Project: Batch DS2402

Assignment: Worksheet Set 2

Submitted by: Geetanjali Joshi

Datatrained batch : DSG0823

MACHINE LEARNING

ASSIGNMENT - 5

Q1 to Q15 are subjective answer-type questions, Answer them briefly.

1. **R-squared or Residual Sum of Squares (RSS) which one of these two is a better measure of the goodness of fit model in regression and why?**

Solution: Both R-squared and Residual Sum of Squares (RSS) are model evaluation tools that are used to measure a better fit for a model in regression analysis.

R-squared: R-squared measures the proportion of variation in the dependent variable that is explained by the independent variables in the model. In other words, it indicates how well the model fits the data, with values ranging from 0 to 1.

R square = 1 - RSS/ TSS. (Where RSS = Residual sum of squares,
TSS = Total Residual sum of squares)

Residual Sum of Squares (RSS) : RSS measures the total sum of squared differences between the actual values of the dependent variable and the predicted values by the model. It represents the amount of unexplained variation in the data, and lower RSS values indicate a better fit, as they mean that the model can explain more of the variation in the data.

Which one is better?

R- squared is usually preferred as a measure of the goodness of fit of a model because it provides a stan. Higher R-squared values indicate a better fit, as they mean that a larger proportion of the variation in the dependent variable is explained by the independent variables in the model.

2. What are TSS (Total Sum of Squares), ESS (Explained Sum of Squares) and RSS (Residual Sum of Squares) in regression. Also mention the equation relating these three metrics with each other.

Solution: All the above metrics are used to measure the goodness of a fit of a model in regression analysis. These metrics are defined as follows:

- **Total Sum of Squares (TSS):** it measures how dependent variable varies regardless of the independent variables. It is the sum of all squared differences between mean of a sample and the individual values in the sample.

$$\mathrm{TSS} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

- **Explained sum of squares (ESS):** is the sum of the squares of the deviations of the predicted values from the mean value of a response variable, in a standard regression model.

$$\mathrm{ESS} = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2.$$

- **Residual sum of squares (RSS) :** Residual Sum of Squares (RSS) is a statistical method that helps identify the level of discrepancy in a dataset not predicted by a regression model. Thus, it measures the variance in the value of the observed data when compared to its predicted value as per the regression model. Hence, RSS indicates whether the regression model fits the actual dataset well or not.

$$\mathrm{RSS} = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

Relationship between the three metrics :

TSS = ESS + RSS

3. What is the need of regularization in machine learning?

Solution: When we are training a model, it can result in overfitting or underfitting. To avoid this from occuring we use regularization. The regularization technique helps address this issue by adding a penalty to the model's objective function.

Need for regularization:
- To prevent overfitting by penalizing large coefficients or model complexity, regularization encourages models to generalize better to unseen data.
- To improve generalisation - Regularization helps to capture the underlying patterns in the data while avoiding fitting noise and irrelevant details present in the training data.
- Helps in handling collinearity
- Leads to better feature selection.

4. What is Gini–impurity index?

Solution: Gini impurity is a measure used in decision tree algorithms to quantify a dataset's impurity level or disorder.

Gini impurity = 1 - Gini

Gini ranges from zero to one, as it is a probability and the higher this value, the more will be the purity of the nodes. And of course, a lesser value means fewer pure nodes.
Lower the Gini impurity we can safely infer the purity will be more and hence a higher chance of the homogeneity of the nodes.

5. Are unregularized decision trees prone to overfitting? If yes, why?

Solution: Yes, unregularized decision trees are prone to overfitting. Here are the following reasons behind it:

- Decision trees are sensitive to noise and outliers. As a result, the tree may overfit by memorizing the noise rather than learning the underlying structure of the data.
- They can grow deep and result in excessive specific rules that cannot be generalised effectively.
- Decision trees are very complex and try to fit the training data, especially if there are many features and less constraints. This can lead to poor generalization of data.

6. What is an ensemble technique in machine learning?

Solution: An ensemble technique is that **combines multiple models to improve the accuracy of predictions**. It works by training multiple models on the same dataset and combining their outputs to produce a final prediction. By this it combines prediction of multiple models, so that its weaknesses can be mitigated and strengths can be amplified.

Ensemble methods are used widely in machine learning tasks, such as classification, regression, and anomaly detection etc.

7. What is the difference between Bagging and Boosting techniques?

Solution: Bagging: Bagging is a method of merging the same type of predictions. Boosting is a method of merging different types of predictions. In Bagging, each model receives an equal weight.

Boosting: In Boosting, models are weighed based on their performance.
Models are built independently in Bagging. New models are affected by a previously built model's performance in Boosting.

Difference Between Bagging and Boosting or Bagging vs Boosting

Feature	Bagging	Boosting
Objective	Reduce variance and prevent overfitting	Reduce bias and improve accuracy
Base Learners	Independent models trained in parallel	Sequentially trained weak learners
Weighting	Equal weight for all base learners	Weighted based on performance
Error Correction	Independent errors; no re-weighting	Emphasis on correcting mistakes
Training Speed	Parallel training; faster	Sequential training; slower
Final Model	Average or voting of base models	Weighted sum of base learners

Robustness	Less prone to overfitting	Prone to overfitting if not controlled
Examples	Random Forest	AdaBoost, Gradient Boosting

In Bagging, training data subsets are drawn randomly with a replacement for the training dataset. In Boosting, every new subset comprises the elements that were misclassified by previous models. Bagging is usually applied where the classifier is unstable and has a high variance. Boosting is usually applied where the classifier is stable and simple and has high bias.

8. What is out-of-bag error in random forests?

Solution: Out-of-Bag Error, also known as OOB Error, is a concept used in ensemble machine learning algorithms such as random forest When building a random forest model, each tree is trained using a subset of the original data, known as the bootstrap sample. During the training process, some observations are left out or "out-of-bag" (OOB) for each tree.

Out-of-bag error works by utilizing the OOB observations to estimate the model's performance on unseen data. For each observation, the OOB error is computed by comparing the model's prediction with the ground truth value. This process is repeated for all the OOB observations and averaged across all the trees in the ensemble.
The OOB Error provides an unbiased estimate of the model's performance without the need for a separate validation set. It serves as an internal validation mechanism within the random forest algorithm.

Out-of-bag error estimation works in random forests:

Bootstrapping: When constructing each decision tree in the random forest, a random subset of the original dataset is sampled with replacement. This means that some data points may be selected multiple times, while others may not be selected at all.

Out-of-bag samples: Since the sampling is done with replacement, around 1/3 of the original dataset on average is not included in the bootstrap sample for each tree. These data points that are not selected form the out-of-bag samples for that particular tree.

Evaluation: For each tree in the random forest, the out-of-bag samples are used as a validation set. This means that each data point in the out-of-bag samples has not been used to train the tree it is being evaluated on.

Error estimation: The out-of-bag error is calculated by aggregating the predictions made by each tree on its corresponding out-of-bag samples and comparing them to the true labels. This error estimate provides an unbiased assessment of the model's performance because each data point is effectively used for testing only once across the ensemble of trees.

9. What is K-fold cross-validation?

Solution: K-Fold Cross Validation, . K-fold cross-validation is a technique used to assess the performance of a machine-learning model and to ensure that it generalizes well to unseen data. In this we split the dataset into k number of subsets (known as folds) then we perform training on all the subsets but leave one(k-1) subset for the evaluation of the trained model. In this method, we iterate k times with a different subset reserved for testing purposes each time. K-fold cross-validation is a technique used to assess the performance of a machine-learning model and to ensure that it generalizes well to unseen data.

How does it work?
Dataset splitting: The dataset is divided into k subsets of approximately equal size. Each subset is referred to as a fold.
Training and validation: In each iteration of the cross-validation process, one of the k folds is held out as the validation set, and the model is trained on the remaining k-1 folds.
Evaluation: After training on the training set, the model is evaluated on the validation set. The performance metrics (such as accuracy, precision, recall, etc.) are computed based on the predictions made on the validation set.
Iteration: The process is repeated k times, with each fold being used exactly once as the validation set. This ensures that every data point is used for both training and validation exactly once across the k iterations.
Performance estimation: Once all iterations are complete, the performance metrics obtained from each iteration are typically averaged to obtain a single performance estimate for the model.

10. What is hyperparameter tuning in machine learning and why it is done?

Solution: Hyperparameter tuning is the process of selecting the optimal values for a machine learning model's hyperparameters. Hyperparameters are settings that control the learning process of the model, such as the learning rate, the number of neurons in a neural network, or the kernel size in a support vector machine. The goal of hyperparameter tuning is to find the values that lead to the best performance on a given task.

Why is it done?

Hyperparameters directly control model structure, function, and performance. Hyperparameter tuning allows data scientists to tweak model performance for optimal results. This process is an essential part of machine learning, and choosing appropriate hyperparameter values is crucial for success.

11. What issues can occur if we have a large learning rate in Gradient Descent?

Solution: The learning rate in gradient descent controls the size of the step and determines how quickly the algorithm moves towards the minimum. A large learning rate can cause issues such as :

- divergence - When the learning rate is too large, gradient descent can suffer from divergence. This means that weights increase exponentially, resulting in exploding gradients. This can cause problems such as instabilities and overly high loss values.
- Large learning rates can lead to oscillations or erratic behavior during training.
- Training with a large learning rate may result in a model that performs well on the training data but generalizes poorly to unseen data.
- If the learning rate is too large, one step of gradient descent can actually vastly "overshoot" and actually increase the value of $f(\theta 0, \theta 1)$

12. Can we use Logistic Regression for classification of Non-Linear Data? If not, why?

Solution: Logistic Regression has traditionally been used as a linear classifier, i.e. when the classes can be separated in the feature space by linear boundaries. But it can also be used as a for classification of non-linear data, but it may not perform well in cases where the decision boundary between classes is highly non-linear.

logistic regression can still be used for non-linear classification tasks by employing techniques like feature engineering, polynomial feature expansion, or transformation of features to capture non-linear relationships between predictors and the target variable. These techniques can help create non-linear decision boundaries by transforming the input features into higher dimensions where linear separation might be possible.
Despite these possibilities, logistic regression may not always perform as well as other non-linear classifiers, such as decision trees, support vector machines (SVMs), or neural networks, especially when the data is highly non-linear and complex. In such cases, more sophisticated models capable of capturing intricate non-linear relationships may be more suitable.

13. Differentiate between Adaboost and Gradient Boosting.

Solution: Adaboost: A popular ensemble learning algorithm used primarily for classification tasks. The idea behind AdaBoost is to combine multiple weak learners (classifiers that perform slightly better than random guessing) to create a strong classifier.

Gradient boosting: Another popular ensemble learning technique that builds a strong predictive model by sequentially combining multiple weak learners. Unlike AdaBoost, which adjusts the weights of data points at each iteration, gradient boosting fits each new model to the residual errors made by the previous model. This approach allows gradient boosting to optimize arbitrary differentiable loss functions.

Features	Gradient boosting	Adaboost
Model	It identifies complex observations by huge residuals calculated in prior iterations	The shift is made by up-weighting the observations that are miscalculated prior
Trees	The trees with week learners are constructed using a greedy algorithm based on split points and purity scores. The trees are grown deeper with eight to thirty-two terminal nodes. The week learners should stay a week in terms of nodes, layers, leaf nodes, and splits	The trees are called decision stumps.
Classifier	The classifiers are weighted precisely and their prediction capacity is constrained to learning rate and increasing accuracy	Every classifier has different weight assumptions to its final prediction that depend on the performance.

Prediction	It develops a tree with help of previous classifier residuals by capturing variances in data. The final prediction depends on the maximum vote of the week learners and is weighted by its accuracy.	It gives values to classifiers by observing determined variance with data. Here all the week learners possess equal weight and it is usually fixed as the rate for learning which is too minimum in magnitude.
Short-comings	Here, the gradients themselves identify the shortcomings.	Maximum weighted data points are used to identify the shortcomings.
Loss value	Gradient boosting cut down the error components to provide clear explanations and its concepts are easier to adapt and understand	The exponential loss provides maximum weights for the samples which are fitted in worse conditions.
Applications	This method trains the learners and depends on reducing the loss functions of that week learner by training the residues of the model	Its focus on training the prior miscalculated observations and it alters the distribution of the dataset to enhance the weight on sample values which are hard for classification

14. What is a bias-variance trade off in machine learning?

Solution: The bias-variance tradeoff is a fundamental concept in machine learning that describes the balance between model bias and variance. It's a central problem in supervised learning that impacts a model's predictive performance

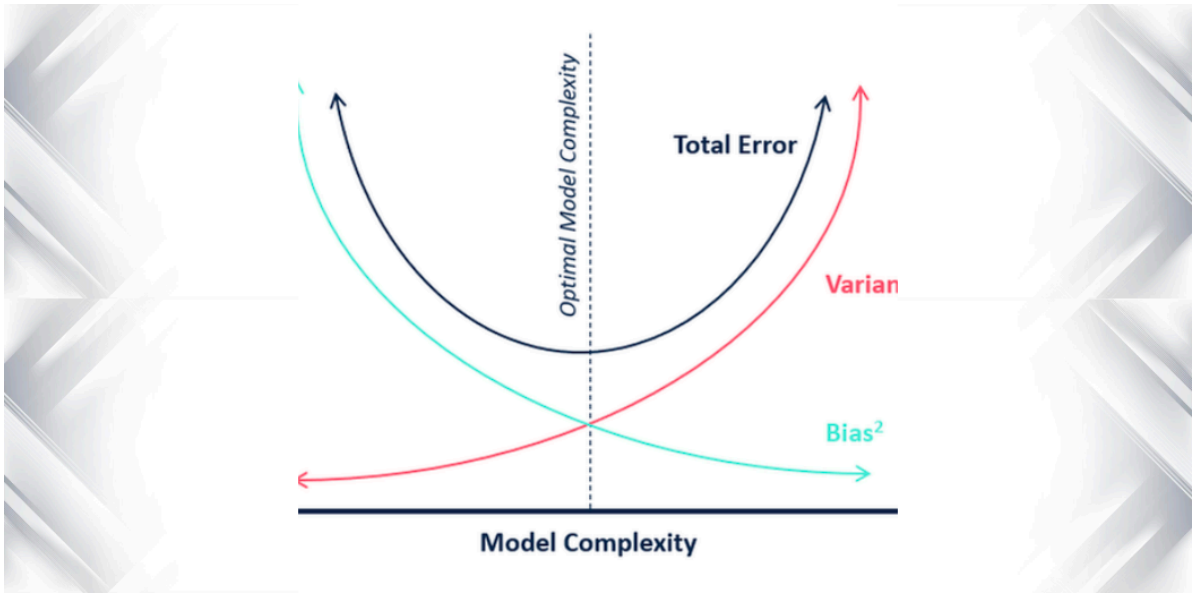
Bias in machine learning refers to the difference between a model’s predictions and the actual distribution of the value it tries to predict. Models with high bias oversimplify the data distribution rule/function, resulting in high errors in both the training outcomes and test data analysis results.

Variance stands in contrast to bias; it measures how much a distribution on several sets of data values differs from each other. The most common approach to measuring variance is by performing cross-validation experiments and looking at how the model performs on different random splits of your training data.

The bias–variance tradeoff describes the relationship between a model's complexity, the accuracy of its predictions, and how well it can make predictions on previously unseen data that were not used to train the model.

15. Give short description each of Linear, RBF, Polynomial kernels used in SVM.

Solution:



1. Linear kernels : It is simple kernel function that works with linearly seperable data. it defines the dot product between the input vectors in the original feature space. These are less time consuming but provides less accuracy.
2. Radial basis function: The RBF kernel is commonly used in support vector machine classification. It's also known as the kernel function. It uses gaussian function to measure similarity between data points.
3. Polynomial kernels: used to capture non-linear relationships between features. The polynomial kernel takes the dot product of the input data points and adds a constant to the result, which is raised to a power specified by the degree parameter of the function. It considers the similarity between vectors under the same dimension, but also across dimensions. When used in machine learning algorithms, this allows to account for feature interaction.