Assignment-Linear Regression

Name: Rajesh Bisht

Email Id: rbisht.india@gmail.com

Batch-8

Git:

https://github.com/RajeshBisht28/LinearRegressionAssignment.git

[Above Git Hub link includes, Jupyter notebook]

Assignment-Linear Regression

A. Explain the main assumptions of Linear Regression in detail.

Answer A:

Linear regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables. For linear regression to produce reliable results, certain assumptions must be satisfied. Assumptions of linear regression:

Linearity:

he relationship between the independent and dependent variables should be linear. This means that a change in the independent variable(s) will lead to a proportional change in the dependent variable.

Independence:

The observations should be independent of each other. This means that the residuals (errors) should not be correlated with each other.

Homoscedasticity:

The residuals (errors) should have constant variance at all levels of the independent variable(s). This means that the spread or "scatter" of the residuals should be the same for all predicted values.

No Multicollinearity:

If multiple independent variables are used, they should not be highly correlated with each other. Multicollinearity can inflate the variance of the coefficient estimates and make them unstable.

Conclusion:

When using linear regression, it is crucial to verify that these assumptions hold. Violations can lead to misleading results, so various diagnostic tests and plots should be used to assess these assumptions. If assumptions are violated, it may be necessary to consider transforming variables, using different modeling techniques, or employing robust regression methods.

B. What is the difference between R-squared and Adjusted R-squared?

Answer B:

R-squared and **Adjusted R-squared**

Assignment-Linear Regression

Both statistical measures used to assess the goodness-of-fit of a regression model, but they serve slightly different purposes and are interpreted in distinct ways. Here's a detailed comparison of the two:

R-squared (R^2):

R-squared is the proportion of the variance in the dependent variable that can be explained by the independent variable(s) in a regression model.

Formula is : $R2=1-(SStot / SSres)$

Where SSres is the sum of the squared residuals (the differences between observed and predicted values), and SStot is the total sum of squares (the variance of the dependent variable).

R-squared values range from 0 to 1, 0 indicates that the model explains none of the variance. 1 indicates that the model explains all the variance.

A higher R-squared value means a better fit for the model; however, it does not necessarily indicate that the model is a good one. It simply quantifies how well the model accounts for variability in the data.

Limitations:

Increases with More Predictors:

R-squared will always increase when more predictors are added to the model, regardless of whether those predictors are truly meaningful. This can lead to overfitting, where the model becomes too complex and captures noise rather than the underlying relationship.

Doesn't Penalize Complexity:

R-squared does not take into account the number of predictors in the model, which can give a misleading impression of model quality.

Adjusted R-squared (Adjusted R^2):

Adjusted R-squared modifies the R-squared value to account for the number of predictors in the model. It provides a more accurate measure of goodness-of-fit when comparing models with different numbers of predictors.

Formula: Adjusted $R2 = 1 – [(1-R2)*(n-1)/(n-k-1)]$

R2: The R2 of the model

n: The number of observations

k: The number of predictor variables.

Adjusted R-squared can be negative, though it typically ranges from 0 to 1. A negative value indicates that the model is worse than a simple mean-based model.

Assignment-Linear Regression

Adjusted R-squared provides a better measure of how well the model generalizes to other
datasets. It only increases when a new predictor improves the model more than would be expected by chance, thus helping to avoid overfitting.

Advantages:

Penalizes for Complexity:

Unlike R-squared, Adjusted R-squared accounts for the number of predictors, making it a better criterion for model selection when comparing models with different numbers of predictors.

Provides a More Accurate Measure of Fit:

It offers a more reliable assessment of how well the independent variables explain the variability in the dependent variable.

So, while R-squared is a useful statistic for understanding how well a model fits the data, Adjusted R-squared is generally preferred when comparing models with different numbers of predictors. It provides a more nuanced view of model performance, accounting for both the goodness-of-fit and the complexity of the model.

C. What are the different types of Regularization techniques in Regression. Explain in detail with cost functions of each technique.

Answer-C:

In regression, regularization techniques are used to prevent overfitting by penalizing large coefficients in the model. These techniques add a regularization term to the cost function, discouraging complex models by shrinking the coefficients. The main types of regularization techniques in regression are:

1. Lasso Regression (L1 Regularization)
2. Ridge Regression (L2 Regularization)
3. Elastic Net Regression.

Lasso Regression (L1 Regularization):

Lasso Regression

(Least Absolute Shrinkage and Selection Operator) is a type of linear regression that incorporates L1 regularization. In Lasso, a penalty equivalent to the absolute values of the regression coefficients is added to the cost function. This regularization forces some coefficients to be exactly zero, effectively performing feature selection.
The objective of Lasso regression is to minimize the following cost function

Assignment-Linear Regression

Cost Function:

In Lasso regression, the regularization term is the sum of the absolute values of the coefficients (L1 norm). This form of regularization can result in sparse models, where some coefficients are reduced to zero, effectively selecting features.

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(y^{(i)} - \hat{y}^{(i)} \right)^2 + \lambda \sum_{j=1}^{n} |\theta_j|$$

m is the number of training examples.
y(i) is the actual target value, and y Hat(i) is the predicted target value.
J(θj is the coefficient of the J-th feature.
λ is the regularization parameter that controls the strength of regularization.

Effect:

Lasso tends to shrink some coefficients to exactly zero, leading to sparse models. This property is useful for feature selection.

Ridge Regression (L2 Regularization):

Ridge Regression (also known as L2 Regularization) is a type of linear regression that includes a regularization term to prevent overfitting by penalizing large coefficients. The regularization term adds the square of the magnitude of the coefficients (i.e., L2 norm) to the loss function.
The goal of Ridge Regression is to minimize the following objective function:

Cost Function:

In Ridge regression, the regularization term is the sum of the squared values of the coefficients (L2 norm). This type of regularization penalizes large weights but does not drive them to zero, unlike Lasso.

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(y^{(i)} - \hat{y}^{(i)} \right)^2 + \lambda \sum_{j=1}^{n} \theta_j^2$$

Effect:

Assignment-Linear Regression

Ridge tends to shrink the coefficients but keeps them all non-zero, which helps reduce model complexity without eliminating any features.

Elastic Net Regression:

Elastic Net Regression is a type of regularized regression technique that combines the properties of both Lasso (L1 regularization) and Ridge (L2 regularization) regression. It is particularly useful when there are many predictors that are highly correlated or when the number of predictors exceeds the number of observations.

Cost Function:

Elastic Net combines both L1 and L2 regularization, mixing the penalties of both Ridge and Lasso. It includes both the absolute value (L1) and squared value (L2) terms in the regularization.
Where λ1\lambda_1λ1 controls the L1 penalty and λ2\lambda_2λ2 controls the L2 penalty.

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(y^{(i)} - \hat{y}^{(i)} \right)^2 + \lambda_1 \sum_{j=1}^{n} |\theta_j| + \lambda_2 \sum_{j=1}^{n} \theta_j^2$$

Effect:

Elastic Net can handle situations where Lasso performs poorly (e.g., highly correlated features). It retains the feature selection ability of Lasso while also addressing the limitations of Lasso by incorporating the Ridge penalty, which helps maintain model stability.

D. How logistic regression works for multiclass classification. Explain in detail.

Answer-D:

Logistic regression is a classification algorithm traditionally used for binary classification tasks. However, it can be extended to handle multiclass classification problems as well. There are two main ways to extend logistic regression to deal with multiple classes: one-vs-rest (OvR) and multinomial logistic regression (also known as softmax regression).

Assignment-Linear Regression

One-vs-Rest (OvR) or One-vs-All (OvA) Approach:

In this approach, the multiclass classification problem is broken down into multiple binary classification problems. Each class is treated as a separate binary classification problem, where the goal is to distinguish that class from all the other classes.

How It Works:

- For a problem with K classes, K separate binary classifiers are trained.
- Each classifier is responsible for predicting whether a sample belongs to its class (positive class) or one of the other K-1 classes (negative class).
- The input is run through each classifier, and the classifier that outputs the highest probability determines the class of the input.

For example, if we have 3 classes: A, B, and C, we will create 3 separate logistic regression models:

- Classifier 1: Classifies between class A and {B, C}.
- Classifier 2: Classifies between class B and {A, C}.
- Classifier 3: Classifies between class C and {A, B}.

Decision Process:

After training, for a new input, each of the classifiers provides a probability score. The class with the highest score is selected as the predicted class.

Pros:

- Conceptually simple and easy to implement.
- Can use any binary classification algorithm for each class.

Cons:

- Computationally expensive if the number of classes is large since it trains multiple models.
- The classes are not treated jointly, which could lead to suboptimal results if the classes are not linearly separable.

Multinomial Logistic Regression (Softmax Regression):

In this approach, logistic regression is generalized to directly handle multiclass classification by using a softmax function.

How It Works:

- For K classes, the model computes K linear functions (one for each class).
- These functions are then passed through a softmax function to convert the output into a probability distribution over all the classes.

Softmax Function:

For an input X and K classes, the model computes:

Assignment-Linear Regression

$$z_k = w_k \cdot x + b_k \quad \text{for each class } k = 1, 2, \ldots, K$$

Where Wk is the weight vector and bk is the bias term for class k. The softmax function converts these linear scores zk into probabilities:

$$P(y = k|x) = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}} \quad \text{for each class } k = 1, 2, \ldots, K$$

The sum of these probabilities is 1, and each probability represents how likely the input belongs to each class.

Training:

- The model is trained using cross-entropy loss for multiclass classification. The goal is to maximize the likelihood of the true class labels.
- The optimization process (such as gradient descent) adjusts the weights wk and biases bk to minimize the loss.

Prediction:

- For a new input, the model computes the linear scores and applies the softmax function to obtain class probabilities.
- The class with the highest probability is chosen as the predicted class.

Pros:

- Softmax regression treats all classes jointly, so it can model the relationship between classes better than OvR.
- It is mathematically elegant and typically more efficient than the OvR approach.

Cons:

- It assumes that the classes are mutually exclusive, which may not always be true in some tasks.
- May not perform well if the data is highly imbalanced.

Comparison: One-vs-Rest vs. Multinomial Logistic Regression:

One-vs-Rest (OvR):

- Simpler to understand and implement.
- More flexible since each classifier can have its parameters tuned independently.
- Can be slower due to training multiple models.

Assignment-Linear Regression

Multinomial Logistic Regression (Softmax):

- More computationally efficient since it trains a single model for all classes.
- Treats the relationship between all classes simultaneously, leading to better performance when class relationships are important.
- Can struggle with class imbalance or overlapping class boundaries.

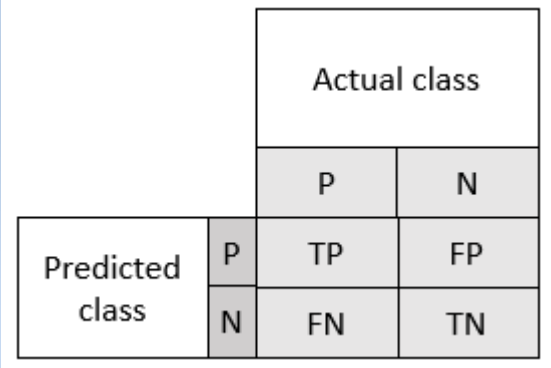
E. Explain the performance metrics of logistic regression.

Ans E:

Logistic regression is used for binary classification tasks, where the goal is to predict one of two possible outcomes. To evaluate its performance, a variety of metrics can be used.

Confusion matrix:

Confusion matrix is method used to summarize classification algorithm on set of test data for which the true values are previously known. Sometime it also refer as error matrix. Confusion matrix will look like this,



P = Positive; N = Negative; TP = True Positive; FP = False Positive; TN = True Negative; FN = False Negative.

True positive :

TP means model predicted yes and correct answer for that is also yes

True negative :

TN means model predicted no and correct answer for that is also no.

False positive :

FP means model predicted yes but actual answer is no

False negative :

FN means model predicted no but actual answer is yes

So there is list of rate calculated using this matrix

Assignment-Linear Regression

1) Accuracy = (TP+TN/Total) tells about overall how classifier Is correct.

2) True positive rate = TP/(actual yes) it says about how much time yes is predicted correctly. It is also called as “sensitivity” or “recall”

3) False positive rate = FP/(actual number) it says about how much time yes is predicted when actual answer is no

4) True negative rate = TN/(actual number) it says about how much time no is predicted correctly and actual answer is also no. it is also known as “specificity”

5) Misclassification rate = (FP+FN)/(Total) it is also known as error rate and tells about how often our model is wrong

6) Precision = (TP/ (predicted yes)) if it predict yes then how often it is correct ?

7) Prevalence = (actual yes /total) how often yes condition really/actually occurs.

Value of actual data is

Variable	Categories	Frequencies	%
Renewed	0	7	35.000
	1	13	65.000

Now by computing confusion matrix our model will get

from \ to	0	1	Total	% correct
0	17	7	24	70.83%
1	6	30	36	83.33%
Total	23	37	60	78.33%

Here we can see that our model is 78.33% correct in predicting. Now let us find the same for our 2nd model where I have changed value of parameter 18, 19,20,
Actual data will be,

Variable	Categories	Frequencies	%
Renewed	0	8	40.000
	1	12	60.000

Assignment-Linear Regression

And confusion matrix will be,

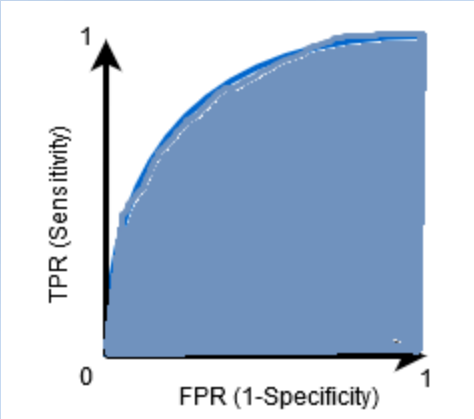
from \ to	0	1	Total	% correct
0	1	7	8	12.50%
1	0	12	12	100.00%
Total	1	19	20	65.00%

It is clearly observed that first model is more accurate compare to second.

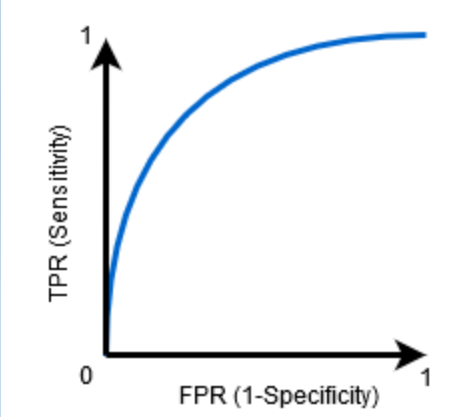
ROC curve (Receiver Operating Characteristic):

ROC- Receiver operating characteristic curve will help to summarize model's performance by calculating trade-offs between TP rate (sensitivity) and FN rate (1-specificity) it will plot these 2 parameters. To classify this term AUC (Area under the curve) is introduced which gives summary of ROC curve.

AUC simply means area situated under the curve in below graph blue colored region is called as AUC

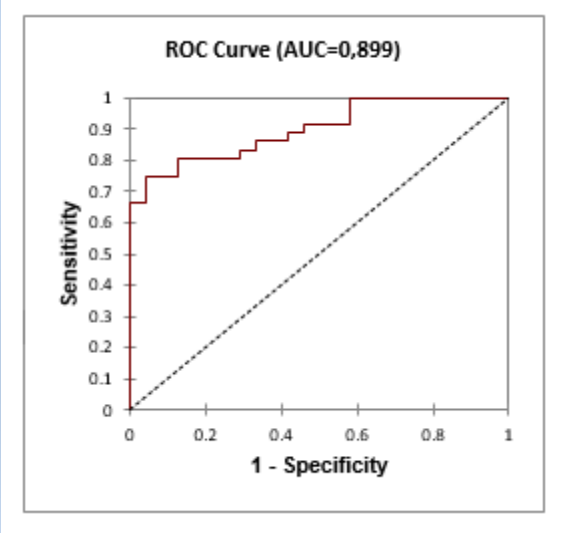


The higher the AUC, the better the performance of classifier. ROC curve may look like this,

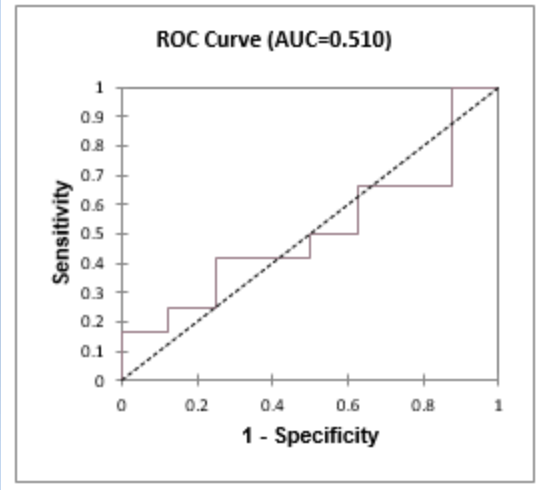


Assignment-Linear Regression

AUC is classified on following basis, If AUC = 1, then the classifier is able to perfectly distinguish between all the Positive and the Negative class points correctly
1) If AUC =0 then classifier is predicting all the positive as negative and negative as positive.
2) If 0.5< AUC < 1 means classifier will distinguish the positive class value from negative class value because it is finding more number of TP and TN compare to FP and FN.
3) If AUC = 0.5 it means classifier is not able to distinguish between positive and negative values.
So we can conclude that higher the value of AUC better its ability to distinguish between positive and negative classes.
Now let's find value of AUC for our both model and compare it, ROC for first model is,



ROC for second model is,



Here we can see that value of AUC in first model is 0.899 while in second it is 0.510 which means our first model is good compare to our 2nd model and, in first model it is

Assignment-Linear Regression

predicting some of negative value as positive while in 2nd it is very close to 0.5 means our classifier is not able to distinguish between positive and negative value.

F. Use the Mobile price prediction dataset from below Kaggle link and create an end to end project on Jupyter/Colab.

Answer: MobilePricePrediction.ipynb is on below Git Link
https://github.com/RajeshBisht28/LinearRegressionAssignment.git

https://www.kaggle.com/datasets/mohannapd/mobile-price-prediction/data

i. Download the dataset from above link and load it into your Python environment.
ii. Perform the EDA and do the visualizations.
iii. Check the distributions/skewness in the variables and do the transformations if required.
iv. Check/Treat the outliers and do the feature scaling if required.
v. Create a ML model to predict the price of the phone based on the specifications given.
vi. Check for overfitting and use the Regularization techniques if required
vii. Compare the performance metrics of training dataset and testing dataset for all the different algorithms used (Linear/Ridge/Lasso/ElasticNet)