Question-1:

Rahul built a logistic regression model having a training accuracy of 97% while the test accuracy was 48%. What could be the reason for the seeming gulf between test and train accuracy and how can this problem be solved.

Answer-1

We observe a substantial difference between the test and train accuracy. This can be attributed to the fact that the model excessively applied and learned the rules for the train set but suffers with bad generalization properties. Such a phenomenon is referred to as Overfitting.

Overfitting is associated with complexity of model. Such models work very well with the train data as they learn the peculiarities of the train set during the training process, and give high accuracy with train set. However, even minor fluctuations in data sets generate errors and low accuracy, and these models do not work well with parameters that are different from the training set. Hence, they lack the generalization capability.

To solve the Overfitting problem, we can adopt a process called Regularization. **Regularization** is a process used to create an optimally complex model, i.e. a model which is as simple as possible while performing well on the training data. Through this, our aim is to adjust the model so that it becomes simple, and yet we should keep the balance such that it should not lose its relevance, and not be of any use.

Simplification can happen at several levels like the choice of functions, number of model parameters, degree of polynomials, etc. These decisions are made by the model designer. Regularization is the simplification process carried by the training algorithm, to control the model complexity.

The regression does not account for model complexity - it only tries to minimize the error (e.g. MSE). In regularized regression, the objective function has two parts - the **error term** and the **regularization term**. To solve our problem of overfitting for regression, regularization involves adding a regularization term to the cost that adds up the absolute values or the squares of the parameters of the model.

Question-2:

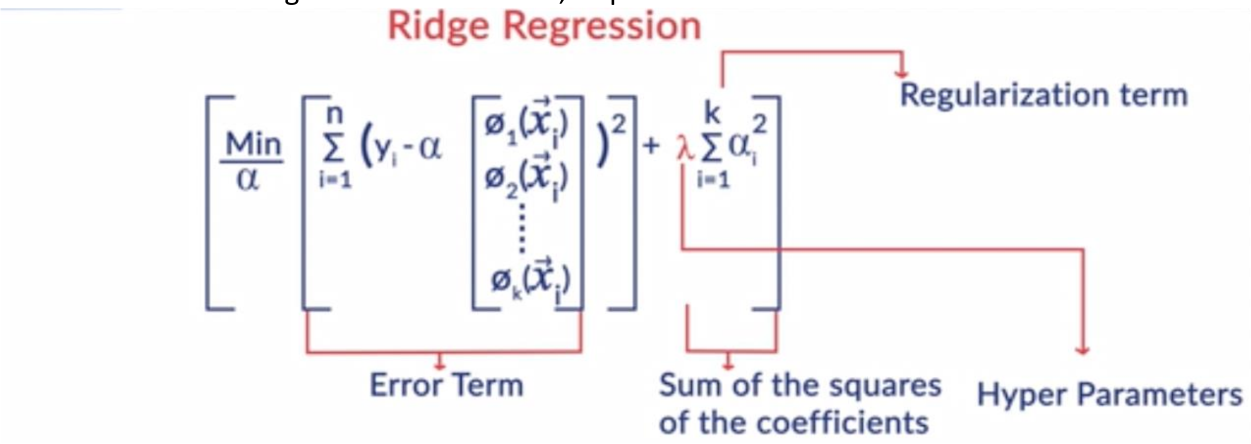
List at least 4 differences in detail between L1 and L2 regularization in regression.

Answer-2:

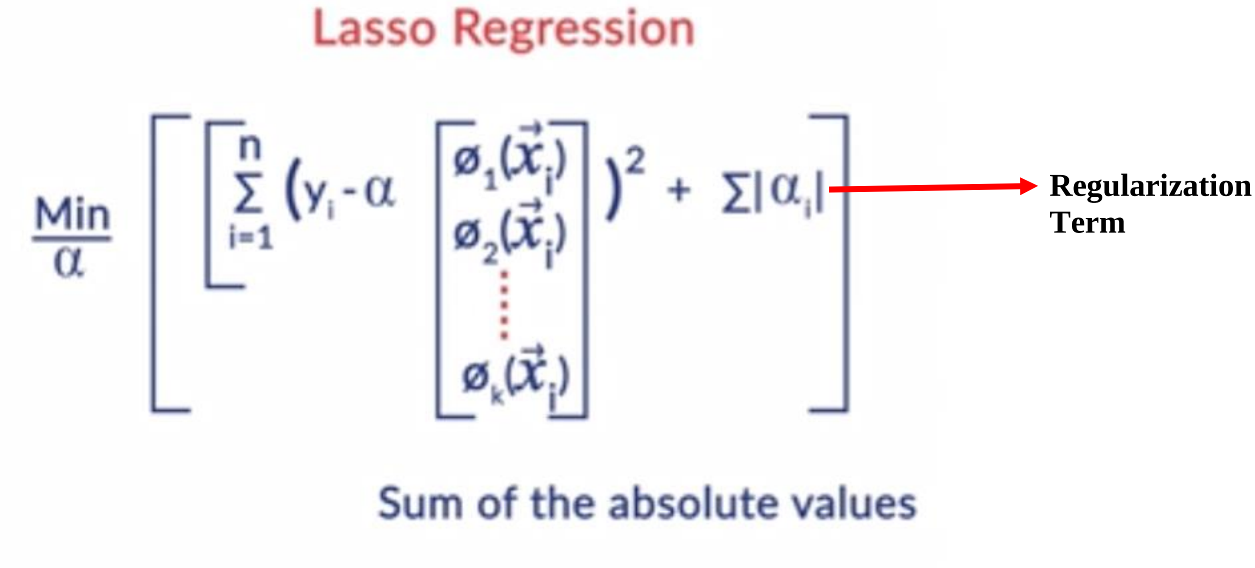
A regression model that uses L1 regularization technique is called Lasso Regression, and a model that uses L2 regularization is called Ridge Regression. The main difference between the two are listed below:

Cost Function

In ridge regression, an additional term of "sum of the squares of the coefficients" is added to the cost function along with the error term, as per the below formula:


Ridge Regression
Regularization term
$\underset{\alpha}{\mathrm{Min}} \left[\left[\sum_{i=1}^{n} \left(y_i - \alpha \begin{bmatrix} \phi_1(\vec{x}_i) \\ \phi_2(\vec{x}_i) \\ \vdots \\ \phi_k(\vec{x}_i) \end{bmatrix} \right)^2 \right] + \lambda \left[\sum_{i=1}^{k} \alpha_i^2 \right] \right]$
Error Term
Sum of the squares of the coefficients
Hyper Parameters

In case of lasso regression, a regularisation term of "sum of the absolute value of the coefficients" is added, as per the below formula:


Lasso Regression
$\underset{\alpha}{\mathrm{Min}} \left[\left[\sum_{i=1}^{n} \left(y_i - \alpha \begin{bmatrix} \phi_1(\vec{x}_i) \\ \phi_2(\vec{x}_i) \\ \vdots \\ \phi_k(\vec{x}_i) \end{bmatrix} \right)^2 + \sum |\alpha_i| \right] \right]$
Regularization Term
Sum of the absolute values

Feature Selection

Ridge and Lasso perform different measures of shrinkage which depend on the value of hyper parameter “Lambda”. In the process of shrinkage, Lasso shrinks the less important feature’s coefficient to zero thus, removing some feature altogether. So, this works well for **feature selection** in case we have a huge number of features.

Computational Efficiency

L1-norm does not have an analytical solution, but L2-norm does. This allows the L2-norm solutions to be calculated computationally efficiently. However, L1-norm solutions does have the sparsity properties which allows it to be used along with sparse algorithms, which makes the calculation more computationally efficient.

Sparsity

It refers to only a few entries in a matrix (or vector) are non-zero. L-1 norm has the property or producing many coefficients with zero values, or very small values with few large coefficients. This results in L-1 regularization producing sparse output.

On the other hand, L2-norm reduces the coefficients to low values, though not zero. Thus, it is not able to produce sparse output.

Question-3:

Consider two linear models

L1: y = 39.76x + 32.648628

And

L2: y = 43.2x + 19.8

Given the fact that both the models perform equally well on the test dataset, which one would you prefer and why?

Answer-3:

Ocam’s Razor – a fundamental tenet to all Machine Learning states that a predictive model has to be as simple as possible, but no simpler. The thumb rule states that from the two models available that show similar performance in the finite train or test data set, we should choose the model that makes fewer assumptions about the data that is yet to be seen; or follows a generalization property. This essentially means that we need to choose the “simpler” of the two models.

Both L1 and L2 are linear models, but L1 has more complex coefficients. Based on the thumb rule, we should choose L2, primarily because of the fact that for L1 is a more complex model (complex coefficients) and can run into the problem of overfitting. Essentially, L2 is a simpler model than L1. There is rather deep relationship between the complexity of model and its usefulness in a learning context. We elaborate on this relationship below:

1. Simpler models are usually more ‘generic’ and more widely applicable. They follow generalization properties.
2. Simpler models require fewer training set and the sample complexity required is lower. This makes them easier to train.
3. Complex models work very well for the training set, but fall miserable when applied to the test set; leading to overfitting. Simpler models however, make more errors in the training set but serve with greater predictability than the complex models. This is a trade-off required for better predictability.
4. Simpler models are more robust as they have low variance and high bias while the complex model have a low bias, and high variance. ‘Variance’ refers to variance in the model and ‘bias’ is the deviation from expected behavior. This phenomenon is referred as bias-variance tradeoff. It signifies that, it is better for a model to not be too sensitive about the specifics of the data set on which it is trained. It should rather pick the essential characteristics which is invariant across the any training data set.

Question-4:

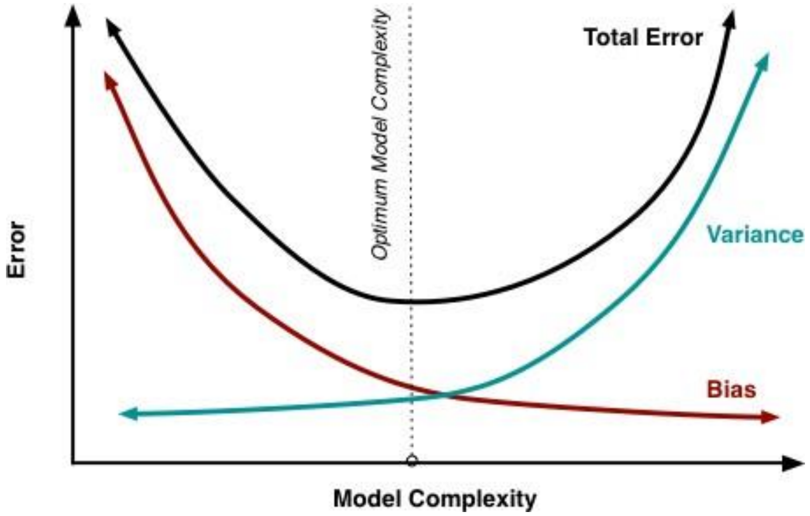
How can you make sure that a model is robust and generalisable? What are the implications of the same for the accuracy of the model and why?

Answer-4:

In previous questions, we have seen that simple model are more robust and generalizable, but choosing just simple model is not enough. We should test the accuracy as well on train and test data set, and if there are two model which perform equally well on test data set then we should always go for the simpler model, as simpler model more robust and generalizable.

Complex models are more sensitive and learn the peculiarity of the training data set. Owing to this sensitivity, complex models is very unstable for changing data sets. However, Simpler models pick the essential characteristics and patterns of the data points. Therefore, a simpler model is more robust and unlikely to change wildly if more points are added, or removed or if the points are varied.

Now there is a tradeoff between variance and bias to select optimal model. The 'variance' of a model is the variance in its output on some test data with respect to changes in the training dataset. In other words, variance refers to the degree of changes in the model itself with respect to changes in the training data. Bias quantifies how accurate is the model likely to be on future (test) data.



The figure illustrates the typical trade-off between bias and variance. Low complexity models have high bias and low variance. Opposite is applicable for complex models which have low bias and high variance. The best model for a task is the one that balances both bias and variance; without compromising too much on accuracy.

To achieve this in machine learning, an effective technique is regularization. **Regularization** is a process used to create an optimally complex model, i.e. a model which is as simple as possible while performing well on the training data. Through this, our aim is to adjust the model so that it becomes simple, and yet we should keep the balance such that it should not lose its relevance, and not be of any use. Simplification can happen at several levels like the choice of functions, number of model parameters, degree of polynomials, etc. These decisions are made by the model designer. Regularization is the simplification process carried by the training algorithm, to control the model complexity. Some typical regularization steps for various classes of ML algorithms:

1. For regression, regularization involves adding a regularization term to the cost that adds up the absolute values or the squares of the parameters of the model.
2. For Decision Trees, this could mean 'pruning' the tree to control its depth and/or size
3. For Neural Networks, a common strategy is to include a 'dropout' – dropping a few neurons or weights at random.

Question-5:

As you have determined the optimal value of lambda for ridge and lasso regression during the

assignment, which one would you choose to apply and why?

Answer-5:

Ridge and Lasso perform different measures of shrinkage which depend on the value of hyper parameter "Lambda". In the process of shrinkage, Lasso shrinks the less important feature's coefficient to zero thus, removing some feature altogether. So, this works well for **feature selection** in case we have a large number of features.

Since we have a huge set of variables available in our data set, we will use lasso regression as it will shrinks the less important feature's coefficient to zero thus, removing some feature altogether. And we have got almost same r2 score for ridge and lasso regression with optimal alpha value.

Values for optimal alpha and corresponding r2 score for ridge and lasso regression is given below:

Optimal value of alpha of lasso regression is 50 and r2 score for optimal value of alpha is given below

R2 score for train : 0.9372405328256925
R2 score for test : 0.9254664123086983

Optimal value of alpha of ridge regression is 4 and r2 score for optimal value of alpha is given below

R2 score for train : 0.9371096095852764
R2 score for test : 0.9253982765709686

Optimal value of alpha is 1 for ridge regression on variables selected by lasso regession and r2 score for optimal value of alpha is given below

R2 score for train : 0.9392476999525955
R2 score for test : 0.9231948226312643

From above r2 score for optimal value of alpha, we can see that both the model ridge and lasso give almost same r2 score. Hence, we will choose lasso regression which will do the variable selection as well with r2 score.