

LOGISTIC REGRESSION

Q.1 What is Regression?

Ans: - Regression analysis consists of a set of machine learning methods that allow us to predict a continuous outcome variable (y) based on the value of one or multiple predictor variables (x). Briefly, the goal of regression model is to build a mathematical equation that defines y as a function of the x variables.

Regression is a statistical method used in finance, investing, and other disciplines that attempts to determine the strength and character of the relationship between one dependent variable (Y) and a series of other independent variables(X).

Q.2 What is Classification?

Ans: - In machine learning, classification refers to a predictive modelling problem where a class label is predicted for a given example of input data.

Examples of classification problems include: Given an data, classify if it will rain or not. Given a data, classify it as new or not .

Q.3 What are the types of Regression?

Ans: - There are seven types of regression: -

1. Linear Regression
2. Logistic Regression
3. Polynomial Regression
4. Stepwise Regression
5. Ridge Regression
6. Lasso Regression
7. ElasticNet Regression

Q.3 What is Logistic Regression?

Ans: - Logistic Regression is one of the basic and popular algorithms to solve a classification problem.

It is mainly used in situations where there is a binary classification needed.

Q.4 Why is logistic regression called regression if it does the job of classification?

Ans: - It is called 'Logistic Regression' because its underlying technique is quite the same as Linear Regression. The term "Logistic" is taken from the logit function that is used in this method of classification.

Q.5 What is the similarity between linear regression and logistic regression?

Ans: - With linear regression you're looking for the k_i parameters:

$$h = k_0 + \sum k_i \cdot X_i = K_t \cdot X$$

With logistic regression you've the same aim but the equation is:

$$h = g(K_t \cdot X)$$

Where g is the sigmoid function:

$$g(w) = 1 / (1 + e^{-w})$$

So:

$$h = 1 / (1 + e^{-K_t \cdot X})$$

and you need to fit K to your data.

Assuming a binary classification problem, the output h is the estimated probability that the example x is a positive match in the classification task:

$$P(Y = 1) = 1 / (1 + e^{-K_t \cdot X})$$

When the probability is greater than 0.5 then we can predict "a match".

The probability is greater than 0.5 when:

$$g(w) > 0.5$$

and this is true when:

$$w = K_t \cdot X \geq 0$$

The hyperplane:

$$Kt \cdot X = 0$$

is the decision boundary.

Logistic regression is a generalized linear model using the same basic formula of linear regression but it is regressing for the probability of a categorical outcome.

Q.6 Explain the mechanism of Logistic Regression

Ans: -

- Unlike actual regression, logistic regression does not try to predict the value of a numeric variable given a set of inputs.
- Instead, the output is a probability that the given input point belongs to a certain class.

For simplicity, let's assume that we have only two classes, and the probability in Q is

- P_{+} -> the probability that a certain data point belongs to the '+' class.
- $P_{-} = 1 - P_{+}$.

Thus, the output of Logistic Regression always lies in $[0, 1]$.

Q.7 What are the real-life examples of Logistic Regression?

Ans: -

We have data on 1000 random customers from a given city. We want to know what determines their

decision to subscribe to a magazine.

Subscribe: Indicates if a customer has subscribed to the magazine

- Age: Examine how age influences the likelihood of the subscription
- Other attributes: ...

Q.8 What are the problems of linear approach?

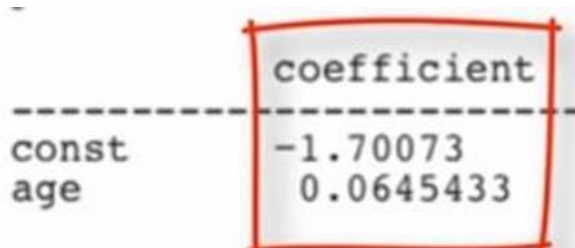
Ans: -

Besides the outcome being binary, there is nothing special about the DV (y, subscribe)

- If a customer subscribes, the value of y is higher (from 0 to 1)
- We can apply the linear regression: -

$$y(\text{subscribe}) = \beta_0 + \beta_1 \text{Age} + \varepsilon$$

$$y(\text{subscribe}) = -1.700 + 0.064 * \text{Age}$$



	coefficient
const	-1.70073
age	0.0645433

- If the DV is binary then the focus should be to see what makes it change from y=0 to y=1

- This is also explained as the likelihood of subscription or p (subscribe = 1)

$$y(\text{subscribe}) = -1.700 + 0.064 * \text{Age}$$

$$P(\text{subscribe} = 1) = p = -1.700 + 0.064 * \text{Age}$$

- Every additional year of Age, increases the probability of subscription by 6.4%
- Probabilities are bounded,) $0 \leq p \leq 1$
- The range of age in the data is $20 \leq \text{age} \leq 55$
- The probability that a 35 year old person subscribes is

$$P = -1.700 + 0.064 * 35 = 0.54$$

- The probability that a person 25 years or 45 years subscribes?

$$P = -1.700 + 0.064 * 25 = -0.09 \text{ ... possible?}$$

$$P = -1.700 + 0.064 * 45 = +1.20$$

Q.9 What is probability plot?

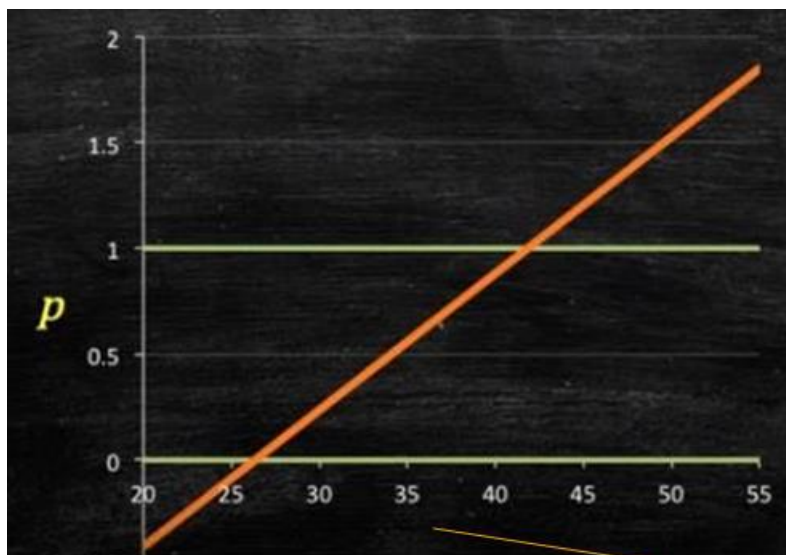
Ans:- The probability plot is a graphical technique for assessing whether or not a data set follows a given distribution such as the normal .

The data are plotted against a theoretical distribution in such a way that the points should form approximately a straight line. Departures from this straight line indicate departures from the specified distribution.

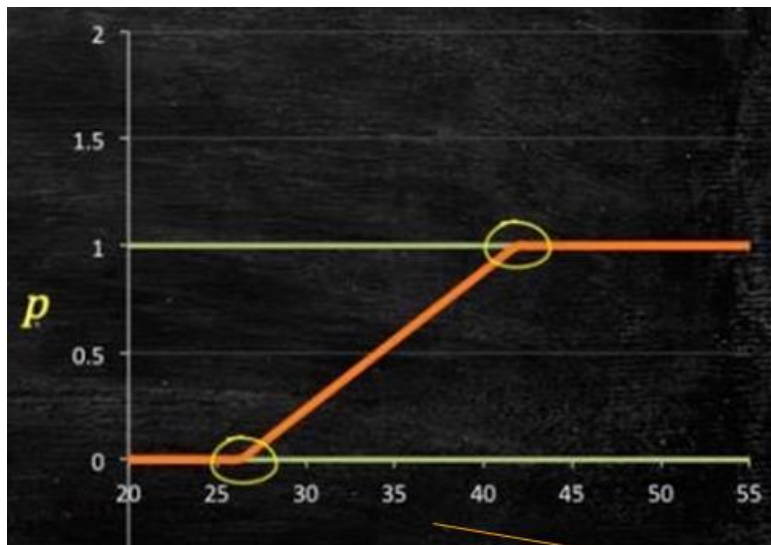
Q.10 How does Logistic Regression plot transforms the probability plot?

Ans:-

1. Customers of more than 45 years of age have probability > 1
2. Customer who are less than 25 years age, the probability is less than 0



1. Any probability > 1.0, can be made 1.0
2. Any probability < 0.0, can be made 0.0



Q.11 What are between Logistic Regression and Linear Regression?

Ans:-

Linear Regression	Logistic Regression
In linear regression, the outcome (dependent variable) is continuous.	<ul style="list-style-type: none"> • Binary classification; • is used when the response variable is categorical in nature. E.g. yes/no, true/false, red/green
The data is modelled using a straight line.	The probability of some obtained event is represented as a linear function of a combination of predictor variables.
Linear relationship between dependent and independent variables is required	Linear relationship between dependent and independent variables is NOT required
In the linear regression, the independent variable can be correlated with each other.	The variable must not be correlated with each other

Q.12 What is a Sigmoid Function?

Ans:-

In order to map predicted values to probabilities, we use the sigmoid function. The function maps any real value into another value between 0 and 1. In machine learning, we use sigmoid to map predictions to probabilities.

Formula:- $S(z) = 1 / (1 + e^{-z})$

Code:-

def sigmoid(z):

return 1.0 / (1 + np.exp(-z))

Handwritten derivation of the Sigmoid function:

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 \text{Age} \rightarrow (1)$$

Now, our function gives values $(-\infty \text{ to } \infty)$

Odds ratio = $\left(\frac{p}{1-p}\right)$

Probability	$\ln\left(\frac{p}{1-p}\right)$	$\ln\left(\frac{p}{1-p}\right)$
0	$\frac{0}{1-0} = 0$	$\ln(0) \rightarrow -\infty$
1	$\frac{1}{1-1} = \infty$	$\ln(\infty) \rightarrow \infty$

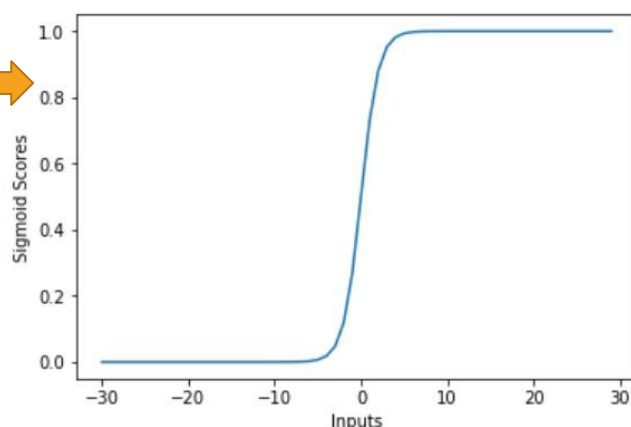
Since, $\ln\left(\frac{p}{1-p}\right) = z$ (function) from (1)

$$e^z = \frac{p}{1-p}$$

$$\Rightarrow e^z(1-p) = p \Rightarrow e^z = p(1+e^z)$$

$$\Rightarrow p = \frac{e^z}{1+e^z} \Rightarrow \frac{1}{\frac{1}{e^z} + 1} = \left(\frac{1}{1+e^{-z}}\right)$$

(Sigmoid function)



Q .13 What is the difference between Sigmoid function and SoftMax function?

Ans:-

SoftMax Function	Sigmoid Function
Used for multi-classification in logistic regression model.	Used for binary classification in logistic regression model.
The probabilities sum will be 1	The probabilities sum need not be 1.
Used in the different layers of neural networks.	Used as activation function while building neural networks.
The high value will have the higher probability than other values	The high value will have the high probability but not the higher probability.

Q.14 In a nutshell, Explain the advantages and disadvantages of Logistic Regression?

Ans:-

Advantages:-

1. Highly interpretable, Outputs well-calibrated predicted probabilities.
2. Model training and prediction are fast.
3. Can perform well with a small number of observations.

Disadvantages:-

1. Presumes a linear relationship between the features and the log-odds of the response.
2. Is it not possible to apply a logistic regression algorithm on a larger Classification problem?

Q.15 What are the assumptions of Logistic Regression?

Ans:-

- The Response Variable should be Binary in nature.
- The Observations are Independent
- There is No Multicollinearity Among Explanatory Variables.
- There are No Extreme Outliers
- There is a Linear Relationship Between Explanatory Variables and the Logit of the Response Variable
- The Sample Size is Sufficiently Large.

Q.16 What are the assumptions that in contrast to linear regression, logistic regression does not require?

Ans:-

- A linear relationship between the explanatory variable(s) and the response variable.
- The residuals of the model to be normally distributed.

- The residuals to have constant variance, also known as homoscedasticity.

Q.17 Why does the response variable in the data should be binary in nature when using the logistic regression algorithm?

Ans:- Logistic Regression can work on multivariate variables but it will not give a precise accuracy since works and classifies on the sigmoid function and that sigmoid function is used for only binary nature variables. Hence Logistic Regression is preferred in the data which is binary in nature.

Q.18 What are the applications of Logistic Regression?

Ans: -

Logistic regression is used in various fields, including machine learning, most medical fields, and social sciences. For example, it is used to predict mortality in injured patients, to predict the risk of developing a given disease. based on observed characteristics of the patient (age, sex, body mass index, results of various blood tests, etc.).

Another example might be to predict whether a voter will vote which party. The technique can also be used in engineering, especially for predicting the probability of failure of a given process, system or product. It can also be used in weather forecasting, i.e. whether it will rain or not based on the weather condition variables.

Other example will be of the credit card issue. The Credit Card Fraud Detection problem is of significant importance to the banking industry because banks each year spend hundreds of millions of dollars due to fraud. When a credit card transaction happens, the bank makes a note of several factors. For instance, the date of the transaction, amount, place, type of purchase, etc. Based on these factors, they develop a Logistic Regression model of whether or not the transaction is a fraud. For instance, if the amount is too high and the bank knows that the concerned person never makes purchases that high, they may label it as a fraud.

The logistic regression can also be used in marketing domain. Every day, when you browse your Facebook newsfeed, the powerful algorithms running behind the scene predict whether or not you would be interested in certain content (which could be, for instance, an advertisement). Such algorithms can be viewed as complex variations of Logistic Regression algorithms where the Q to be answered is simple – will the user like this particular advertisement in his/her news feed?

Another example will be in the medical domain. A Logistic Regression classifier may be used to identify whether a tumour is malignant or if it is benign. Several medical imaging techniques are used to extract various features of tumours. For instance, the size of the tumour, the affected body area, etc. These features are then fed to a Logistic Regression classifier to identify if the tumour is malignant or if it is benign.

Q.19 What is the formula for sigmoid Function?

Ans : -

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}}$$

Q.20 What is Hypothesis test?

Ans:-

In logistic regression, hypotheses are of interest:

Null hypothesis :- Null hypothesis which is when all the coefficients in the regression equation take the value zero.

Alternate hypothesis :- Alternate hypothesis is that the model currently under consideration is accurate and differs significantly from the null of zero, i.e. gives significantly better than the chance or random prediction level of the null hypothesis.

Q 21. What is the purpose of Logistic Regression?

Ans:-

- 1) The logistic regression predicts group membership
 - 2) Since logistic regression calculates the probability of success over the probability of failure, the results of the analysis are in the form of an odds ratio. Logistic regression determines the impact of multiple independent variables presented simultaneously to predict membership of one or other of the two dependent variable categories.
 - 3) The logistic regression also provides the relationships and strengths among the variables
- Assumptions of (Binary) Logistic Regression.**
- Logistic regression does not assume a linear relationship between the dependent and independent variables. Logistic regression assumes linearity of independent variables and log odds of dependent variable. The independent variables need not be interval, nor normally distributed, nor linearly related, nor of equal variance within each group. Homoscedasticity is not required. The error terms (residuals) do not need to be normally distributed. The dependent variable in logistic regression is not measured on an interval or ratio scale.

Q.22 What is Log Transformation?

Ans: -

The log transformation is, arguably, the most popular among the different types of transformations used to transform skewed data to approximately conform to normality. Log transformations and sq. root transformations moved skewed distributions closer to normality. So what we are about to do is common. This log transformation of the p values to a log distribution enables us to create a link with the normal regression equation. The log distribution (or logistic transformation of p) is also called the logit of p or logit(p).

In logistic regression, a logistic transformation of the odds (referred to as logit) serves as the dependent variable:

$$\text{Log(odds)} = \text{logit}(P) = \ln(P/1-P)$$

If we take the above dependent variable and add a regression equation for the independent variables, we get a logistic regression:

$$\text{logit}(p) = a + b_1x_1 + b_2x_2 + b_3x_3 + \dots$$

As in least-squares regression, the relationship between the logit(P) and X is assumed to be linear.

Q.23 What is the general workflow of Logistic Regression Algorithm?

Ans:-

The general workflow is:

- 1) get a dataset
- 2) train a classifier
- 3) make a prediction using such classifier

Q.24 What are the libraries required for implementing Logistic Regression?

Ans:-

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
import warnings
warnings.filterwarnings("ignore")
pd.set_option("display.max_columns",None)
```

Note :- I will be working on an Employee Dataset in which we have to predict whether the salary of the employee will be higher than 50k or less than 50k.

Q.25 How to import the dataset into the python Environment?

Ans:-

Code:-

```
adult_df = pd.read_csv('adult_data.csv',header = None, delimiter=' ', *')
```

```
adult_df.head()
```

OP:-

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K

Q.26 The dataset does not has headers, how to define the headers on the dataset?

Ans: -

This can be done with columns functions:-

```
adult_df.columns = ['age', 'workclass', 'fnlwgt', 'education',  
'education_num','marital_status','occupation', 'relationship','race', 'sex',  
'capital_gain', 'capital_loss',  
'hours_per_week', 'native_country', 'income']  
adult_df.head()
```

Q.27 How can you revert the original data frame in case any failure in your data analysis?

Ans: -

#CREATE A COPY OF THE DATAFRAME

```
adult_df_rev=pd.DataFrame.copy(adult_df)
```

Q.28 How to drop some of the variables in the dataset?

Ans:-

Code:-

```
adult_df_rev = adult_df_rev.drop(['education','fnlwgt'], axis=1)
```

Q.29 How to have look at the dataset?

Ans: - We can have a look at the dataset using head() and tail() functions

Code:- `adult_df_rev.head()`

OP:-

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	emp_length	home_ownership
0	1077501	1296599	5000.0	5000.0	4975.0	36 months	10.65	162.87	B	B2	NaN	10+ years	RENT
1	1077430	1314167	2500.0	2500.0	2500.0	60 months	15.27	59.83	C	C4	Ryder	< 1 year	RENT
2	1077175	1313524	2400.0	2400.0	2400.0	36 months	15.96	84.33	C	C5	NaN	10+ years	RENT
3	1076863	1277178	10000.0	10000.0	10000.0	36 months	13.49	339.31	C	C1	AIR RESOURCES BOARD	10+ years	RENT
4	1075358	1311748	3000.0	3000.0	3000.0	60 months	12.69	67.79	B	B5	University Medical Group	1 year	RENT

Q.30 How to check the null values in the dataset?

Ans:- The null values in the dataset can be checked by using isnull() function.

Code:-

```
df_rev.isnull().sum()
```

OP:-

loan_amnt	0
funded_amnt	0
funded_amnt_inv	0
term	0
int_rate	0
installment	0
grade	0
sub_grade	0
emp_length	43061
home_ownership	0
annual_inc	0
verification_status	0
issue_d	0
purpose	0
zip_code	0
dti	0
delinq_2yrs	0
earliest_cr_line	0
inq_last_6mths	0
open_acc	0
pub_rec	0
revol_bal	0
revol_util	446
total_acc	0
initial_list_status	0
out_prncp	0
out_prncp_inv	0
total_pymnt	0
total_pymnt_inv	0
total_rec_prncp	0
total_rec_int	0
total_rec_late_fee	0
recoveries	0
collection_recovery_fee	0
last_pymnt_d	8862
last_pymnt_amnt	0
next_pymnt_d	252971

Q.31 How to impute the missing values?

Ans:-

1. By using the mean

Code:-

```
colname2=['revol_util','collections_12_mths_ex_med',
          'total_rev_hi_lim']
```

```
for x in colname2[:]:
```

```
    data[x].fillna(data[x].mean(),inplace=True)
```

```
data.isnull().sum()
```

data.shape

2.By using info from other variables

```
emp_avg_income = data.groupby('emp_length').annual_inc.agg('mean')
```

emp_avg_income

```
emp_length
1 year      70905.942739
10+ years   82152.634924
2 years     72577.282604
3 years     73437.968649
4 years     73806.577465
5 years     74378.125078
6 years     74309.575394
7 years     74690.965159
8 years     76023.805407
9 years     75746.361255
< 1 year    70475.918204
Name: annual_inc, dtype: float64
```

```
def impute_emp_length(cols):
```

```
    emp_length = cols[0]
```

```
    annual_inc = cols[1]
```

```
    if pd.isnull(emp_length):
```

```
        if annual_inc < 70800:
```

```
            return '< 1 year'
```

```
        elif annual_inc in range(70801,72000):
```

```
            return '1 year'
```

```
        elif annual_inc in range(72000,72800):
```

```
            return '2 years'
```

```
        elif annual_inc in range(72800,73600):
```

```
            return '3 years'
```

```
        elif annual_inc in range(73600,74000):
```

```
            return '4 years'
```



```

elif annual_inc in range(74000,74500):
    return '5 years'
elif annual_inc in range(74500,74600):
    return '6 years'
elif annual_inc in range(74600,74700):
    return '7 years'
elif annual_inc in range(74700,74800):
    return '8 years'
elif annual_inc in range(74800,75900):
    return '9 years'
else:
    return '10+ years'
else:
    return emp_length

```

```

data['emp_length'] =
data[['emp_length','annual_inc']].apply(impute_emp_length, axis=1)

```

OP:-

loan_amnt	0
funded_amnt	0
funded_amnt_inv	0
term	0
int_rate	0
installment	0
grade	0
sub_grade	0
emp_length	0
home_ownership	0
annual_inc	0
verification_status	0
issue_d	0
purpose	0
zip_code	0
dti	0
delinq_2yrs	0
earliest_cr_line	0
inq_last_6mths	0
open_acc	0
pub_rec	0
revol_bal	0

Q.32 How to know the count of occurrences of the variables?

Ans: -

Code: - `df.workclass.value_counts()`

Q.33 How to check for Outliers ?

Ans:- Outliers can be detected by the following ways: -

1. Extreme Value Analysis by Box Plot
2. Visualizing the data

Q.33 How to convert the categorical data into numerical data?

Ans: - The different ways by Which you can convert the categorical variables into numerical ones are: -

1. Label Encoder
2. Manually Mapping
3. Dummy Variables
4. One hot label Encoding

Q.34 On what basis should we decide that outliers should be eliminated or not ??

Ans:- If the quantity of outlier is less then we should eliminate them since the logistic regression doesn't allow outliers and if the quantity of outliers is more then we should keep them as it is and let the algorithm handle it.

Q.34 How to check for outliers in the data?

Ans: -

- 1.By Using the Boxplot range

Code:-

```
df.boxplot(column='age')  
plt.show()
```

2.Create a for loop that will calculate the IQR

```
#for value in colname:
```

```
q1 = df['age'].quantile(0.25)    #first quartile value
```

```
q3 = df['age'].quantile(0.75)    #third quartile value
```

```
iqr = q3-q1 #Interquartile range
```

```
low  = q1-1.5*iqr #acceptable range
```

```
high = q3+1.5*iqr #acceptable range
```

```
df_include = df.loc[(df['age'] >= low) & (df['age'] <= high)]
```

```
df_exclude = df.loc[(df['age'] < low) | (df['age'] > high)]
```

3.Finding the mean of the acceptable range.

```
age_mean=int(df_include.age.mean())
```

```
print(age_mean)
```

```
df_exclude.age=age_mean
```

4.Getting back the original shape of the dataframe.

```
df_rev=pd.concat([df_include,df_exclude]) #concatenating both dfs to  
get
```

```
#the original shape
```

```
df_rev.shape
```

5.The capping approach

```
df_exclude.loc[df_exclude["age"] < low, "age"] = low
df_exclude.loc[df_exclude["age"] > high, "age"] = high
```

Q.34 Explain Label Encoder

Ans:- One hot encoding is used to encode the categorical column. It replaces a categorical column with its labels and fills values either 0 or 1. For example, you can see the “color” column, there are 3 categories such as red, yellow, and green. 3 categories labeled with binary values.

Code:-

```
colname1=['grade','term','sub_grade','emp_length','home_ownership','verification_status',
          'purpose','zip_code','earliest_cr_line','last_pymnt_d',
          'next_pymnt_d','last_credit_pull_d','application_type','initial_list_status']
```

```
data.head()
from sklearn import preprocessing
le={}
for x in colname1:
    le[x]=preprocessing.LabelEncoder()
for x in colname1:
    data[x]=le[x].fit_transform(data[x])
data.head()
```

OP:-

Before transforming

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	emp_length
0	1077501	1296599	5000.0	5000.0	4975.0	36 months	10.65	162.87	B	B2	NaN	10+ years
1	1077430	1314167	2500.0	2500.0	2500.0	60 months	15.27	59.83	C	C4	Ryder	< 1 year
2	1077175	1313524	2400.0	2400.0	2400.0	36 months	15.96	84.33	C	C5	NaN	10+ years
3	1076863	1277178	10000.0	10000.0	10000.0	36 months	13.49	339.31	C	C1	AIR RESOURCES BOARD	10+ years
4	1075358	1311748	3000.0	3000.0	3000.0	60 months	12.69	67.79	B	B5	University Medical Group	1 year

After Transforming

	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_length	home_ownership	annual_inc	verification_status
0	5000.0	5000.0	4975.0	0	10.65	162.87	1	6	1	5	24000.0	2
1	2500.0	2500.0	2500.0	1	15.27	59.83	2	13	10	5	30000.0	1
2	2400.0	2400.0	2400.0	0	15.96	84.33	2	14	1	5	12252.0	0
3	10000.0	10000.0	10000.0	0	13.49	339.31	2	10	1	5	49200.0	1
4	3000.0	3000.0	3000.0	1	12.69	67.79	1	9	0	5	80000.0	1

Q.35 Explain Manual Mapping

Ans: - Manual mapping is a technique where we individually take one by one element and assign them a value. This is done where you need to convert specific values and the number of these values is less.

Syntax:-

Example : -

```
df["column_name"]=df.column_name.map({Desired Value : Actual Value })
```

Q.36 What are Dummy Variables?

Ans:- A Dummy variable is an artificial variable created to represent an attribute with two or more distinct categories/levels. Its requires less computational power compared to other techniques. However the coding length is more compared to other techniques.

Syntax:-

```
import pandas as pd
```

```
raw_data = {'first_name': ['Saurabh', 'Amit', 'Mansi', 'Pranjali', 'Ankita'],  
            'last_name': ['Parab', 'Parab', 'Rane', 'Gawde', 'Lokande'],  
            'sex': ['male', 'male', 'female', 'female', 'female']}
```

```
df = pd.DataFrame(raw_data, columns = ['first_name', 'last_name', 'sex'])
```

df

	first_name	last_name	sex
0	Saurabh	Parab	male
1	Amit	Parab	male
2	Mansi	Rane	female
3	Pranjali	Gawde	female
4	Ankita	Lokande	female

```
pd.get_dummies(df, columns=['sex'])
```

	first_name	last_name	sex_female	sex_male
0	Saurabh	Parab	0	1
1	Amit	Parab	0	1
2	Mansi	Rane	1	0
3	Pranjali	Gawde	1	0
4	Ankita	Lokande	1	0

Q.37 Explain One Hot Label Encoding

Ans:- It is a process that converts categorical data to integers or a vector of ones and zeros. The length of vector is determined by number of expected classes or categories. Each element in the vector represents a class. Therefore, a one is used to indicate which class it is and everything else will be zero.

Code:-

```
from sklearn.preprocessing import OneHotEncoder  
type_one_hot = OneHotEncoder(sparse=False).fit_transform(  
train_new.array.to_numpy().reshape(-1,1))
```

If we have categorical data that we think may be important, we want to be able to use this in the model. This is because regression algorithms and classification algorithms won't be able to process it. This is when one-hot encoding is useful.

Note: - So we will use the most convenient way to transform the categorical variables

Q.38 Explain the steps of Label Encoding?

Ans: -

1. Create a user defined function which will loop through the entire dataset and will return all the categorical variables into the list.

Code:-

```
colname1=['grade','term','sub_grade','emp_length','home_ownership','verification_status',  
          'purpose','zip_code','earliest_cr_line','last_pymnt_d',  
          'next_pymnt_d','last_credit_pull_d','application_type','initial_list_status']
```

```

data.head()

from sklearn import preprocessing

le={}

2. Use the Label Encoder Function.

for x in colname1:

    le[x]=preprocessing.LabelEncoder()

for x in colname1:

    data[x]=le[x].fit_transform(data[x])

data.head()

```

OP:-

	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_length	home_ownership	annual_inc	verification_status
0	5000.0	5000.0	4975.0	0	10.65	162.87	1	6	1	5	24000.0	2
1	2500.0	2500.0	2500.0	1	15.27	59.83	2	13	10	5	30000.0	1
2	2400.0	2400.0	2400.0	0	15.96	84.33	2	14	1	5	12252.0	0
3	10000.0	10000.0	10000.0	0	13.49	339.31	2	10	1	5	49200.0	1
4	3000.0	3000.0	3000.0	1	12.69	67.79	1	9	0	5	80000.0	1

Q.39 What is Scaling?

Ans:- Feature Scaling is a technique to standardize the independent features present in the data in a fixed range. ... If feature scaling is not done, then a machine learning algorithm tends to weigh greater values, higher and consider smaller values as the lower values, regardless of the unit of the values.

Machine learning is like making a mixed fruit juice. If we want to get the best-mixed juice, we need to mix all fruit not by their size but based on their right proportion. We just need to remember apple and strawberry are not the same unless we make them similar in some context to compare

their attribute. Similarly, in many machine learning algorithms, to bring all features in the same standing, we need to do scaling so that one significant.

The two major techniques for Feature Scaling are:

- Normalization is used when we want to bound our values between two numbers, typically, between [0,1] or [-1,1].
- Standardization transforms the data to have zero mean and a variance of 1, they make our data unitless.

Q.40 Explain the syntax of Scaling and which scaling technique we will be using in this algorithm?

Ans:- We will be using the standard scaler in this algorithm. Standard Scaler assumes a normal distribution for data within each feature. The scaling makes the distribution centred around 0, with a standard deviation of 1 and the mean removed.

Formula:-

$$x(i) - \text{mean}(x)$$

$$\text{Sd}(x)$$

Where sd is the standard deviation of x.

Syntax:-

```
from sklearn.preprocessing import StandardScaler  
scaler = StandardScaler()  
scaler.fit(X)  
X=scaler.transform(X)  
print(X)
```

Q.41 How will we decide how to train the model and how to test the model on the data which is available to us?

Ans:- In Machine Learning, we split the data into 2 parts, training and testing parts.

We train the model on training data and compare its results with the test data.

Q.42 What is the threshold for splitting the data?

Ans: - Usually we follow the threshold of 70:30 of the data i.e., 70 % of the data to the training and 30% of the data. It depends on the situation whether you need more data for your model if it's not giving you the accuracy.

Syntax: -

```
from sklearn.model_selection import train_test_split
```

```
X_ , X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3,  
random_state=101)
```

Q.43 What do you mean by feature splitting?

Ans: - A feature splitting is a technique to generate a few other features from the existing one to improve the model performance. For example, splitting names into first and last names.

For Example :- Deriving the profit variable from selling price and cost price variable .

Q.44 What do you mean by feature selection?

Ans:- Feature selection means the process of selecting the independent variables and the dependant variables for your model.

Syntax:-

```
Independent variables alias name = dataframe_name.values[column  
names]
```

```
dependent variable alias name = dataframe_name.values[column name]
```

Code:-

```
X=adult_df_rev.values[:,0:-1]
```

```
Y=adult_df_rev.values[:,-1]
```

Q.42 What is loc and iloc in python and what is the difference between them?

Ans:- The main distinction between the two methods is:-

- loc gets rows (and/or columns) with particular labels.
- iloc gets rows (and/or columns) at integer locations.

For Example:-

ILOC:-

```
X=adult_df_rev.values[:,0:-1]
```

```
Y=adult_df_rev.values[:,-1]
```

LOC:-

```
X= adult_df_rev.values['age','post',gender',etc]
```

```
Y=adult_df_rev.values['income']
```

Q.43 What is the code for building the Logistic Regression Model?

Ans:-

```
from sklearn.linear_model import LogisticRegression
```

```
#create a model
```

```
classifier=LogisticRegression()
```

```
#build train the model
```

```
classifier.fit(X_train,Y_train)
```

```
#predict using the model you created
```

```

Y_pred=classifier.predict(X_test)
#we are using this for comparison
#print(list(zip(Y_test,Y_pred)))

print(classifier.coef_)
print(classifier.intercept_)

```

Q.44 Can we create a custom function of Confusion Matrix so that we can picturized it beautifully?

Ans:-

```

import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix
from sklearn.utils.multiclass import unique_labels
import itertools

def plot_confusion_matrix(cm, classes,
                           normalize=False,
                           title='Confusion Matrix',
                           cmap=plt.cm.Greens):
    """this function prints and plot the confusion matrix
    Normalization can be applied by setting 'normalize=True'
    """
    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized Confusion Matrix")
    else:
        print("Confusion Matrix")

```

```

print(cm)

plt.imshow(cm, interpolation='nearest', cmap=cmap)
plt.title(title)
plt.colorbar()
tick_marks = np.arange(len(classes))
plt.xticks(tick_marks, classes, rotation=35)
plt.yticks(tick_marks, classes)

fmt = '.2f' if normalize else 'd'
thresh = cm.max() / 2.

for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[0])):
    plt.text(j, i, format(cm[i,j], fmt),
             horizontalalignment='center',
             color='white' if cm[i, j] > thresh else 'black')

plt.ylabel('True label')
plt.xlabel('Predicted label')
plt.tight_layout()

```

Q.44 How to check whether our model is performing well or not?

Ans:- Sklearn provides various functions for this purpose like accuracy ,confusion Matrix,classification report etc,

Code:-

```

from sklearn.metrics import confusion_matrix, accuracy_score,
classification_report

cfm = confusion_matrix(Y_test,Y_pred)

print(cfm)

print("CLASSIFICATION MATRIX:")

print(classification_report(Y_test,Y_pred))

acc = accuracy_score(Y_test,Y_pred)

print("ACCURACY OF THE MODEL:",acc)

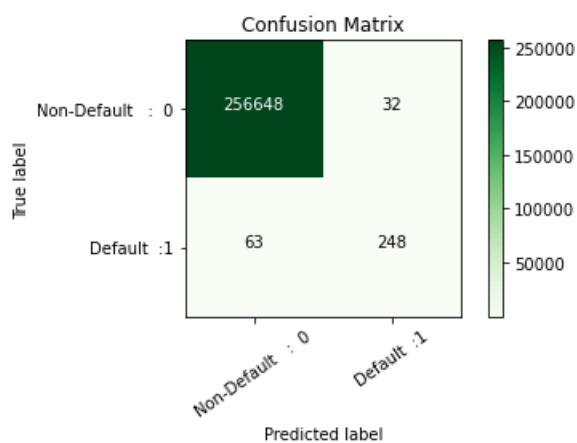
```

OP:-

```

Confusion Matrix
[[256648    32]
 [    63   248]]

```



```

Classification report
precision    recall  f1-score   support

     0       1.00      1.00      1.00    256680
     1       0.89      0.80      0.84      311

   accuracy          0.94      0.90      0.92    256991
  macro avg          0.94      0.90      0.92    256991
weighted avg          1.00      1.00      1.00    256991

```

Accuracy of the model: 0.9996303372491644

Q.45 What is Accuracy of model?

Ans:-

Accuracy is the quintessential classification metric. It is pretty easy to understand. And easily suited for binary as well as a multiclass classification problem.

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{FP} + \text{FN} + \text{TN})$$

Accuracy is the proportion of true results among the total number of cases examined.

When to use?

Accuracy is a valid choice of evaluation for classification problems which are well balanced and not skewed or No class imbalance.

So here we have an accuracy of 82 which is pretty good.

Q.45 What is Precision of model?

Ans:-

Precision means what proportion of predicted Positives is truly Positive?

$$\text{Precision} = (\text{TP}) / (\text{TP} + \text{FP})$$

In the asteroid prediction problem, we never predicted a true positive.

And thus precision=0

When to use?

Precision is a valid choice of evaluation metric when we want to be very sure of our prediction. For example: If we are building a system to predict if we should decrease the credit limit on a particular account, we want to be very sure about our prediction or it may result in customer dissatisfaction.

Q.46 What is Recall Factor ?

Ans:- Recall Factor means proportion of actual Positives is correctly classified?

$$\text{Recall} = (\text{TP}) / (\text{TP} + \text{FN})$$

In the asteroid prediction problem, we never predicted a true positive.

And thus recall is also equal to 0.

When to use?

Recall is a valid choice of evaluation metric when we want to capture as many positives as possible. For example: If we are building a system to predict if a person has cancer or not, we want to capture the disease even if we are not very sure.

Here we have an recall factor of 0.95 for 0 and 0.45 for 1 meaning our algorithm is performing well for 0 and not 1

Q.47 Does Logistic Regression provide any feature in these condition where the recall factor is not satisfying?

Ans:- Logistic Regression provides the proba function in such cases

Proba function adjusts the threshold and the capps the values so that the recall factor improves resulting in better accuracy.

```
#Store the predicted probabilitiles
```

```
y_pred_prob=classifier.predict_proba(X_test)
```

```
print(y_pred_prob)
```

```
y_pred_class=[]
```

```
for value in y_pred_prob[:,1]:
```

```
    if value > 0.46:
```

```
        y_pred_class.append(1)
```

```
    else:
```

```
        y_pred_class.append(0)
```

```
print(y_pred_class)
```

```
TYPE 1 ERROR TYPE 2 ERROR
```

```
for a in np.arange(0.4,0.61,0.01):
```

```
    predict_mine = np.where(y_pred_prob[:,1] > a, 1, 0)
```

```
    cfm=confusion_matrix(Y_test, predict_mine)
```



```
total_err=cfm[0,1]+cfm[1,0]
print("Errors at threshold ", a, ":",total_err, " , type 2 error :",
      cfm[1,0]," , type 1 error:", cfm[0,1])
```

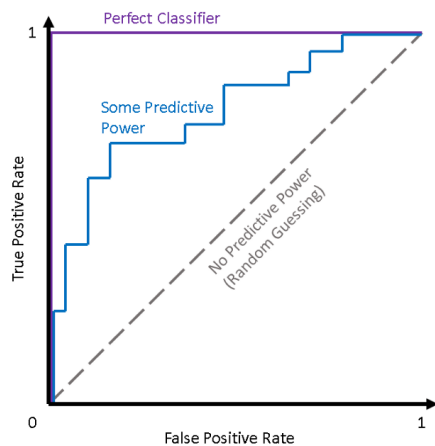
Q.48 What is the difference between SVM and Logistic Regression?

Ans:-

SVM	Logistic Regression
<ul style="list-style-type: none"> SVM tries to find the “best” margin (distance between the line and the support vectors) that separates the classes and this reduces the risk of error on the data 	<ul style="list-style-type: none"> Logistic regression does not, instead it can have different decision boundaries with different weights that are near the optimal point.
<ul style="list-style-type: none"> SVM works well with unstructured and semi-structured data like text and images 	<ul style="list-style-type: none"> Logistic regression works with already identified independent variables.
<ul style="list-style-type: none"> SVM is based on geometrical properties of the data 	<ul style="list-style-type: none"> Logistic regression is based on statistical approaches.
<ul style="list-style-type: none"> The risk of overfitting is less in SVM 	<ul style="list-style-type: none"> Logistic regression is vulnerable to overfitting.

Q.49 What is ROC?

Ans:-



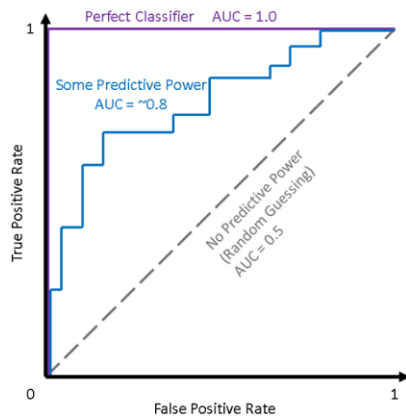
The ROC curve is produced by calculating and plotting the true positive rate against the false positive rate for a single classifier at a variety of thresholds. For example, in logistic regression, the threshold would be the predicted probability of an observation belonging to the positive class. Normally in logistic regression, if an observation is predicted to be positive at > 0.5 probability, it is labelled as positive. ROC curves help us visualize how these choices affect classifier performance.

One advantage presented by ROC curves is that they aid us in finding a classification threshold that suits our specific problem. For example, if we were evaluating an email spam classifier, we would want the false positive rate to be really, really low. We wouldn't want someone to lose an important email to the spam filter just because our algorithm was too aggressive. We would probably even allow a fair amount of actual spam emails (true positives) through the filter just to make sure that no important emails were lost.

Q.50 What is AUC?

Ans:-

While it is useful to visualize a classifier's ROC curve, in many cases we can boil this information down to a single metric — the AUC. AUC stands for area under the (ROC) curve. Generally, the higher the AUC score, the better a classifier performs for the given task.



Q.51 What is the code for plotting ROC curve?

Ans:-

Plot ROC curves

```
fig, ax = plt.subplots(figsize=(6,6))
```

```
ax.plot(lr_fp_rates, lr_tp_rates, label='Logistic Regression')
```

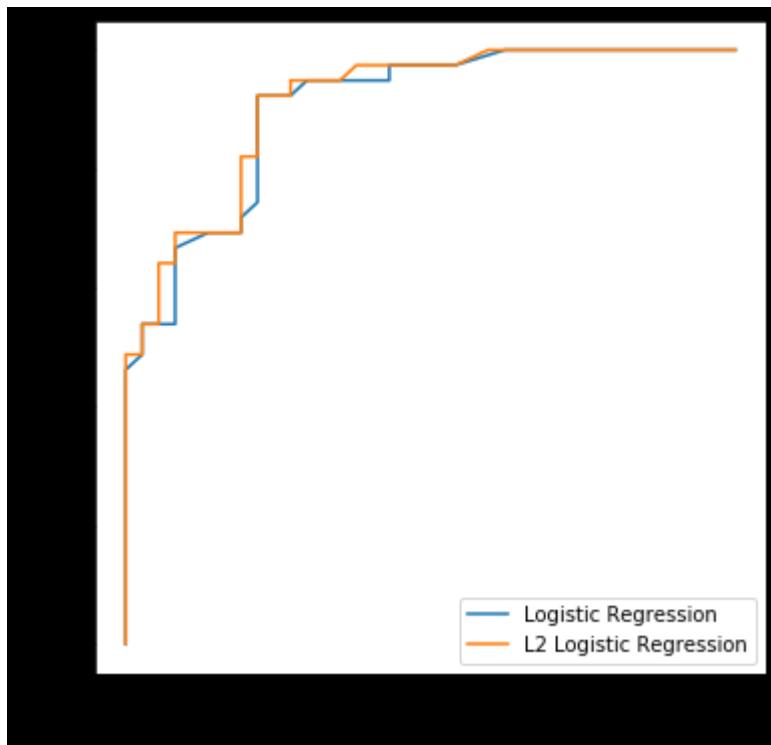
```
ax.plot(l2_fp_rates, l2_tp_rates, label='L2 Logistic Regression')
```

```
ax.set_xlabel('False Positive Rate')
```

```
ax.set_ylabel('True Positive Rate')
```

```
ax.legend();
```

OP:-



Q.52 How to calculate the AUC scores?

Ans:-

The sklearn library has an `auc()` function, which I'll make use of here to calculate the AUC scores for both versions of the classifier. `auc()` takes in the true positive and false positive rates we previously calculated it and returns the AUC score to you.

Code:-

```
# Get AUC scores
```

```
from sklearn.metrics import auc
```

```
print(f'Logistic Regression (No reg.) AUC {auc(lr_fp_rates, lr_tp_rates)}')
```

```
print(f'Logistic Regression (L2 reg.) AUC {auc(l2_fp_rates, l2_tp_rates)}')
```

OP:-

```
Logistic Regression (No reg.) AUC 0.902979902979903
```

```
Logistic Regression (L2 reg.) AUC 0.9116424116424116
```

Q.53 What is SGD Stochastic Gradient Descent Classifier?

Ans:- Stochastic Gradient Descent (SGD) is a simple yet very efficient approach to fitting linear classifiers and regressors under convex loss functions such as Support Vector Machines and Logistic Regression.

The advantages of Stochastic Gradient Descent are:

1. Efficiency.
2. Ease of implementation (lots of opportunities for code tuning).

The disadvantages of Stochastic Gradient Descent include:

1. SGD requires a number of hyperparameters such as the regularization parameter and the number of iterations.
2. SGD is sensitive to feature scaling.

Q.54 What is the difference between (SGD) Stochastic Gradient Descent Classifier and Logistic Regression?

Ans:-

SGD is an optimization method, while Logistic Regression (LR) is a machine learning algorithm/model. You can think of that a machine learning model defines a loss function, and the optimization method minimizes/maximizes it. Some machine learning libraries could make users confused about the two concepts. For instance, in scikit-learn there is a model called SGD Classifier which might mislead some user to think that SGD is a classifier. But no, that's a linear classifier optimized by the SGD.

