**CHAPTER 1: Supervised Learning (Regression Models)**

**Theory**

In the previous chapter, we studied the introduction to machine learning, what is supervised learning, what is unsupervised learning, what is a classification problem, understanding the k-nearest neighbor’s algorithm, evaluation metrics for a classification model, model fitting, and hyperparameter tuning. In this chapter, we will look at bias-variance trade-off, what is regression, what is simple linear regression, what is multiple linear regression, evaluating regression models, and finally Logistic Regression for classification.

Let’s start our chapter with bias and variance.

**What is Bias?**

In simple terms, Bias is the difference between the Predicted values and Expected values. To explain further, the model makes certain assumptions when it trains on the data provided. When it is introduced to the testing/validation data, these assumptions may not always be correct.

In our model, if we use a large number of nearest neighbors, the model can totally decide that some parameters are not important at all. For example, it can just consider that the Glucose level and the Blood Pressure decide if the patient has diabetes. This model would make very strong assumptions about the other parameters not affecting the outcome. You can also think of it as a model predicting a simple relationship when the data points clearly indicate a more complex relationship.

So Bias is basically a part of a generalization error that comes across due to some wrong assumptions that we make during our development process i.e., assuming that the data is linear when it is actually quadratic.

A picture containing chart

Description automatically generated

Mathematically, let the input variable be X and a target variable Y. We map the relationship between the two using a function f.

Therefore,

Logo

Description automatically generated with low confidence

Here ‘e’ is the error that is normally distributed. The aim of our model f’(x) is to predict values as close to f(x) as possible. Here, the Bias of the model is:

A picture containing text

Description automatically generated

As I explained above, when the model makes the generalizations i.e., when there is a high bias error, it results in a very simplistic model that does not consider the variations very well. Since it does not learn the training data very well, it is called **Underfitting**.

**What is Variance?**

Contrary to bias, the Variance is when the model takes into account the fluctuations in the data i.e., the noise as well. So, what happens when our model has high variance?

The model will still consider the variance as something to learn from. That is, the model learns too much from the training data, so much so, that when confronted with new (testing) data, it is unable to predict accurately based on it.

So, variance comes into play whenever our model is extremely sensitive to minor variations in the training data. If the model is having many degrees of freedom like a high degree polynomial model is likely to have high variance.

A high-variance model is most likely to **Overfit** our training data.

Mathematically, the variance error in the model is:



Since in the case of high variance, the model learns too much from the training data, it is called **overfitting**.

In the context of our data, if we use very few nearest neighbors, it is like saying that if the number of pregnancies is more than 3, the glucose level is more than 78, Diastolic BP is less than 78, Diastolic BP is less than 98, Skin thickness is less than 23 mm and so on for every feature… decide that the patient has diabetes. All the other patients who don’t meet the above criteria are not diabetic. While this may be true for one particular patient in the training set, what if these parameters are the outliers or were even recorded incorrectly? Clearly, such a model could prove to be very costly!

Additionally, this model would have a high variance error because the predictions of the patient being diabetic or not vary greatly with the kind of training data we are providing it. So even changing the Glucose Level to 75 would result in the model predicting that the patient does not have diabetes.

To make it simpler, the model predicts very complex relationships between the outcome and the input features when a quadratic equation would have sufficed. This is how a classification model would look like when there is a high variance error/when there is Overfitting.

Diagram

Description automatically generated

**Bias-Variance Trade-off**

How do we relate the above concepts to our KNN model from earlier? Let’s find out!

In our model, say, for, k=1, the point closest to the datapoint in question will be considered. Here, the prediction might be accurate for that particular data point so the bias error will be less.

However, the variance error will be high since only the one nearest point is considered and this doesn’t consider the other possible points. What scenario do you think this corresponds to? Yes, you are thinking right, this means that our model is overfitting.

On the other hand, for higher values of k, many more points closer to the datapoint in question will be considered. This would result in higher bias error and underfitting since many points closer to the datapoint are considered and thus it can’t learn the specifics from the training set. However, we can account for a lower variance for the testing set which has unknown values.

To achieve a balance between the Bias error and the Variance error, we need a value of k such that the model neither learns to form the noise (overfit of data) nor makes sweeping assumptions on the data (underfit on data). To keep it simpler, a balanced model would look like this:

Diagram

Description automatically generated

Though some points are classified incorrectly, the model generally fits most of the data points accurately. The balance between the Bias error and the Variance error is the **Bias-Variance Trade-off.**

The following bulls-eye diagram explains the trade-off better.

Diagram, shape, circle

Description automatically generated

As depicted in the figure, our goal is to minimize the variance as well as Bias to get accurate results. If there are a lesser number of predictors in the data set than to having big coefficients, it is a possibility that the model obtained is favored towards those predictors i.e., a small change in those predictors leads to a big change in the predicted result. In other words, that predictor is driving our model, or our model biases for that predictor. An easy way to overcome this is to introduce more predictors which contribute towards the result of the prediction. This increases the complexity of the model or in other words, the variance in the model increases.

Mathematically,



Irreducible error is the error due to noise in the data.

Graphically it looks like:

Diagram

Description automatically generated

As the bias decreases, the variance increases. So, we need to find a balance so that we get a model which has low variance as well as low bias. This is called a bias-variance trade-off.

**Exercise**

We will be working with the diabetes dataset to understand the Bias-Variance Trade-off with the KNN Classification algorithm.

We will do this by importing the necessary libraries:

Text

Description automatically generated

Now, we will load the data into a data frame and observe some rows to get insights into the data.

Text

Description automatically generated

Graphical user interface, application

Description automatically generated

We need to predict the “Outcome” column. Let us separate it and assign it to a target variable ‘y’. The rest of the data frame will be the set of input variables x.

A picture containing diagram

Description automatically generated

Now let’s scale the predictor variables and then separate the training the testing data.

Text

Description automatically generated

Since the outcomes are classified in a binary form, we will use the simplest K-nearest neighbor classifier (KNN) to classify whether the patient has diabetes or not.

However, how do we decide the value of ‘k’?

* Maybe we should use k = 1 so that we will get very good results on our training data? That might work, but we cannot guarantee that the model will perform just as well on our testing data since it can get too specific.
* How about using a high value of k, say like k = 100 so that we can consider a large number of nearest points to account for the distant points as well? However, this kind of model will be too generic, and we cannot be sure if it has considered all the possible contributing features correctly.

Let us take a few possible values of k and fit the model on the training data for all those values. We will also compute the training score for all those values.

Text, letter

Description automatically generated

To derive more insights from this, let us plot the training data(in red) and testing data(in blue).

Text

Description automatically generated

Chart, line chart

Description automatically generated

To calculate the scores for a particular value of k,

Text, letter

Description automatically generated



We can make the following conclusions from the above plot:

* For low values of k, the training score is high, while the testing score is low.
* As the value of k increases, the testing score starts to increase, and the training score starts to decrease.
* However, at some value of k, both the training score and the testing score are close to each other.

So, what do you think is the optimum value for k?

From the above explanation, we can conclude that the k for which

* The testing score is the highest, and
* Both the test score and the training score are close to each other.

Is the optimal value of k. So, even though we are compromising on a lower training score, we still get a high score for our testing data which is more crucial – the test data is after all unknown data.

Now, let’s understand the concept of regression analysis.

**Regression Analysis**

**What is Regression Analysis?**

Regression is a parametric technique used to predict continuous (dependent) variables given a set of independent variables. It is parametric in nature because it makes certain assumptions (discussed next) based on the data set. If the dataset follows those assumptions, regression gives incredible results. Otherwise, it struggles to provide convincing accuracy. Don’t worry. There are several tricks (we’ll learn shortly) we can use to obtain convincing results.

Diagram

Description automatically generated

Regression analysis includes several variations, such as linear, multiple linear, and nonlinear. The most common models are simple linear and multiple linear. Nonlinear regression analysis is commonly used for more complicated data sets in which the dependent and independent variables show a nonlinear relationship.

**The use of Regression**

Regression analyses the relationship between two or more features. Let’s take an example:

Let’s suppose we want to make an application that predicts the chances of admission of a student to a foreign university. In that case, the

The benefits of using Regression analysis are as follows:

* It shows the significant relationships between the label (dependent variable) and the feature (independent variable).
* It shows the extent of the impact of multiple independent variables on the dependent variable.
* It can also measure these effects even if the variables are on a different scale.

These features enable the data analysts to find the best set of independent variables for predictions.

**Linear Regression**

Linear Regression is one of the most fundamental and widely known Machine Learning Algorithms which people start with.

Building blocks of a Linear Regression Model are:

* Discrete/Continuous independent variable
* A best-fit regression line
* Continuous dependent variable. I.e., A Linear Regression model predicts the dependent variable using a regression line based on the independent variables. The equation of the Linear Regression is:

**Y = MX + C**

Chart, diagram, line chart

Description automatically generated

Were,

* Y = Dependent variable
* M = Slope
* X = Independent variable
* C = Constant or Intercept

**Exercise**

**Problem Statement:**

This data is about the amount spent on advertising through different channels like TV, Radio, and newspapers. The goal is to predict how the expense on each channel affects the sales and is there a way to optimize the sales?

Importing the necessary packages

Graphical user interface, text, application

Description automatically generated

Loading and exploring the data

Graphical user interface, text

Description automatically generated

Table

Description automatically generated

What are the **features**?

1. TV: Dollars spent on TV ads for a single product in each market (in thousands of dollars).
2. Radio: Dollars spent on Radio ads.
3. Newspaper: Dollars spent on Newspaper ads.

What is the **response**?

* Sales: sales of a single product in each market (in thousands of widgets)

Dimensions of the data

Graphical user interface, text

Description automatically generated

Find the missing values from different columns

Text

Description automatically generated

Let’s showcase the relationship between the feature and target variables

Text

Description automatically generated with medium confidence

Chart, scatter chart

Description automatically generated

From the relationship diagrams above, it can be observed that there seems to be a linear relationship between the features TV ad, Radio ad, and the sales is almost a linear one. A linear relationship typically looks like this:

Shape

Description automatically generated

Hence, we can build a model using the Linear Regression Algorithm.



The mathematical equation is:

***𝑦* =*𝛽*0 + *𝛽*1*𝑥***

What do terms represent?

* Y: Response or Target Variable
* X: Feature Variable
* 𝛽1: Coefficient of X
* 𝛽0: Intercept

𝛽0 and 𝛽1 are the **model coefficients**. To create a model, we must "learn" the values of these coefficients. And once we have the value of these coefficients, we can use the model to predict the Sales!



### Ordinary Least Squared Estimation

Chart, line chart

Description automatically generated

* In each two-dimensional space, an infinite number of lines can be plotted through the scatter diagram between two variables.
* Calculate the distance between the observed values and predicted values, which are called Errors or Residuals.
* Errors can be positive and negative, therefore square the error term (squared Error).
* Repeat the process with all the infinite number of lines and identify the line with a minimum sum of squared error.
* Hence the name “Ordinary least Squared”.

### The mathematics involved

Take a quick look at the plot created. Now consider each point and know that each of them has a coordinate in the form (X, Y). Now draw an imaginary line between each point and the current "best-fit" line. We'll call the distance between each point and the current best-fit line as D. To get a quick image of what we're trying to visualize, look at the picture below:

Chart, line chart

Description automatically generated

What elements are present in the diagram?

* The red points are the observed values of X and Y.
* The blue line is the least square line.
* The green lines are the residuals, which is the distance between the observed values and the least squared line.

Before, we're labeling each green line as having a distance D, and each red point as having a coordinate of (X, Y). Then we can define our best fit line as the lines having the property were:

𝐷21+𝐷22+𝐷23+𝐷24+....+𝐷2𝑁

So how do we find this line? The least-square line approximating the set of points:

**(𝑋,𝑌)1,(𝑋,𝑌)2,(𝑋,𝑌)3,(𝑋,𝑌)4,(𝑋,𝑌)5,**

Has the equation:

***𝑦* =*𝛽*0 + *𝛽*1*𝑥***

This is basically just a rewritten form of the standard equation for a line:

**Y = MX + C**

### Derivation of OLS by Minimizing Errors

Minimize the sum of squared error term by substituting as below

Text

Description automatically generated with medium confidence

Solving the above equation by calculus – partial differencing by ***𝛽*0 and *𝛽*1** respectively and solving for the two variables we get the below equations

Text, letter

Description automatically generated

Building Simple Linear Regression Model to predict the sales based on TV ads

Graphical user interface, text, application, email

Description automatically generated

### Interpreting the model

How do we interpret the coefficient for spends on TV ads (β1)?

* A “unit” increase in spends on a TV ad is **associated with** a 0.04753 “unit” increase in sales.
* Or, an additional $1,000 on TV ads is translated to an increase in sales by $47.53.

### Prediction using the model

If the expense on a TV ad is $50000, what will be the sales prediction for that market?

***𝑦* =*𝛽*0 + *𝛽*1*𝑥***

Y = 7.032594 + 0.047537 \* (50)

Graphical user interface, text, application

Description automatically generated

Thus, we would predict Sales of 9,409 widgets in that market.

Let’s do the same thing using code.

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

Plotting the Least Squares Line

Text

Description automatically generated

A picture containing application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

Chart, scatter chart

Description automatically generated

### Model Confidence

**Question:** Is linear regression a low bias/high variance model or a high bias/low variance model?

**Answer:** It's a high bias/low variance model. Even after repeated sampling, the best fit line will stay roughly in the same position (low variance), but the average of the models created after repeated sampling won't do a great job in capturing the perfect relationship (high bias). Low variance is helpful when we don't have less training data!

If the model has calculated a 95% confidence for our model coefficients, it can be interpreted as follows: If the population, from which this sample is drawn, is **sampled 100 times**, then approximately **95 (out of 100) of those confidence intervals** shall contain the "true" coefficients.

Text

Description automatically generated

Table

Description automatically generated with low confidence

### Hypothesis Testing and p-values

**Hypothesis testing** is closely related to confidence intervals. We start with a **null hypothesis** and an **alternate hypothesis** (that is opposite to the null). Then, we check whether the data **reject the null hypothesis** or **fails to reject the null hypothesis**.

The conventional hypothesis test is as follows:

* **Null hypothesis:** No relationship exists between TV advertisements and Sales (and hence *𝛽*1 equals zero).
* **Alternative hypothesis:** There exists a relationship between TV advertisements and Sales (and hence, *𝛽*1 is not equal to zero).

How do we test this? We reject the null hypothesis (and thus believe the alternative hypothesis) if the 95% confidence interval **does not include zero**. The **p-value** represents the probability of the coefficient actually being zero.



Chart

Description automatically generated with medium confidence

If the 95% confidence interval **includes zero**, the p-value for that coefficient will be **greater than 0.05**. If the 95% confidence interval **does not include zero**, the p-value will be **less than 0.05**.

Thus, a p-value of less than 0.05 is a way to decide whether there is any relationship between the feature in consideration and the response or not. Using 0.05 as the cutoff is just a convention.

In this case, the p-value for TV ads is way less than 0.05, and so we **believe** that there is a relationship between TV advertisements and Sales.

Note that we generally ignore the p-value for the intercept.



Till now, we have studied models based on only one feature. Now, we’ll include multiple features and create a model to see the relationship between those features and the target column. This is called **Multiple Linear Regression**.

**A picture containing text, antenna

Description automatically generated**

**Estimation of model parameters**

Consider the model where

A picture containing text

Description automatically generated

The columns of X are each covariate for the n patients, with the first column being all 1’s to include the intercept in the model.

Based on this model we get the following expansion for the first subject:



Then using matrix calculus, we find that the least square estimate for β is given by



Hence, the least-squares regression line is



The beta values are obtained by calculating below equation

A picture containing text, watch, clock

Description automatically generated

Note:

1. (XTX)-1 should be a non-singular matrix, otherwise, we cannot calculate beta
2. Beta cannot be calculated if columns are linear combinations of others.

We know that

Text, letter

Description automatically generated

H is called a Hat matrix

Properties of Hat matrix:

1. H is symmetric; HT = H
2. H is idempotent; HH = H

**Assumptions of OLS Regression**

Regression is a parametric approach. ‘Parametric’ means it makes assumptions about data for the purpose of analysis. Due to its parametric side, regression is restrictive in nature. It fails to deliver good results with data sets that don’t fulfill its assumptions. Therefore, for successful regression analysis, it’s essential to validate these assumptions.

So, how would you check (validate) if a data set follows all regression assumptions? You check it using the regression plots (explained below) along with some statistical test.

Let’s look at the important assumptions in regression analysis:

1. There should be a linear and additive relationship between the dependent (response) variable and the independent (predictor) variable(s). A linear relationship suggests that a change in response Y due to one unit change in X1 is constant, regardless of the value of X1. An additive relationship suggests that the effect of X1 on Y is independent of other variables.
2. There should be no correlation between the residual (error) terms. The absence of this phenomenon is known as Autocorrelation.
3. The independent variables should not be correlated. The absence of this phenomenon is known as multicollinearity.
4. The error terms must have constant variance. This phenomenon is known as homoscedasticity. The presence of non-constant variance is referred to as heteroskedasticity.
5. The error terms must be normally distributed.

**What if these assumptions get violated?**

Let’s dive into specific assumptions and learn about their outcomes (if violated):

1. **Linear and Additive:**  If you fit a linear model to a non-linear, non-additive dataset, the regression algorithm would fail to capture the trend mathematically, thus resulting in an inefficient model. Also, this will result in erroneous predictions on an unseen dataset.

**How to check:** Look for residual vs fitted value plots (explained below).

1. **Autocorrelation:** The presence of correlation in error terms drastically reduces a model's accuracy. This usually occurs in time series models where the next instant is dependent on the previous instant. If the error terms are correlated, the estimated standard errors tend to underestimate the true standard error.

If this happens, it causes confidence intervals and prediction intervals to be narrower. A narrower confidence interval means that a 95% confidence interval would have a lesser probability than 0.95 that it would contain the actual value of coefficients. Let’s understand narrow prediction intervals with an example:

For example, the least square coefficient of X1 is 15.02 and its standard error is 2.08 (without autocorrelation). But in presence of autocorrelation, the standard error reduces to 1.20. As a result, the prediction interval narrows down to (13.82, 16.22) from (12.94, 17.10).

Also, lower standard errors would cause the associated p-values to be lower than actual. This will make us incorrectly conclude a parameter to be statistically significant.

**How to check:** Look for Durbin – Watson (DW) statistics. It must lie between 0 and 4. If DW = 2, implies no autocorrelation, 0 < DW < 2 implies positive autocorrelation while 2 < DW < 4 indicates negative autocorrelation. Also, you can see residual vs time plots and look for the seasonal or correlated pattern in residual values.

1. **Multicollinearity:** This phenomenon exists when the independent variables are found to be moderately or highly correlated. In a model with correlated variables, it becomes a tough task to figure out the true relationship of predictors with response variables. In other words, it becomes difficult to find out which variable is contributing to predict the response variable.

Another point, with the presence of correlated predictors, the standard errors tend to increase. And, with large standard errors, the confidence interval becomes wider leading to less precise estimates of slope parameters.

Also, when predictors are correlated, the estimated regression coefficient of a correlated variable depends on which other predictors are available in the model. If this happens, you’ll end up with an incorrect conclusion that a variable strongly / weakly affects the target variable. Since, even if you drop one correlated variable from the model, its estimated regression coefficients would change. That’s not good!

**How to check:** You can use a scatter plot to visualize the correlation effect among variables. Also, you can also use the VIF factor. VIF value <= 4 suggests no multicollinearity whereas a value of >= 10 implies serious multicollinearity. Above all, a correlation table should also solve the purpose.

1. **Heteroscedasticity:** The presence of non-constant variance in the error terms results in heteroscedasticity. Generally, non-constant variance arises in presence of outliers or extreme leverage values. Look like, these values get too much weight, thereby disproportionately influences the model’s performance. When this phenomenon occurs, the confidence interval for out-of-sample prediction tends to be unrealistically wide or narrow.

**How to check**: You can look at the residual vs fitted values plot. If heteroscedasticity exists, the plot will exhibit a funnel shape pattern. Also, you can use Breusch-Pagan / Cook – Weisberg test or the White general test to detect this phenomenon.

1. **Normal Distribution of error terms:** If the error terms are non-normally distributed, confidence intervals may become too wide or narrow. Once confidence interval becomes unstable, it leads to difficulty in estimating coefficients based on minimization of least squares. The presence of non – normal distribution suggests that there are a few unusual data points that must be studied closely to make a better model.

**How to check:** You can look at the QQ plot (shown below). You can also perform statistical tests of normality such as the Kolmogorov-Smirnov test, Shapiro-Wilk test.

## Interpretation of Regression Plots

Until here, we’ve learned about the important regression assumptions and the methods to undertake, if those assumptions get violated.

But that’s not the end. Now, you should know the solutions also to tackle the violation of these assumptions. In this section, I’ve explained the 4 regression plots along with the methods to overcome limitations on assumptions.

## Residual vs Fitted Values

## Diagram Description automatically generated

This scatter plot shows the distribution of residuals (errors) vs fitted values (predicted values). It is one of the most important plots which everyone must learn. It reveals various useful insights including outliers. The outliers in this plot are labeled by their observation numbers which make them easy to detect.

There are two major things which you should learn:

* If there exists any pattern (maybe, a parabolic shape) in this plot, consider it as signs of non-linearity in the data. It means that the model doesn’t capture non-linear effects.
* If a funnel shape is evident in the plot, consider it as the signs of non-constant variance i.e. heteroscedasticity

**Solution:** To overcome the issue of non-linearity, you can do a nonlinear transformation of predictors such as log (X), √X, or X² transform the dependent variable. To overcome heteroscedasticity, a possible way is to transform the response variable such as log(Y) or √Y. Also, you can use a weighted least square method to tackle heteroscedasticity.

## Normal Q-Q Plot

## Chart, line chart Description automatically generated

This q-q or quantile-quantile is a scatter plot that helps us validate the assumption of normal distribution in a data set. Using this plot we can infer if the data comes from a normal distribution. If yes, the plot would show a fairly straight line. The absence of normality in the errors can be seen with deviation in the straight line.

If you are wondering what is ‘quantile’, here’s a simple definition: Think of quantiles as points in your data below which a certain proportion of data falls. Quantile is often referred to as percentiles. For example: when we say the value of the 50th percentile is 120, it means half of the data lies below 120.

**Solution:** If the errors are not normally distributed, the nonlinear transformation of the variables (response or predictors) can bring improvement in the model.

## Scale Location Plot

## Chart, scatter chart Description automatically generated

This plot is also used to detect homoscedasticity (assumption of equal variance). It shows how the residuals are spread along with the range of predictors. It’s similar to the residuals vs fitted value plot except it uses standardized residual values. Ideally, there should be no discernible pattern in the plot. This would imply that errors are normally distributed. However, if the plot shows any discernible pattern (probably a funnel shape), it would imply non-normal distribution of errors.

Solution: Follow the solution for heteroskedasticity given in plot 1.

## Residuals vs Leverage Plot

## Chart, scatter chart Description automatically generated

It is also known as Cook’s Distance plot. Cook’s distance attempts to identify the points which have more influence than other points. Such influential points tend to have a sizable impact on the regression line. In other words, adding or removing such points from the model can completely change the model statistics.

But, can these influential observations be treated as outliers? This question can only be answered after looking at the data. Therefore, in this plot, the large values marked by the cook’s distance might require further investigation.

**Solution:** For influential observations which are nothing but outliers, if not many, you can remove those rows. Alternatively, you can scale down the outlier observation with maximum value in data or else treat those values as missing values.

## Model Evaluation Metrics

Model evaluation is very important in data analytics. It helps you to understand the performance of your model and makes it easy to present your model to other people. There are many different evaluation metrics out there but only some of them are suitable to be used for regression.

There are 5 main metrics for model evaluation in regression:

1. R Square
2. Adjusted R Square
3. Mean Square Error (MSE)
4. Root Mean Squared Error (RMSE)
5. Mean Absolute Error
6. **R Square/Adjusted R Square**

R Square measures how much variability in the dependent variables can be explained by the model. It is a square of Correlation Coefficient (R) and that is why it is called R Square. It takes the form of a proportion --- the proportion of variance explained --- and so it always takes on a value between 0 and 1.

For example, **R2** statistic = 0.75, says that our model fits 75% of the total dataset. Similarly, if it is 0, it means none of the data points is being explained and a value of 1 represents 100% data explanation. Mathematically **R2** statistic is calculated as:

Text

Description automatically generated

Where RSS: is the Residual Sum of Squares and is given as:

A picture containing text, watch

Description automatically generated

RSS is the residual (error) term we have been talking about so far. And. TSS: is the Total Sum of Squared and given as:



TSS is calculated when we consider the line passing through the mean value y. to be the best fit line. Just like RSS, we calculate the error term when the best fit line is the line passing through the mean value of y and we get the value of TSS.

A picture containing timeline

Description automatically generated

The closer the value of R2 is to 1 the better the model fits our data. If R2 comes below (which is the possibility) that means the model is so bad that it is performing even worse than the average best fit.

1. **Adjusted R Square**

R Square is a good measure to determine how well the model fits the dependent variables. However, it does not take into consideration the Overfitting problem. If your regression model has many independent variables, because the model is too complicated, it may fit very well to the training data but performs badly for testing data. That is why Adjusted R Square is introduced because it will penalize additional independent variables added to the model and adjust the metric to prevent Overfitting issues.

Mathematically it is calculated as:

A picture containing diagram

Description automatically generated

Where:

* R2: Sample R Square
* N: Total Sample Size
* P: Number of Independent variables

1. **Mean Square Error (MSE)**

While R Square is a relative measure of how well the model fits dependent variables, Mean Square Error is an absolute measure of the goodness of fit.

MSE is calculated by the sum of prediction error which is real output minus predicted output and then divided by the number of data points. It gives you an absolute number on how much you’re predicted results deviate from the actual number. You cannot interpret much insight from one single result but it gives you a real number to compare against other model results and help you to select the best regression model.

Mathematically it is calculated as:

A picture containing text, clock, gauge

Description automatically generated

Where:

* Yi: is an actual value
* Yi hat: is predicted value
* N: No. of observations

1. **Root Mean Square Error (RMSE)**

Root Mean Square Error (RMSE) is the square root of MSE. It is used more commonly than MSE because sometimes MSE values can be too big to compare easily. Secondly, MSE is calculated by the square of error, and thus square root brings to the same level of prediction error and makes it easier for interpretation.

Mathematically is calculated as:

Text

Description automatically generated

Where:

* Yi: is an actual value
* Yi hat: is predicted value
* N: No. of observations

1. **Mean Absolute Error**

Mean Absolute Error (MAE) is like Mean Square Error (MSE). However, instead of the sum of square error in MSE, MAE is taking the Sum of Absolute value of error.

Compared to MSE, RMSE, and MAE, MAE is a more direct representation of the sum of error terms. MSE gives larger penalization to big prediction error by square it while MAE treats all errors the same.

Mathematically it is calculated as:

A picture containing text, clock, gauge

Description automatically generated

Where:

* Yi: is an actual value
* Yi hat: is predicted value
* N: No. of observations

**Overall Recommendation/Conclusion**

R Square/Adjusted R Square is better used to explain the model to other people because you can explain the number as a percentage of the output variability. MSE, RMSE, or MAE is better to be used to compare performance between different regression models.

**Problem Statement**

A Chinese automobile company Geely Auto aspires to enter the US market by setting up their manufacturing unit there and producing cars locally to give competition to their US and European counterparts.

They have contracted an automobile consulting company to understand the factor on which the price of cars depends. Specifically, they want to understand the factors affecting the pricing of cars in the American market, since those may be very different from the Chinese market. The company wants to know:

Which variables are significant in predicting the price of a car? How well those variables describe the price of a car Based on various market surveys, the consulting firm has gathered a large dataset of different types of cars across the American market.

**Business Goal**

We are required to model the price of cars with the available independent variables. It will be used by the management to understand how exactly the prices vary with the independent variables. They can accordingly manipulate the design of the cars, the business strategy, etc. to meet certain price levels. Further, the model will be a good way for management to understand the pricing dynamics of a new market.

Let’s start our analysis by importing the required libraries and data

Text

Description automatically generated

A picture containing text

Description automatically generated

Graphical user interface, text

Description automatically generated

Let’s see some basic data exploration

Shape

Description automatically generated with medium confidence

Table

Description automatically generated

Overall data types seem ok.

Let’s check for any duplicate records in the dataset.

Graphical user interface, application

Description automatically generated

There are no duplicate records.

Now, let’s check for missing values in the columns

Text

Description automatically generated with medium confidence

Table

Description automatically generated

Table

Description automatically generated

There are no missing values and duplicates values, this made our job easy.

Let’s go for unique values check.

Table

Description automatically generated

We can see the highest number of missing values are in car\_ID column followed by CarName in categorical features.

* There is no Zero variance variable
* Car\_ID column is the repetition of the index. So I’ll drop it.
* Carname has 147 different entities. I’ll check it. And try to find a way to reduce the variance.
* Other than there is no problem

Let’s make a copy of the dataset and start to work on it.

Text

Description automatically generated

Before moving forward, I’ll handle the ‘CarName’

Text, letter

Description automatically generated

Text

Description automatically generated

We use only the brands/make not the models.

A picture containing text

Description automatically generated

Table

Description automatically generated

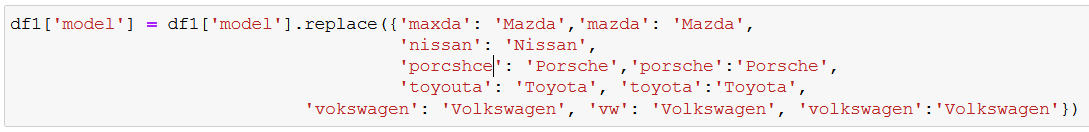
Let’s drop the ‘CarName’ column

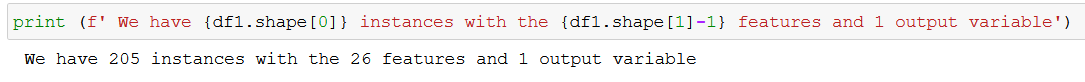


Text, letter

Description automatically generated

There seems to be some spelling error in the ‘model’ column.





Let’s create numeric and categorical data frames separately.

Text

Description automatically generated

Text

Description automatically generated

Let’s explore the target variable ‘price’

Text

Description automatically generated

Looking at the mean and median, the data seems skewed. Let’s confirm it.

Graphical user interface

Description automatically generated

Text

Description automatically generated

Chart, box and whisker chart

Description automatically generated

Yes! Our assumption is true. Even though the target variable has the right skewness, we will not make any transformation on it.

* The plot seemed to be right-skewed, meaning that the most prices in the dataset are low (Below 15,000).
* There is a significant difference between the mean and the median of the price distribution.
* The data points are far spread out from the mean, which indicates a high variance in the car prices. (85% of the prices are below 18500, whereas the remaining 15% are between 18,500 and 45,400.)

**Visualizing Categorical Data**

Text

Description automatically generated

Chart, bar chart, histogram

Description automatically generated

Chart, bar chart, histogram

Description automatically generated

Chart, bar chart, histogram

Description automatically generated

**Inference:**

* Toyota seemed to be the favored car company.
* A number of ‘gas’ fuelled cars are more than diesel.
* The sedan is the top car type preferred.

Graphical user interface, text, application

Description automatically generated

Chart, box and whisker chart

Description automatically generated

Inference:

* It seems that the symbolling with 0 and 1 values have a high number of rows (i.e., They are most sold)
* The cars with -1 symbolling seem to be high priced (as it makes sense too, insurance risk rating -1 is quite good). But it seems that symbolling with 3 values has a price range like -2 value. There is a dip in price at symbolling 1.

Graphical user interface, text, application

Description automatically generated

Chart, bar chart

Description automatically generated

Chart, box and whisker chart

Description automatically generated

Chart, bar chart

Description automatically generated

Inference:

* ohc Engine type seems to be the most favored type.
* ohcv has the highest price range (While dohcv has only one row), ohc and ohcf have the low-price range.

Text

Description automatically generated

Chart, bar chart, histogram

Description automatically generated

Chart, bar chart

Description automatically generated

Chart, bar chart

Description automatically generated

Inference:

* Jaguar and Buick seem to have the highest average prices.
* Diesel has a higher average price than gas.
* Hardtop and convertible have higher average prices.

Chart, box and whisker chart

Description automatically generated

Chart, box and whisker chart

Description automatically generated

Inference:

* doornumber variable is not affecting the price much. There is no significant difference between the categories in it.
* It seems aspiration with turbo has a higher price range than the std (though it has some high values outside the whiskers.)

Text

Description automatically generated

Chart, box and whisker chart

Description automatically generated

Chart, box and whisker chart

Description automatically generated

Chart, box and whisker chart

Description automatically generated

Chart, box and whisker chart

Description automatically generated

Inference:

* Very few data points for ‘enginelocation’ categories to make an inference.
* The most common number of cylinders are four, six, and five. Though eight cylinders have the highest price range.
* Mpfi and 2bb1 are the most common types of fuel systems. Mpfi and ‘idi’ having the highest price range. But there are little data for other categories to derive any meaningful inference.
* A very significant difference in the drivewheel category. Most high ranged cars seem to prefer rwd drivewheel.

**Visualizing numerical data**

Text

Description automatically generated

Chart, scatter chart

Description automatically generated

Inference:

* Carwidth, carlength, and carweight seems to have a positive correlation with price.
* Carheight doesn’t show any significant trend with price.

Text

Description automatically generated

Chart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generated

Inference:

* Enginesize, boreratio, horsepower, wheelbase – seems to have a significant positive correlation with price.
* Citympg, highwaympg – seem to have a significant negative correlation with price

Graphical user interface, text, application, Word

Description automatically generated

**Feature Engineering**

Deriving new features

A picture containing text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

**Bivariate Analysis**

Graphical user interface, text, application, Word

Description automatically generated

Chart, scatter chart

Description automatically generated

Inference:

* fueleconomy has an obvious negative correlation with price and is significant

Text

Description automatically generated

Chart, bar chart

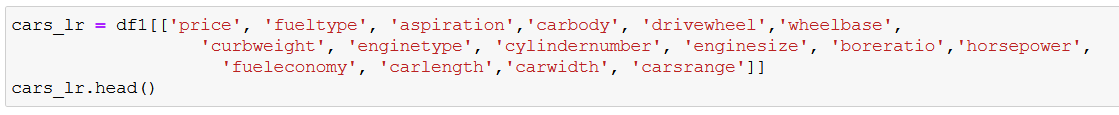
Description automatically generated

Inference:

* High-ranged cars prefer rwd drivewheel with idi or mpfi fuelsystem.

List of significant variables after Visual analysis:

* Car Range
* Engine Type
* Fuel Type
* Car Body
* Aspiration
* Cylinder Number
* Curb weight
* Car Length
* Car width
* Engine Size
* Bore ratio
* Horsepower
* Wheelbase
* Fuel Economy



Graphical user interface

Description automatically generated

Text

Description automatically generated with medium confidence

Text, table, whiteboard

Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

**Train-Test Split and Feature scaling**

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

A computer screen capture

Description automatically generated with low confidence

Highly correlated variables to price are – curbweight, engineseze, horsepower, carwidth, and highend.

**The train and test split**

Text

Description automatically generated

**Model Building**

We will be using Recursive Feature Elimination or RFE for short is a feature selection algorithm.

Feature selection refers to techniques that select a subset of the most relevant features (columns) for a dataset. Fever features can allow machine learning algorithms to run more efficiently (less space or time complexity) and be more effective. Some machine learning algorithms can be misled by irrelevant input features, resulting in worse predictive performance.

RFE is a wrapper-type feature selection algorithm. This means that a different machine learning algorithm is given and used in the core of the method, is wrapped by RFE, and used to help select features, this contrasts with filter-based feature selections that score each feature and select those features with the largest score. Technically, RFE is a wrapper-style feature selection algorithm that also uses filter-based feature selection internally.

RFE works by searching for a subset of features by starting with all features in the training dataset and successfully removing features until the desired number remains.

This is achieved by fitting the given machine learning algorithm used in the core of the model, ranking features by importance, discarding the least important features, and re-fitting mode. This process is repeated until a specified number of features remains.

Features are scored either using the provided machine learning model or by using the statistical method.

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

Table

Description automatically generated

Graphical user interface, text

Description automatically generated

**Model 1**



Table

Description automatically generated

p-value of ‘twelve’ seems to be higher than the significance value of 0.05, hence dropping it as it is insignificant in presence of other variables.



**Model 2**

Text

Description automatically generated with low confidence

Table

Description automatically generated



**Model 3**

Text

Description automatically generated

Table

Description automatically generated

Calculating the variance inflation factor



Graphical user interface

Description automatically generated with medium confidence

Dropping curbweight because of high VIF value. (Shows that curbweight has high multicollinearity)

**Model 4**



Table

Description automatically generated



Graphical user interface

Description automatically generated with medium confidence

Dropping sedan because of high VIF value.

**Model 5**



Table

Description automatically generated

A picture containing diagram

Description automatically generated

Graphical user interface, table

Description automatically generated with medium confidence

Dropping wagon because of the high p-value.

**Model 6**



Table

Description automatically generated

Graphical user interface, application, table

Description automatically generated

**Model 7**

Text

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

**Residual Analysis of Model**

Diagram

Description automatically generated with medium confidence

Text

Description automatically generated

Chart, histogram

Description automatically generated

Error terms seem to be approximately normally distributed, so the assumption on the linear modeling seems to be fulfilled.

**Prediction and Evaluation**

Graphical user interface, text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**Model Evaluation**

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

Chart, scatter chart

Description automatically generated

**Evaluation of the model using statistics**



Table

Description automatically generated

**Inference:**

* R-squared and Adjusted R-squared (extent of fit): 0.899 and 0.896 i.e., 90% of variance explained.
* F-stats and Prob(F-stats) (overall model fit): 308.0 and 1.04e-67 (approximately 0) – Model is significant and explained 90% variance is just not by chance.
* P-value: p-value for all the coefficients seem to be less than the significance level of 0.05. meaning that all the predictors are statistically significant.

**Logistic Regression**

**Introduction**

In linear regression, the type of data we deal with is quantitative, whereas we use classification models to deal with qualitative data or categorical data. The algorithms used for solving a classification problem first predict the probability of each of the categorical of qualitative variables, as the basis for making the classification. And, as the probabilities are continuous numbers, classification using probabilities also behave like regression methods. Logistic regression is one such type of classification model which is used to classify the dependent variable into two or more classes or categories.

Why don’t we use linear regression for classification problems?

Let’s suppose you took a survey and noted the response of each person as satisfied, neutral, or not satisfied. Let’s map each category:

Satisfied – 2

Neutral – 1

Not Satisfied – 0

But this doesn’t mean that the gap between Not satisfied and Neutral is the same as Neutral and satisfied. There is no mathematical significance of this mapping. We can also map the categories like:

Satisfied – 0

Neutral – 1

Not Satisfied – 2

It’s completely fine to choose the above mapping. If we apply linear regression to both the type of mappings, we will get different sets of predictions. Also, we can get prediction values like 1.2, 0.8, 2.3, etc. which makes no sense for categorical values. So, there is no normal method to convert qualitative data into quantitative data for use in linear regression. Although, for binary classification, i.e., when there are only two categorical values, using the least square method can give decent results. Suppose we have two categories Black and White, and we map them as follows:

Black – 0

White - 1



Logistic regression is one such regression algorithm that can be used for performing classification problems. It calculates the probability that a given value belongs to a specific class. If the probability is more than 50%, it assigns the value in that particular class else if the probability is less than 50%, the value is assigned to the other class. Therefore, we can say that logistic regression acts as a binary classifier.

**Working of a Logistic Model**

For linear regression, the model is defined by: 𝑦=𝛽0+𝛽1𝑥 – (i)

and for logistic regression, we calculate probability, i.e., y is the probability of a given variable x belonging to a certain class. Thus, it is obvious that the value of y should lie between 0 and 1.

But, when we use equation(i) to calculate probability, we would get values less than 0 as well as greater than 1. That doesn’t make any sense. So, we need to use such an equation which always gives values between 0 and 1, as we desire while calculating the probability.

**Sigmoid Function**

We use the sigmoid function as the underlying function in Logistic regression. Mathematically and graphically, it is shown as:

Chart

Description automatically generated

Why do we use the Sigmoid Function?

1. The sigmoid function’s range is bounded between 0 and 1. Thus it’s useful in calculating the probability for the logistic function.
2. Its derivative is easy to calculate than other functions which are useful during gradient descent calculation.
3. It is a simple way of introducing non-linearity to the model.

Although there are other functions as well, which can be used, sigmoid is the most common function used for logistic regression.

The logistic function is given as:

Diagram

Description automatically generated with medium confidence

Let’s see some manipulation with the logistic function:

Text, letter

Description automatically generated

We can see that the logit function is linear in terms of x.

Chart, diagram, line chart

Description automatically generated

The y value based on the threshold of 0.5:

* If p(x) < 0.5 then y = 0
* If p(x) >= 0.5 then y = 1

**Cost Function**

The cost function for the whole training set is given as:

Text, whiteboard

Description automatically generated

The values of parameters (θ) for which the cost function is minimum is calculated using the gradient descent algorithm. The partial derivative for cost function is given as:

Text, whiteboard

Description automatically generated

**Multiple Logistic Function**

We can generalize the simple logistic function for multiple features as:

A picture containing text

Description automatically generated

And the logit function can be written as:

Logo

Description automatically generated with low confidence

The coefficients are calculated the same we did for a simple logistic function, bypassing the above equation in the cost function.

Just like we did in multiple linear regression, we will check for correlation between features for Multi Logistic as well.

We will see how we implement all the above concepts through a practical example.

**Logistic Regression Use Case**

In this case study, we will build a Logistic Regression model in Python to predict whether an employee will be promoted or not.

For everyone, promotion or appraisal cycles are the most exciting times of the year. Final promotions are only disclosed after employees have been evaluated on a variety of criteria, which causes a delay in transitioning to new responsibilities. We will build a machine learning model to predict who is qualified for promotion to speed up the process.

**Importing Libraries**

We’ll begin by loading the necessary libraries for creating a Logistic regression model.

Text

Description automatically generated

**Loading the data**

Text

Description automatically generated with low confidence

Graphical user interface

Description automatically generated

**Basic Data Exploration**

Text

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with medium confidence

We have total of 54808 observations and 14 columns.

Graphical user interface

Description automatically generated with medium confidence

Graphical user interface

Description automatically generated

“is\_promoted” is our Target Variable, which has two categories encoded as 1 (promoted) and 0 (not promoted) rest all are input features. In addition, we can observe that our dataset contains both numerical and categorical features.

**Data Cleaning**

Data cleaning is a crucial stage in the data pre-processing process. We’ll remove columns with only one unique value because their variance will be 0 and they won’t help us anticipate anything.

Let’s see whether there are any columns that only have one unique value.

Text

Description automatically generated

This signifies that there isn’t any column having only 1 unique value.

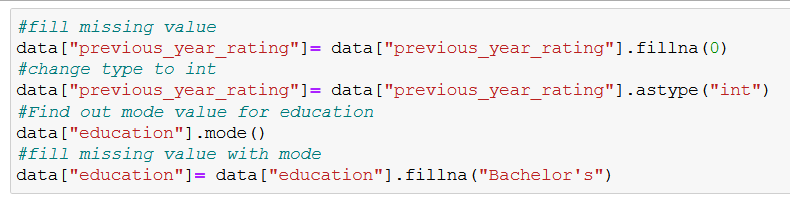
We’ll now drop the employee\_id column because it’s merely a unique identifier, and then verify each field in the dataset for null value percentage.

Table

Description automatically generated

‘previous\_year\_rating’ and ‘education’ both features have null values. As a result, we will impute those null values instead of dropping them. Following our examination of those columns, we discovered that –

* For rows with null ‘previous\_year\_rating’, we can see that their length of service is 1, which could be why they don’t have a previous year rating. As a result, we’ll use 0 to impute null values.
* For the education column, we will impute null values with mode.



Now, we do not have any null or missing values in our data.

So, let’s proceed to our next step. There are no null or missing values in our data now. So, let’s go on to the next step.

**Exploratory Data Analysis**

Getting insights from data and visualizing them is an important stage in machine learning since it provides us with a better view of features and their relationships.

Let’s look at the target variable’s distribution in the dataset.

Graphical user interface, text, application

Description automatically generated

Chart, pie chart

Description automatically generated

We can observe from the above charts that, promoted employee data is less than non-promoted employee data, indicating that there is a class imbalance because class 0 has more data points or observations than class 1.

Let’s visualize if there is any relationship between the target variable and other variables.

Text

Description automatically generated

Chart, box and whisker chart

Description automatically generated

For an employee if the “avg\_training\_score value is higher then the chances of getting promoted are more.

We will plot correlations between different variables using a heatmap.

Text

Description automatically generated

A picture containing Teams

Description automatically generated

None of the features is highly correlated with each other except age and length of service.

**Feature Engineering**

In feature engineering, we apply domain expertise to produce new features from raw data, or we convert or encode features. We’ll encode categorical features or make dummy features out of them in this section.

Text

Description automatically generated

**Train-Test Split**

We will divide the dataset into two subsets: train and test. To perform the train-test split, we’ll use Scikit-learn machine learning.

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

After splitting the dataset, we have 43846 observations in the training subset and 10962 in the test subset.

**Feature Scaling/Normalization**

Let’s see how to feature scaling in python using Scikit-learn.

Text

Description automatically generated

**Class Imbalance**

Graphical user interface, text, application

Description automatically generated

We can observe that most of the labels are from class 0 and only a few are from class 1.

If we use this distribution to develop our model, it may become biased towards predicting the majority class since there will be insufficient data to learn minority class patterns. The model will start predicting every new observation as 0 or majority class. (In our problem employee is not promoted). We’ll get more accuracy here, but it won’t be a decent model because it won’t predict class 1 or minority class, with is a crucial class.

**How to handle class imbalance?**

There are a variety of approaches to dealing with class imbalance, such as increasing minority class samples or decreasing majority class samples to ensure that both classes have the same distribution.

Because we’re using the Scikit-learn machine library to create the model, it has a logistic regression implementation that supports class weighting. We will use the inbuild parameter “class\_weight” while creating an instance of the logistic regression model.

Both the majority and minority classes will be given separate weights. During the training phase, the weights differences will influence the classification of the classes.

The purpose of adding class weights is to penalize the minority class for misclassification by setting a higher-class weight while decreasing the weight for the majority class.

Text

Description automatically generated

After training our model on the training dataset, we used our model to predict values for the test dataset and recorded them in the y\_pred\_basemodel variable.

**Model Evaluation Metrics**

Graphical user interface

Description automatically generated with low confidence

We got a 38% f1 score on our base model created using default parameters.

Up to this point, we saw how to create a logistic regression model using default parameters.

Now, let’s increase model performance and evaluate it again after tuning the hyperparameters of the model.

**Hyperparameter Optimization**

Hyperparameters controls the overfitting or underfitting of the model. Hyperparameter tuning can be done using algorithms like Grid Search or Random Search.

A picture containing text

Description automatically generated

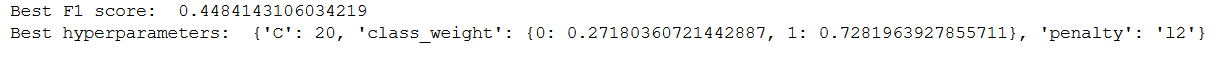
Text

Description automatically generated

After fitting the model, we will extract the best fit values for all specified hyperparameters.

Graphical user interface, text, application

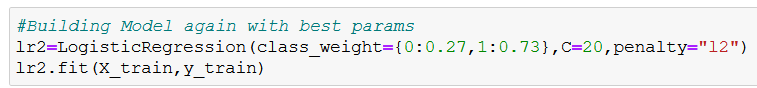
Description automatically generated



We will now build our logistic regression model using the above values we got by tuning hyperparameters.

Build Model using optimal values of Hyperparameters

Let’s use the below code to build our model again.





After training our final model it’s time to evaluate our logistic regression model using chosen metrics.

**Model Evaluation**

Text

Description automatically generated

Text, letter

Description automatically generated

We can see that by tuning hyperparameters, we were able to improve the performance of our model since the F1 Score for the final model (0.43) is higher than that of the base model (0.38). After the hyperparameter tuning model got a 0.69 ROC-AUC score.

With this, we were able to construct our logistic regression model and test it on the test dataset. More feature engineering, hyperparameter optimization, and cross-validation techniques can improve its performance even more.

**SUMMARY**

We have started with Bias-Variance Trade-off, then, we have walked through a complete end-to-end regression and logistic regression use cases. We imported the data, performed data exploration, data cleaning, feature engineering, built a model, and finally did the hyperparameters tuning and fitted model using the best parameters.

**Program Assignment**

Build a Real Estate prediction engine to predict the cost of real estate based on housing market features.

Follow the following steps:

1. Data importing
2. Data exploring
3. Data Cleaning
4. Feature Engineering
5. Feature Selection
6. Model Building
7. Hyperparameter Tuning
8. Building the final model with the best parameters

**Assessment**

**Choose the appropriate option**

1. Which of the following steps/assumption in regression modeling impacts the trade-off between under-fitting and over-fitting
   1. The polynomial degree
   2. Whether we learn the weights by matrix inversion or gradient descent
   3. The use of a constant term
   4. None of the above
2. Let’s say, a “Linear regression” model perfectly fits the training data (train error is zero). Now, which of the following statements is true?
   1. You will always have test error zero
   2. You can not have test error zero
   3. None of the above
   4. All of the above
3. In a linear regression problem, we are using “R-squared” to measure goodness-of-fit. We add a feature in a linear regression model and retrain the same model. Which of the following option is true?
   1. If R-Squared increases, this variable is significant
   2. If R-Squared decreases, this variable is not significant
   3. Individually R Squared cannot tell the above variable importance. We can’t say anything about it right now.
   4. None of these.
4. Which one of the statements is true regarding residuals in regression analysis?
   1. The mean of residuals is always less than zero.
   2. The mean of residuals is always greater than zero.
   3. There is no such rule for residuals
   4. The mean of residuals is always zero
5. Which of the following indicates a fairly strong relationship between X and Y?
   1. Correlation coefficient = 0.9
   2. The p-value for the null hypothesis Beta coefficient = 0 is 0.0001
   3. The t-statistic for the null hypothesis Beta coefficient = 0 is 30
   4. None of the above

**Fill in the spaces with appropriate answers**

1. To test linear relationship of y (dependent) and x (independent) continuous variable, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ plot best suites?
2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ algorithm is used to predict categorical dependent variable?
3. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ method we use to best fit the data in Logistic Regression?
4. The out of the sigmoid function like between \_\_\_\_\_\_\_\_\_\_\_\_.
5. How mang coefficients do you need to estimate in a simple linear regression model \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**True or False**

1. Is Logistic regression mainly used for Regression?
   1. True
   2. False
2. Is it possible to design a logistic regression algorithm using a Neural Network Algorithm?
   1. True
   2. False
3. Is it possible to apply a logistic regression algorithm on a 3-class classification problem?
   1. Ture
   2. False
4. Standardization of features is required before training a Logistic Regression.
   1. True
   2. False
5. Correlated variables can have zero correlation coefficient. True or False?
   1. True
   2. False

**Assessment Solutions**

**Choose the appropriate option**

1. A
2. C
3. C
4. D
5. A

**Fill in the spaces with appropriate answers**

1. Scatter plot
2. Logistic Regression
3. Maximum Likelihood
4. 0 and 1
5. 1

**True or False**

1. False
2. True
3. True
4. False
5. True