Question 1

Rahul built a logistic regression model with a training accuracy of 97% and a test accuracy of 48%. What could be the reason for the gap between the test and train accuracies, and how can this problem be solved?

Ans -1 As the training accuracy is far higher than test accuracy, this is a case of overfitting,

In predictive modelling, "signal" is the true underlying pattern that we wish to learn from the given data at hand and "Noise" is the irrelevant information or the unwanted randomness that we want to avoid.

For example, let's say you're modelling height vs. age in children. If you sample a large portion of the population, you'd find a pretty clear relationship between the height and age:However, if you only sample one local school, the relationship might not be that clear. A small sample is usually affected by outliers (e.g. kid whose dad is an NBA player) and randomness (e.g. kids who hit puberty at different ages).

If the algorithm is too complex or flexible (e.g. it has too many input features or it's not properly regularized), it can end up "memorizing the noise" instead of learning only the signal pattern.

A model that has learned the noise together with the signal is considered "overfit" because it fits the training dataset but has poor fit with new datasets. And in such a case when you look at the model accuracy over train and test data the training accuracy comes higher and test accuracy (kind of new or unseen data) comes lower.

In order to prevent overfitting, the various methods used are

1) **Cross-validation** Use your initial training data to generate multiple mini train-test splits. Use these splits to tune your model. Since you will be covering all the data even with lower available data you will be successful in achieving a much better model

2) **Train with more data**
It won't work every time, but training with more data can help algorithms detect the signal better. In the earlier example of modelling height vs. age in children, it's clear how sampling more schools will help your model.

3) **Remove features**
Some algorithms have built-in feature selection.
For those that don't, you can manually improve their generalizability by removing irrelevant input features. This can be achieved by using RFE or by removing highly correlated or insignificant variables.

4) **Early stopping**

When you’re training a model iteratively, you can measure how well each iteration of the model performs.
Till a certain number of iterations, new iterations improve the model. After that point, however, the model’s ability to generalize can weaken as it begins to overfit the training data. You should stop before this point reaches.

5) **Regularization**
Regularization refers to a broad range of techniques for artificially forcing your model to be simpler. For eg you could reduce the levels in decision tree, reduce no of coefficients in model. Oftentimes, the regularization method is a hyperparameter which can be tuned through cross-validation.

6) Bagging and Boosting
Bagging attempts to reduce the chance of overfitting complex models .It trains a large number of "strong" learners in parallel and then combines all the strong learners together in order to "smooth out" their predictions

Boosting attempts to improve the predictive flexibility of simple models. It trains a large number of "weak" learners in sequence and then combines all the weak learners into a single strong learner.

Question 2

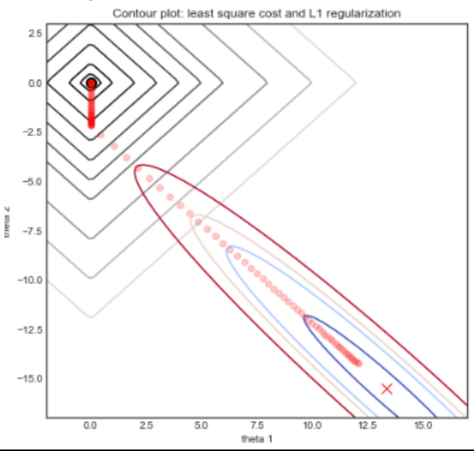
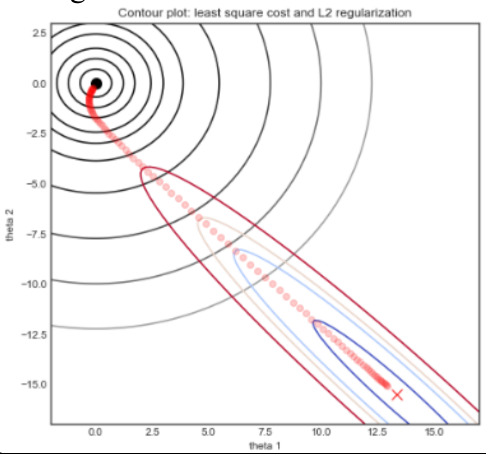
List at least four differences in detail between L1 and L2 regularisation in regression.

Ans -2

A regression model that uses Lasso Regression regularization method is also called L1 regularization and a regression model which uses the Ridge Regression regularization technique is also called L2 regularization.

The differences between these two are :-

S.No	L1 (Lasso Regularisation)	L2 (Ridge Regularization)
1	The penalty term or the regularization term in case of L1 is $\lambda \sum_{j=1}^{p} \lvert\beta_j\rvert$ The Cost function is $\sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \lvert\beta_j\rvert$ Cost function	The penalty or regularization term in case of L2 is $\lambda \sum_{j=1}^{p} \beta_j^2$ The Cost function becomes $\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$ Cost function

2	L1 results in sparse solutions and shrinks the less important feature’s coefficient or weight to 0 and can be used for Feature Selection	L2 is not recommended for feature selection
3	L1 is computationally intensive	L2 is not that computationally intensive
4	L1 Regularization Contour 	L2 regularization Contour 
5	L1 does not have a nice clean mathematical formula and requires iterative procedure to arrive at an appropriate alpha	L2 has a nice clean mathematical formula and we can directly arrive at optimum value.

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 data set, which one would you prefer and why?

Ans -3

I would choose L2. The reason being it is simpler than L1. L1 model requires more bits to be represented. L2 has only one decimal precision for the first term and second term, but L1 has two decimal precisions in first term and 6 decimal precisions in the second term, For representing the L1 we would require at least one float for first term and one double for second term which makes it a bit complex to represent.

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?

Ans -4 In order for the model to be robust and generalizable it is required that it performs well on unseen data. The model will perform well on unseen data if it is simple and only captures the signal from the data and not any noise. Such a model will have almost similar training and test accuracy. The reason behind the model having almost similar training and test accuracy is because the model has only picked up the generalizable signal pattern while training on the train data and this pattern will be similar across entire spread of data. So it does not matter which particular segment of data you choose the accuracy will be near about same. In order to make a model simple sometimes a technique called regularization is applied to intentionally make complex models simple. The regularization technique has a hyper parameter which decides how much simpler the model will be. A higher value of hyper parameter will force the model to be more simple. But simplicity introduces bias. So a tradeoff is necessary in such cases.

Question 5

You have determined the optimal value of lambda for ridge and lasso regression during the assignment. Now, which one will you choose to apply and why?

Answer After finding the optimal value of lambda for ridge and lasso during the assignment, we have chosen Ridge on the basis of model evaluation parameters R2 Accuracy.

From the Model Evaluation we have R-Squared Score for both Train and Test Data respectively :-
Ridge (0.8744,0.8242) is better than
Lasso (0.7907,0.7645)

So, we will apply Ridge.