Machine Learning Worksheet 5

1. **R-squared** is a goodness-of-fit measure :

R-squared evaluates the scatter of the data points around the fitted regression line. It is also called the coefficient of determination, or the coefficient of multiple determination for multiple regression. For the same data set, higher R-squared values represent smaller differences between the observed data and the fitted values.

1. The sum of squares total, denoted SST, is the squared differences between the observed *dependent variable* and its mean. You can think of this as the dispersion of the observed variables around the mean – much like the variance in descriptive statistics. It is a measure of the total variability of the dataset. There is another notation for the SST. It is TSS or total sum of squares.

**SSR:** The second term is the **sum of squares due to regression**, or **SSR**. It is the sum of the differences between the predicted value and the **mean** of the dependent variable. Think of it as a measure that describes how well our line fits the data. If this value of **SSR** is equal to the **sum of squares total**, it means our **regression** **model** captures all the observed variability and is perfect. Once again, we must mention that another common notation is **ESS** or **explained sum of squares**.

SSE: The last term is the **sum of squares error**, or **SSE**. The error is the difference between the *observed* value and the *predicted* value. We usually want to minimize the error. The smaller the error, the better the estimation power of the **regression**. Finally, I should add that it is also known as **RSS** or **residual sum of squares**. Residual as in remaining or unexplained.

How Are They Related?

Mathematically, **SST** = **SSR** + **SSE**.

The rationale is the following: the total variability of the data set is equal to the variability explained by the **regression line** plus the unexplained variability, known as error. Given a constant total variability, a lower error will cause a better **regression**. Conversely, a higher error will cause a less powerful **regression**. And that’s what you must remember, no matter the notation.

1. Regularization refers to techniques that are used to calibrate machine learning models to **minimize the adjusted loss function and prevent overfitting or underfitting**
2. Gini Impurity is **a measurement used to build Decision Trees to determine how the features of a dataset should split nodes to form the tree**.
3. Decision trees are prone to overfitting, especially when a tree is particularly deep. This is due to the amount of specificity we look at leading to smaller sample of events that meet the previous assumptions. This small sample could lead to unsound conclusions.
4. Ensemble methods are techniques that create multiple models and then combine them to produce improved results. Ensemble methods usually produces more accurate solutions than a single model would. This has been the case in several machine learning competitions, where the winning solutions used ensemble methods.
5. Bagging is a technique for reducing prediction variance by producing additional data for training from a dataset by combining repetitions with combinations to create multi-sets of the original data. Boosting is an iterative strategy for adjusting an observation's weight based on the previous classification.
6. The out-of-bag (OOB) error is **the average error for each calculated using predictions from the trees that do not contain in their respective bootstrap sample**. This allows the RandomForestClassifier to be fit and validated whilst being trained.
7. K-fold Cross-Validation is when the dataset is split into a K number of folds and is used to evaluate the model's ability when given new data. K refers to the number of groups the data sample is split into. For example, if you see that the k-value is 5, we can call this a 5-fold cross-validation. Each fold is used as a testing set at one point in the process.
8. Hyperparameter tuning consists of **finding a set of optimal hyperparameter values for a learning algorithm while applying this optimized algorithm to any data set**. That combination of hyperparameters maximizes the model's performance, minimizing a predefined loss function to produce better results with fewer errors
9. A learning rate that is too large can **cause the model to converge too quickly to a suboptimal solution**, whereas a learning rate that is too small can cause the process to get stuck. The challenge of training deep learning neural networks involves carefully selecting the learning rate
10. 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. That can be remedied however if we happen to have a better idea as to the shape of the decision boundary
11. AdaBoost is the first designed boosting algorithm with a particular loss function. On the other hand, Gradient Boosting is a generic algorithm that assists in searching the approximate solutions to the additive modelling problem. This makes Gradient Boosting more flexible than AdaBoost
12. If the algorithm is too simple (hypothesis with linear eq.) then it may be on high bias and low variance condition and thus is error prone. If algorithms fit too complex (hypothesis with high degree eq.) then it may be on high variance and low bias. In the latter condition, the new entries will not perform well. Well, there is something between both conditions, known as Trade-off or Bias Variance Trade-off. This trade-off in complexity is why there is a trade-off between bias and variance. An algorithm can’t be more complex and less complex at the same time.
13. *Linear Kernel* is used **when the data is Linearly separable**, that is, it can be separated using a single Line. It is one of the most common kernels to be used. It is mostly used when there are many Features in a particular Data Set

In machine learning, the *polynomial kernel* is a kernel function commonly used with support vector machines (SVMs) and other kernelized models, that represents the similarity of vectors (training samples) in a feature space over polynomials of the original variables, allowing learning of non-linear models.

*RBF Kernel* is popular because of its similarity to K-Nearest Neighbourhood Algorithm. It has the advantages of K-NN and **overcomes the space complexity problem** as RBF Kernel Support Vector Machines just needs to store the support vectors during training and not the entire dataset