Assignment 2

Of

Data Science

BACHELOR OF TECHNOLOGY

COMPUTER SCIENCE AND ENGINEERING

Submitted By :

Aman Chauhan

1805158

Department of Computer Science and Engineering

Guru Nanak Dev Engineering College

Q1. Illustrate the concept of transformations and its following types with examples:

- **Power function**

  Power transformations are needed when the underlying structure is of the form $Y = aX^b$, and transformations on both variables are needed to linearize the function. The linear form of the power function is $\ln(Y) = \ln(aX^b) = \ln(a)+b\ln(X) = b_0+b_1\ln(X)$. The shape of the power function depends on the sign and magnitude of beta. Figure B-5a depicts examples of power functions with beta greater than zero, while Figure B-5b depicts examples of power functions with beta less than zero.

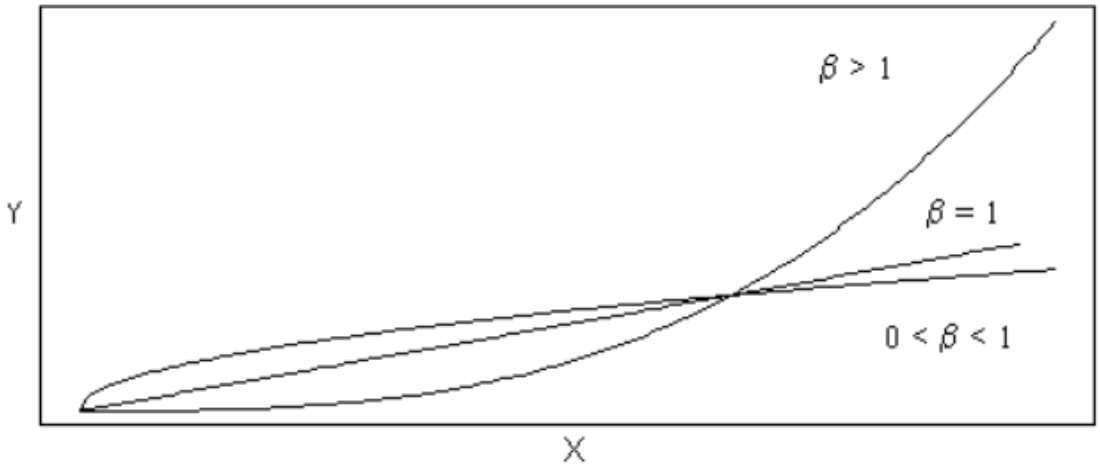


Fig : Power Functions (Functional Form: $Y = aX^b$, where b> 0)

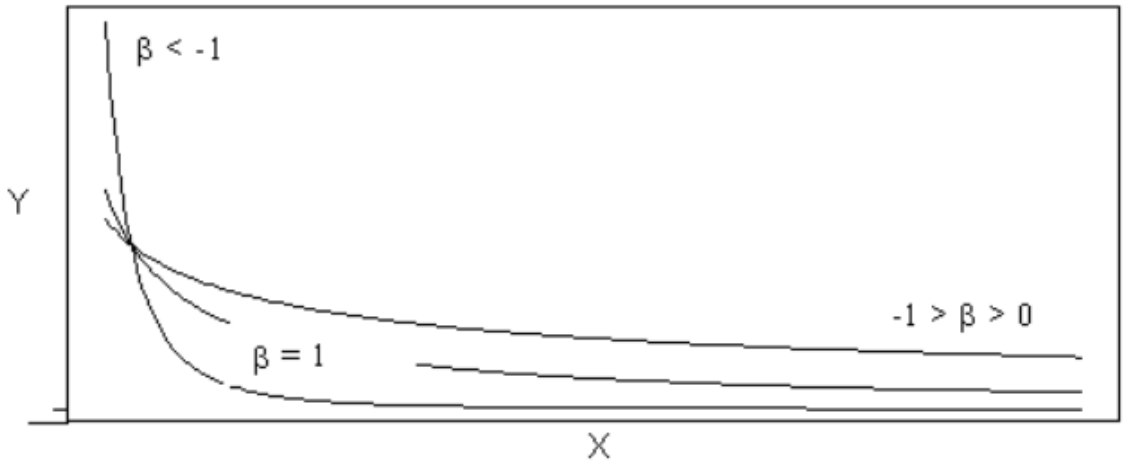


Fig 2 : Power Function (Functional Form: $Y = aX^b$, where $b < 0$)

- **Exponential Function**

The natural log transformation is used to correct heterogeneous variance in some cases, and when the data exhibit curvature between Y and X of a certain type. Figures B-3a and B-3b show the nature of the relationship between Y and X for data that can be linearized using the log transformation. The nature of the underlying relation is $Y = ae^{bx}$, where alpha and beta are parameters of the relation. To get this relation in linear model form, one transforms both sides of the equation to obtain $\ln(Y) = \ln(ae^{bx}) = \ln(a) + \ln(e^{bx}) = \ln(a) + bx = b_0 + b_1x$. In linearized form $b_0 = \ln(a)$ and $b_1 = b$. Figure B-3a shows examples of the relation between Y and X for $b > 0$, while Figure B-3b shows examples of the relation between Y and X for $b < 0$.

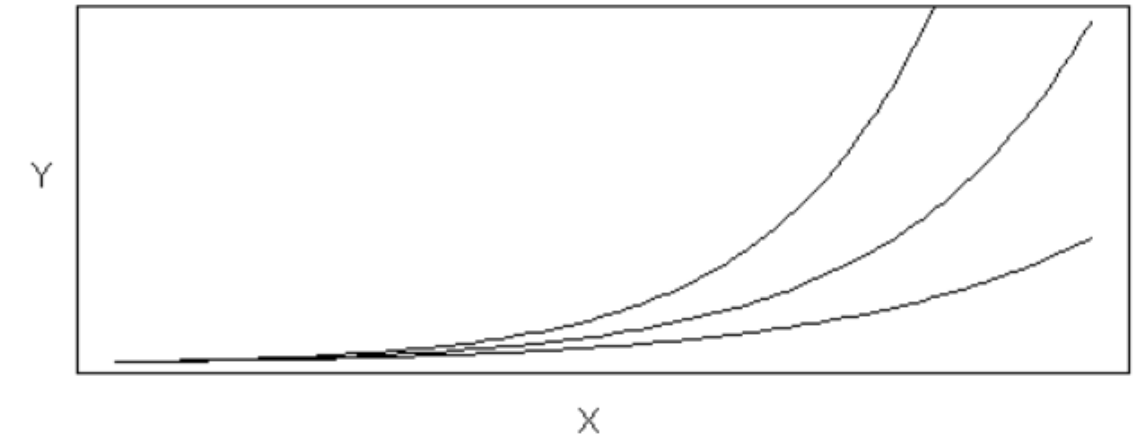


Fig 1 : Exponential Relation (Functional Form: $Y = ae^{bX}$, where $b > 0$)

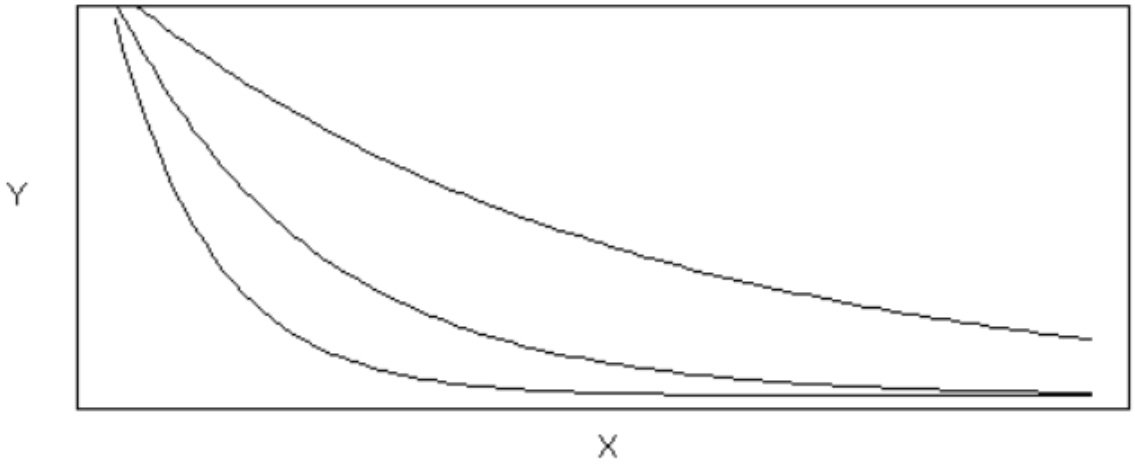


Figure 2 : Exponential Functions (Functional Form: $Y = ae^{bX}$, where $b < 0$)

- **Polynomial Function**

A polynomial function is a function that can be expressed in the form of a polynomial. The definition can be derived from the definition of a polynomial equation. A polynomial is generally represented as P(x). The highest power of the variable of P(x) is known as its degree. Degree of a polynomial function is very important as it tells us about the behavior of the

function P(x) when x becomes very large. The domain of a polynomial function is entire real numbers (R).

Examples:-

x^2+2x+1, $3x-7$, $7x^3+x^2-2$

Types:-

- Constant Polynomial Function: $P(x) = a = ax^0$
- Zero Polynomial Function: $P(x) = 0$; where all a_i's are zero, i = 0, 1, 2, 3, …, n.
- Linear Polynomial Function: $P(x) = ax + b$
- Quadratic Polynomial Function: $P(x) = ax^2+bx+c$

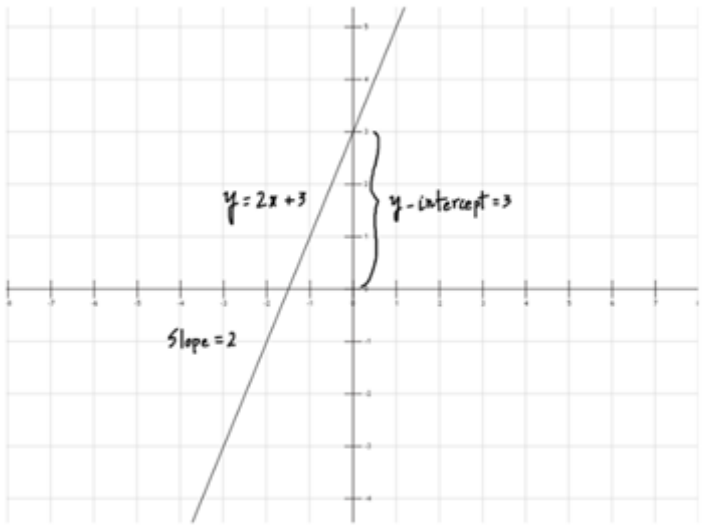


Fig : Polynomial Function

Q2. What is the curse of dimensionality? Explain Principal Component Analysis (PCA) with examples.

Ans: Curse of Dimensionality refers to a set of problems that arise when working with high-dimensional data. The dimension of a dataset corresponds to the number of attributes/features that exist in a dataset. A dataset with a large number of attributes, generally of the order of a hundred or more, is referred to as high dimensional data. Some of the difficulties that come with high dimensional data manifest during analyzing or visualizing the data to identify patterns, and some manifest while training machine learning models. The difficulties related to training machine learning models due to high dimensional data are referred to as the 'Curse of Dimensionality'.

Principal Component Analysis:

Principal Component Analysis is an unsupervised learning algorithm that is used for dimensionality reduction in machine learning. It is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation. These new transformed features are called the Principal Components.

PCA works by considering the variance of each attribute because the high attribute shows the good split between the classes, and hence it reduces the dimensionality. Some real-world applications of PCA are image processing, movie recommendation system, and optimizing the power allocation in various communication channels. It is a feature extraction technique, so it contains the important variables and drops the least important variable.

Some common terms used in the PCA algorithm:

- **Dimensionality:** It is the number of features or variables present in the given dataset. More easily, it is the number of columns present in the dataset.
- **Correlation:** It signifies how strongly two variables are related to each other. Such as if one changes, the other variable also gets changed. The correlation value ranges from -1 to +1. Here, -1 occurs if variables are inversely proportional to each other, and +1 indicates that variables are directly proportional to each other.
- **Orthogonal:** It defines that variables are not correlated to each other, and hence the correlation between the pair of variables is zero.
- **Eigenvectors:** If there is a square matrix M, and a non-zero vector v is given. Then v will be an eigenvector if Av is the scalar multiple of v.
- **Covariance Matrix:** A matrix containing the covariance between the pair of variables is called the Covariance Matrix.

Steps for PCA algorithm:

1. **Getting the dataset :** Firstly, we need to take the input dataset and divide it into two subparts X and Y, where X is the training set, and Y is the validation set.
2. **Representing data into a structure :** Now we will represent our dataset into a structure. Such as we will represent the two-dimensional matrix of independent variable X. Here each row corresponds to the data items, and the column corresponds to the Features. The number of columns is the dimensions of the dataset.
3. **Standardizing the data :** In this step, we will standardize our dataset. Such as in a particular column, the features with high variance are more important

compared to the features with lower variance. If the importance of features is independent of the variance of the feature, then we will divide each data item in a column with the standard deviation of the column. Here we will name the matrix as Z.

4. **Calculating the Covariance of Z :** To calculate the covariance of Z, we will take the matrix Z, and will transpose it. After transpose, we will multiply it by Z. The output matrix will be the Covariance matrix of Z.
5. **Calculating the EigenValues and EigenVectors :** Now we need to calculate the eigenvalues and eigenvectors for the resultant covariance matrix Z. Eigenvectors or the covariance matrix are the directions of the axes with high information. And the coefficients of these eigenvectors are defined as the eigenvalues.
6. **Sorting the EigenVectors :** In this step, we will take all the eigenvalues and will sort them in decreasing order, which means from largest to smallest. And simultaneously sort the eigenvectors accordingly in matrix P of eigenvalues. The resultant matrix will be named as P*.
7. **Calculating the new features Or Principal Components :** Here we will calculate the new features. To do this, we will multiply the P* matrix to the Z. In the resultant matrix Z*, each observation is the linear combination of original features. Each column of the Z* matrices are independent of each other.
8. **Remove less or unimportant features from the new dataset :** The new feature set has occurred, so we will decide here what to keep and what to remove. It means, we will only keep the relevant or important features in the new dataset, and unimportant features will be removed.

Q3. Calculate correlation coefficient for the following data by short-cut method:

X	10	12	14	18	20
Y	5	6	7	10	12

Ans:

X	Y	dx = X-A	dy = Y-A	dx^2	dy^2	$dxdy$
10	5	-4	-2	16	4	8
12	6	-2	-1	4	1	2
14	7	0	0	0	0	0
18	10	4	3	16	9	12
20	12	6	5	36	25	30
N=5	**N=5**	$\sum dx$=4	$\sum dy$=5	$\sum dx^2$=72	$\sum dy^2$=39	$\sum dxdy$ =52

Now, Correlation Coefficient r is

$$r = \frac{\sum dxdy - \frac{(\sum dx \times \sum dy)}{N}}{\sqrt{\sum dx^2 - \frac{(\sum dx)^2}{N}} \times \sqrt{\sum dy^2 - \frac{(\sum dy)^2}{N}}}$$

$$r = \frac{52 - \frac{20}{5}}{\sqrt{72 - \frac{16}{5}} \times \sqrt{39 - \frac{25}{5}}}$$

$$r = \frac{48}{\sqrt{68.8} \times \sqrt{34}} = \frac{48}{8.294 \times 5.830} = \frac{48}{48.354}$$

$$r = 0.992$$

Q4. List the characteristics of the Normal Distribution curve.

Ans :

Characteristics :

- The mean, median, and mode are all equal.
- The curve is known to be symmetric at the center, which is around the mean.
- Exactly 1/2 of all the values are known to be to the left of center whereas exactly half of all the values are to the right of the center.
- The total area under the curve is 1.

Q5. Write a short note on:

i) Confusion Matrix :

The Confusion Matrix is the visual representation of the Actual VS Predicted values. It measures the performance of our Machine Learning classification model and looks like a table-like structure.

This is how a Confusion Matrix of a binary classification problem looks like :

Actual Values

Predicted Values		True	False
	True	True Positive	False Positive
	False	False Negative	True Negative

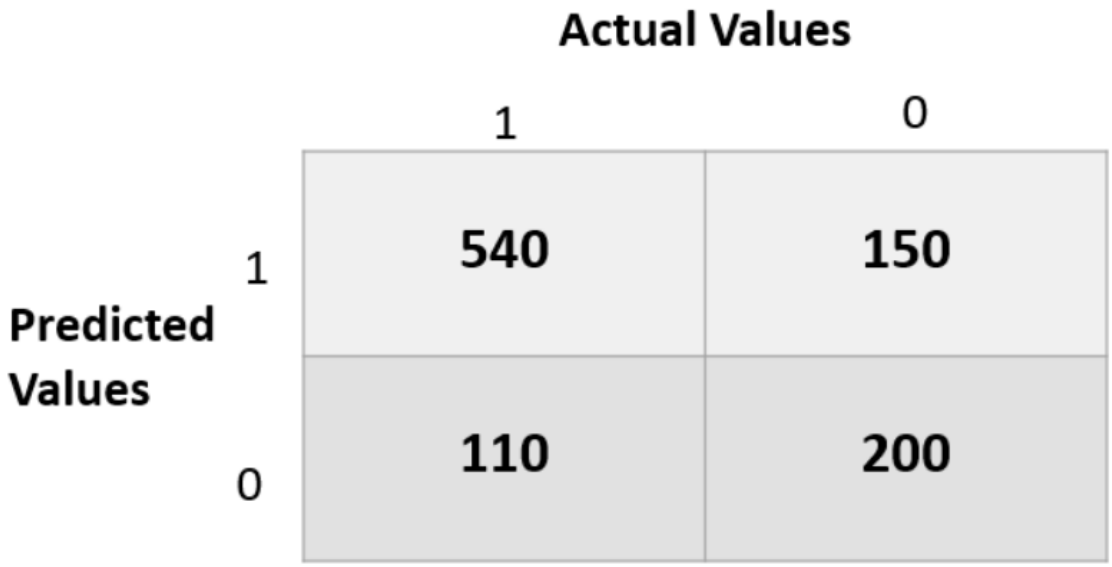
True Positive: The values which were actually positive and were predicted positive.

False Positive: The values which were actually negative but falsely predicted as positive. Also known as Type I Error.

False Negative: The values which were actually positive but falsely predicted as negative. Also known as Type II Error.

True Negative: The values which were actually negative and were predicted to be negative.

Examples:



True positive: 540 records of the stock market crash were **predicted correctly** by the model.

False-positive: 150 records of not a stock market crash were **wrongly predicted** as a market crash.

False-negative: 110 records of a market crash were **wrongly predicted** as not a market crash.

True Negative: 200 records of not a market crash were predicted correctly by the model.

ii) ROC Curve:

The ROC or Receiver Operating Characteristic plot is used to visualize the performance of a binary classifier. It gives us the trade-off between the True Positive Rate (TPR) and the False Positive Rate (FPR) at different classification thresholds.

True Positive Rate:

True Positive Rate is the proportion of observations that are correctly predicted to be positive.

$$\mathbf{TPR} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

False Positive Rate:

False Positive Rate is the proportion of observations that are incorrectly predicted to be positive.

$$\mathbf{FPR} = \frac{False\ Positive}{True\ Negative + False\ Positive}$$

For different threshold values we will get different TPR and FPR. So, in order to visualize which threshold is best suited for the classifier we plot the ROC curve. The following figure shows what a typical ROC curve looks like.

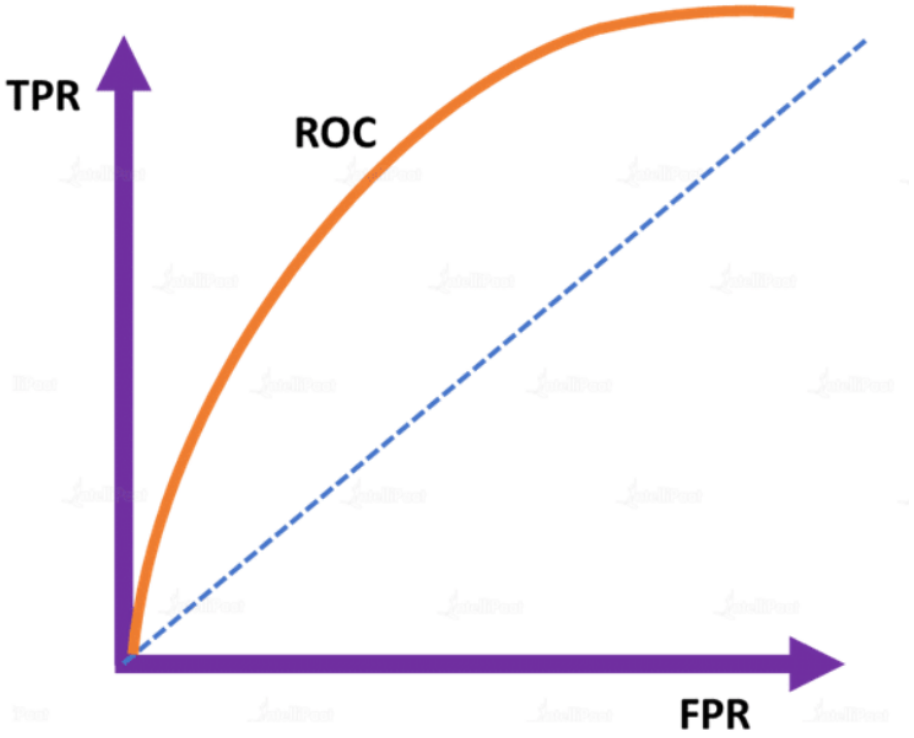


Fig : ROC Curve