction in a region is the average of several points:

$$\hat{f}(x_0) = \frac{1}{K} \sum_{x_i \in N_0} y_i$$

K Means Clustering

- The idea is to first specify the number of clusters, K, then the algorithm will will assign each observation to exactly one of the K-clusters.
- Clusters are non-overlapping: no observation belongs to more than one cluster.
- A good clustering technique is one where the **within-cluster variation** is as small as is possible.
- Generally, an optimal K-Means algo is such that the total within-cluster variation of all the independent clusters is as small as is possible.

K Means Clustering

- Denote the within-cluster variation for cluster, C_k as a measure, $W(C_k)$. Then, the optimization problem to be solved in K-Means is:

$$\underset{C_1,\ldots,C_k}{minimize} \sum_{k=1}^{K} W(C_k)$$

- To solve this in an actionable way, we use the **squared Euclidean distance**. That is:

$$W(C_k) = \frac{1}{|C_k} \sum_{i,i' \in C_k} \sum_{j=1}^{P} (x^{i,j} - xi', j)^2$$

where xi', j is the centroid, and $|C_k|$ represents the observations in the K^{th} cluster.

- The within-cluster variation is the sum of all pairwise squared **Euclidean Distances** between observations in a cluster, divided by the total number of observations in that cluster.

K Means Clustering

- The optimal value of K will depend on the **bias-variance trade-off**.
- A small value of K provides a flexible fit which will have low bias but high variance. The high variance is due to the fact that the prediction in a given region is entirely dependent on one observation.
- Larger values of K provide smoother and less variable fits, i.e., changing one observation has a smaller effect. However, the smoothing may cause some bias by masking some of the structure of $f(X)$, i.e, it runs the risk of ignoring small, but important patterns.

K Means Clustering

- The algorithm converges to a local rather than a global optimum, and results will depend on the initial (random) cluster assignment for each observation. It is important to run the algo multiple times from random different configurations, then select the solution as one with the lowest within-cluster variation.

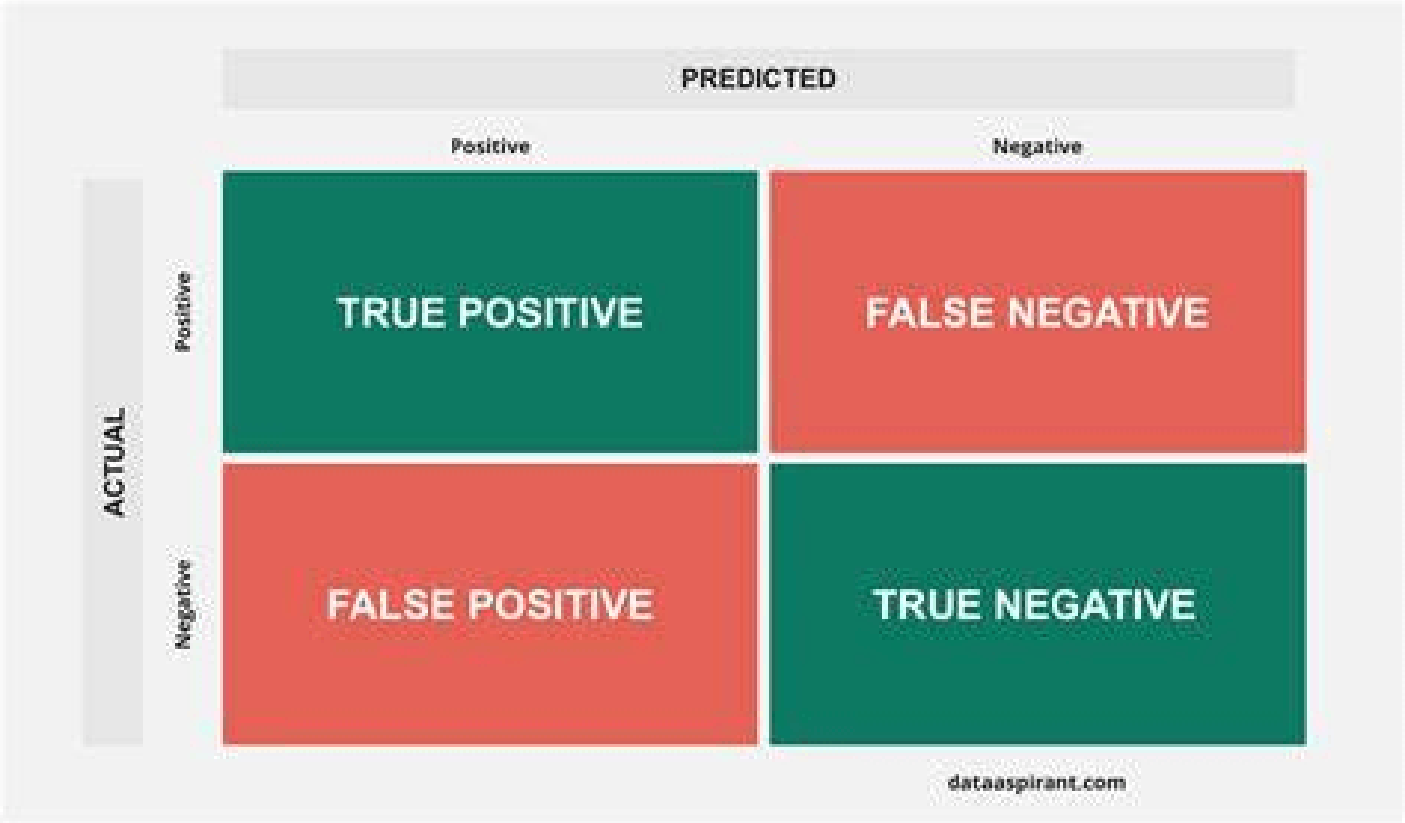
K-Means vs KNN

- KNN is a supervised machine learning classification algorithm, while on the other hand, K-Means is an unsupervised machine learning clustering algorithm.
- Nearest neighbor classifiers are defined by their characteristic of classifying unlabeled examples by assigning them the class of the most similar labeled examples.
- KNN analyzes the 'K' nearest (labeled) data points and then classifies the new data based on the same. It selects the label of the new point as the one to which the majority of the 'K' nearest neighbors belong to.

Table of Contents

Evaluation Metrics for Classification

Confusion Matrix



- N*N matrix where N represented the number of classes in the variable,in this case we have 2 classes, defaulters (1) and non-defaulters (0). As such we will have a 2*2 confusion matrix (binary matrix).

Evaluation Metrics for Classification

- Each row represents actual values while each column represents predicted values from the credit risk model.
- Assume positive values represent non-default (good credit) and negative represents default (bad credit).
- From the knn() analysis in R, we have:

		Predicted	
		0	1
Actual	0	25	38
	1	60	115

- Row entries should sum up to the total actual positive (true positive + false negative) and total actual negative (false positive + true negative).

Evaluation Metrics for Classification

- The model's accuracy can be calculated as:

$$Accuracy = \frac{\text{Correct Predicted Values}}{\text{Total Predicted Values}}$$

$$= \frac{\text{True Positive} + \text{True Negative}}{\text{Total Predicted Values}}$$

Evaluation Metrics for Classification

- Alternatives to accuracy include:
  i. True Positive Rate =

$$\frac{\text{True Positive Predictions}}{\text{Total Actual Positive}}$$

  ii. False Negative Rate =

$$\frac{\text{Actual Positive, Predicted as Negative}}{\text{Total Actual Positive}}$$

  iii. False Positive Rate =

$$\frac{\text{Actual Negative, Predicted as Positive Predictions}}{\text{Total Actual Negative}}$$

  iv. True Negative Rate =

$$\frac{\text{True Negative Predictions}}{\text{Total Actual Negative}}$$

- Row entry rates sum up to 1: TPR + FNR = 1, and FPR + TNR =1.
- For [i.] and [iv.], a higher value implies a better model.

Evaluation Metrics for Classification

Precision

- Of all the positive predictions, how many are actually positive?

$$= \frac{\text{True Positive Predictions}}{\text{Total Predicted Positives} = \text{True Positive} + \text{False Positive}}$$

- Employed where there is a need to minimize false positives (putting false negatives aside) in the real world.
- E.g., model prediction for an individual as a non-defaulter when actually a defaulter...

Evaluation Metrics for Classification

Recall

- Out of all actual positive predictions, how many have been predicted as positive?

$$= \frac{\text{True Positive Predictions}}{\text{Total Actual Positive} = \text{True Positive} + \text{False Negative}}$$

- Employed where there is a need to minimize false negatives as a priority to false positives.
- E.g., model prediction for an individual as a defaulter when actually a non-defaulter...
- Check precision-recall trade-off. They have an inverse relationship.
- Ideally, the choice depends on the use case.

Evaluation Metrics for Classification

F_1 Score

- Precision and recall are combined into the F_1 score, i.e.,

$$F_1 = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}}$$

- It reaches a maximum value of 2 where Precision = Recall. Hence if $F_1 \to 2$, the better the model.

Evaluation Metrics for Classification

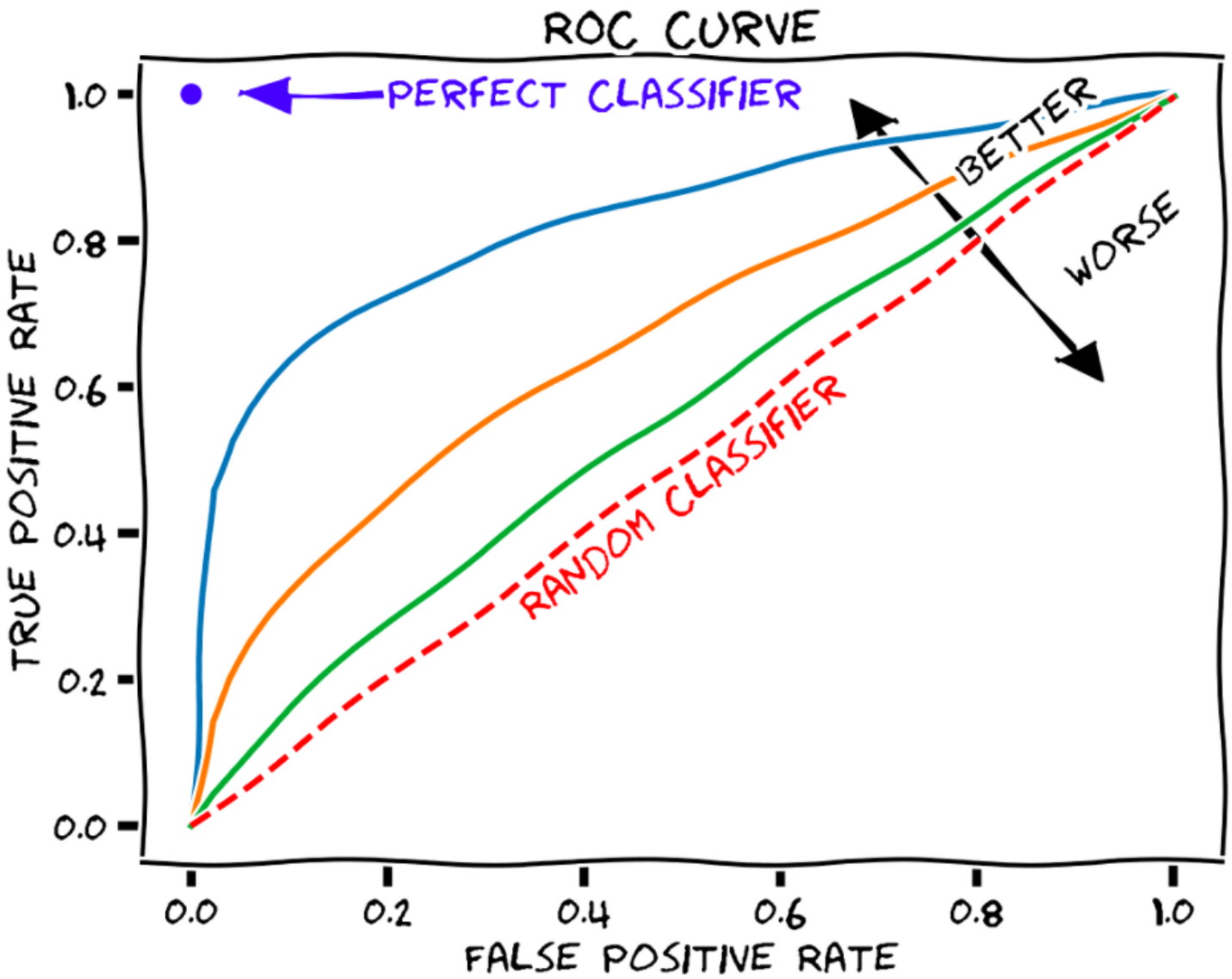
AUC- Area under curve
ROC- Receiver Operating curve
The more the AUC, the better the model.

AUC - ROC

- The ROC is a signal detection metric used to distinguish noise in binary classification.
- It gives the trade-offs between the true positives and the false positives.
- The more the area under the curve, the better the model.
- To plot the curve, we need multiple TPR and FPR values under different threshold.

Classifiers could include; K-nn, K-means, logistic regression.

Evaluation Metrics for Classification



Evaluation Metrics for Classification

- The R package **ROCR** is employed in visualizing the performance of scoring classifiers as discussed above.
- `https://cran.r-project.org/web/packages/ROCR/ROCR.pdf`

Evaluation Metrics for Classification

Log Loss

- One problem with the AUC-ROC is that it cannot be used to compare two or more models.
- The LogLoss is able to achieve this.
- The LogLoss is the negative average of the log of corrected predicted probabilities for each instance.

Evaluation Metrics for Classification

ID	Actual Class Value	Predicted Probability	Corrected Probability (p_i)
ID1	1	0.95	0.95
ID2	1	0.84	0.84
ID3	0	0.78	(1-0.78)
ID4	1	0.66	0.66
ID5	0	0.52	(1-0.52)

Evaluation Metrics for Classification

- Calculate the log of the corrected probabilities : $log(p_i)$, for which we get negative values, thus get the negative average of this:

$$-\frac{1}{N}\sum_{i=1}^{N} log(p_i)$$

- Generally, the LogLoss is calculated as:

$$\text{LogLoss} = -\frac{1}{N}\sum_{i=1}^{N}(y_i * log(p_i) + (1 - y_i) * (log(1 - p_i))),$$

where y_i represents the actual class (0 or 1), p_i is the probability of being in class 1, while (1-p_i) is the probability of being in class 0.

Evaluation Metrics for Regression

- Error metrics measure how far the predicted values are from the observed/actual values.

$$\text{Error} = \text{Actual Value - Predicted Value}$$

- We might have some positive and negative values, which on summing might reduce the aggregate error value to 0.

Evaluation Metrics for Regression

i. Mean Absolute Error (MAE) $= \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$

ii. Mean Squared Error (MSE) $= \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$

- MSE changes the units, e.g., if the data is measured in metres, getting squared error will change the measurement to metres squared.
- To go back to the original unit, we use the Root Mean Squared Error (RMSE).

iii. $RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$

- The RMSE does not still depict the true performance of the model, which is accounted for by the Root Mean Square Log Error (RMSLE), calculated as:

iv. $RMSLE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(log(y_i + 1) - log(\hat{y}_i + 1))^2}$

Evaluation Metrics for Regression

- For all error measures, model performance increases with reduced MAE, MSE, RMSE, RMSLE.
- Error values are however not intuitive. They do not have a benchmark to compare against.
- Considering the MSE(model), we can compare it to a baseline model.

$$\text{Relative Squared Errors} = \frac{MSE(model)}{MSE(baseline)} = \frac{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\frac{1}{N}\sum_{i=1}^{N}(\overline{y_i} - \hat{y}_i)^2}$$

- For the baseline model, we find the MSE where actual values are replaced by the mean of all actual values.
- Where MSE(model) = MSE(baseline), the relative squared error = 1. This is ideal. If the relative squared error > 1, the model performance is worse than the baseline model.

Evaluation Metrics for Regression

R-Squared

- Conventionally, we would want the MSE(model) to be low. The lower the MSE(model), the lower the relative squared error, and we can say that the model performance has improved.
- The R-squared metric accounts for this dilemma.

$$R^2 = 1 - \frac{MSE(model)}{MSE(baseline)}$$

We want to have a very small MSE(model) hence R^2 should be as close as possible to zero to say our model is good.

- A higher value for R^2 is preferred as its equal to reducing the MSE(model).
- A disadvantage of the R^2 metric is that it increases or stays constant with more model features. It does not decrease regardless of how the feature will impact the model.

Evaluation Metrics for Regression

Adjusted R-Squared

- Imposes a penalty for insignificant model features.

$$\overline{R}^2 = 1 - (1 - R^2)[\frac{n-1}{n-(k+1)}],$$

  where n is the sample size and k the number of features.
- $\overline{R}^2$ reduces with an increased number of features found to be insignificant.