



MA-515 PROJECT

KARAN DEEP DAS(2021MCB1236)

VAIBHAV KUMAR(2021MCB1219)

UDAY NALLURI (2021MCB1253)

KANHAIYA KUMAR SAHU(2021MCB1235)

The image features a dark navy blue background. On the left side, there are two overlapping geometric shapes: a blue parallelogram and a light green parallelogram, both tilted at an angle. The word "CARAVAN" is written in a white, serif, all-caps font, centered horizontally in the middle of the image.

CARAVAN



Input data and what to do ?

Using logistic regression and LDA predict whether customer will buy caravan insurance policy .

Compare the outputs obtained from different methods.



LOGISTIC REGRESSION

$$p(X) = \frac{e^{\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p}}$$

$$\beta^1 = \beta^0 + [X^T W X]^{-1} \cdot X^T (y - \mu)$$


- The above mentioned formula is for Logistic regression. $X_1, X_2, X_3, \dots, X_n$ are the parameters, in our case we only need 4 of them.
- $\beta_0, \beta_1, \dots, \beta_k$ are Multiple regression coefficients. We evaluate those by using the formula given in the right
- We predict the class of the object using . depending on this value we put the observations into different classes.



LDA

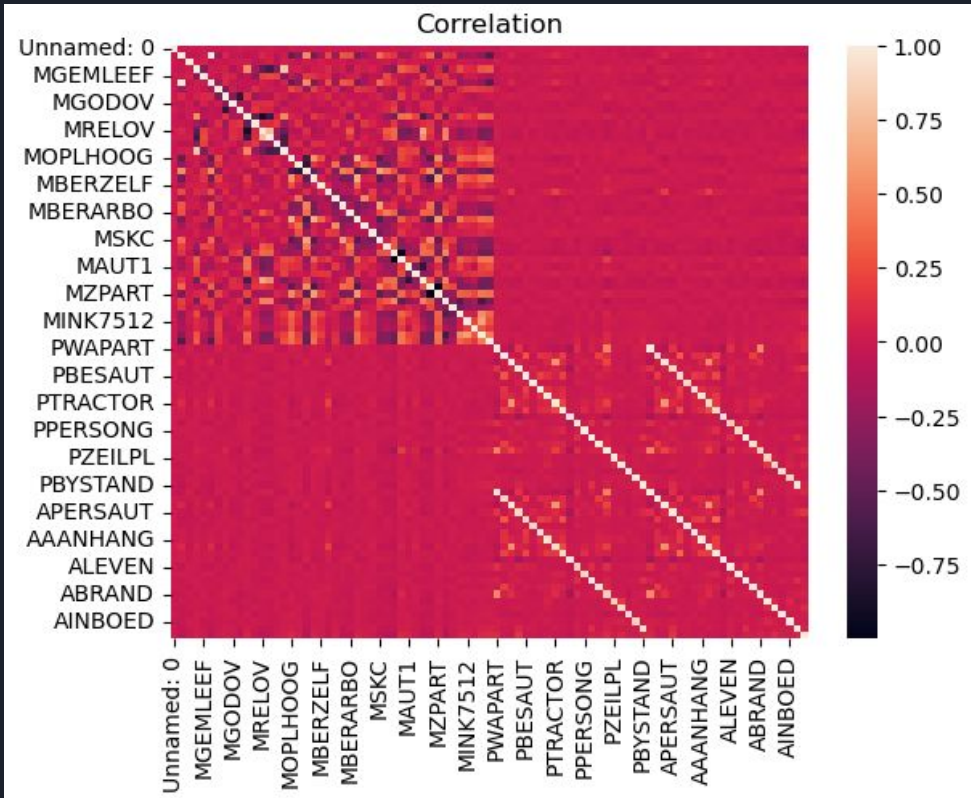
Linear Discriminant Analysis (LDA) is a statistical technique used for dimensionality reduction and classification. It's a supervised algorithm commonly employed in machine learning and statistics. LDA is particularly useful when dealing with classification problems where the goal is to separate two or more classes based on their features.

- Two scatter matrices are computed in LDA: the within-class scatter matrix (S_w) and the between-class scatter matrix (S_b), which measure the spread of data within each class and the spread between the class means. Next, In LDA we determine the eigenvectors and eigenvalues of the matrix $S_w^{-1} * S_b$. The eigenvectors indicate the directions that maximise the separation between classes. Projection onto Lower-Dimensional Space: The data is projected onto a lower-dimensional subspace using the transformation matrix formed by the eigenvectors corresponding to the largest eigenvalues.
- In the lower-dimensional space, a decision rule is established for classification. For example, in a binary classification problem, a new data point is assigned to the class with the closest mean in the transformed space.



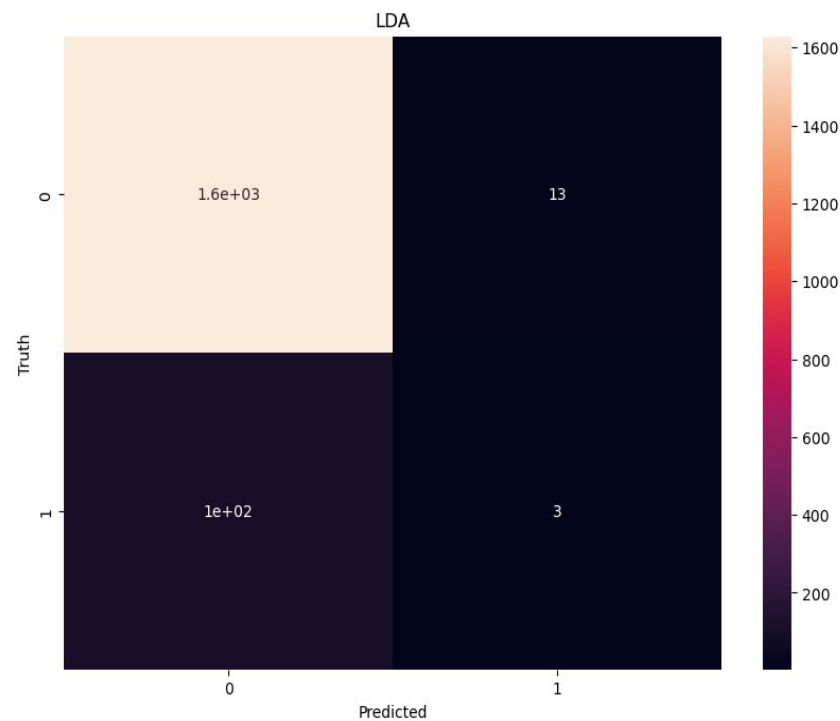
Some necessary transforms before applying the algorithms

- Using panda we read the data given in Caravan.csv file and get data info, description, conversion of categorical values such as “Yes” “No” in purchase column to 0 and 1 respectively to be able to apply our models.
- We then get the correlation of columns and remove the column with high correlation factor in order to reduce redundancy and the amount of noise present in the dataset.
- From the given input data we are splitting into training (70%) and testing data (30%) using `train_test_split`.
- We scaled the data using Standard scaler , MinMax scalar depending on our requirement .
- After these preprocessing we apply Logistic Regression, LDA and then we generate the corresponding confusion matrix for comparison of the models

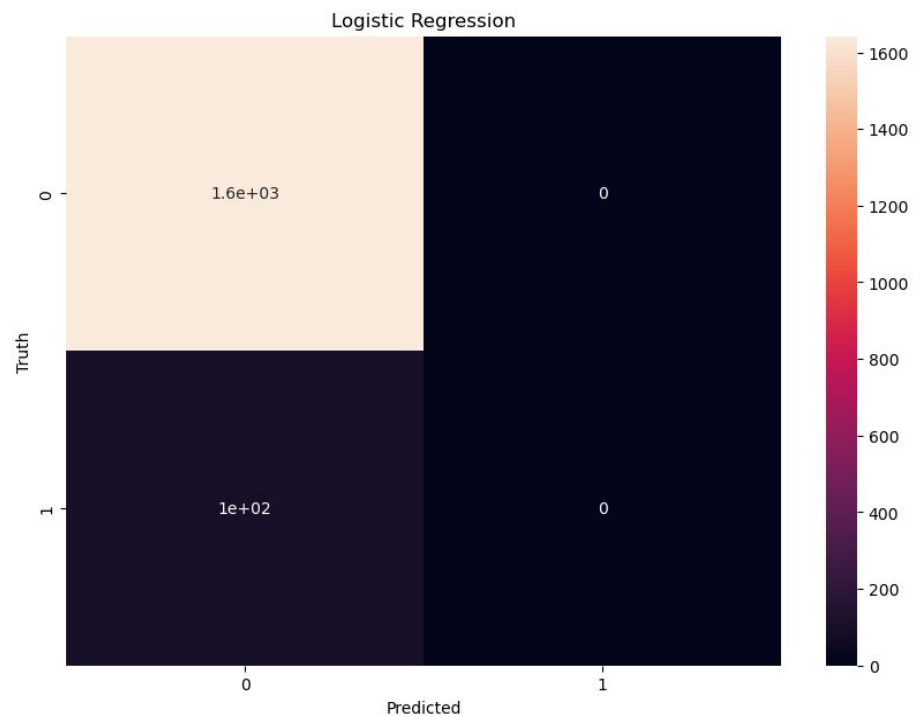


HeatMap of correlation of columns in
Caravan Policy Dataset

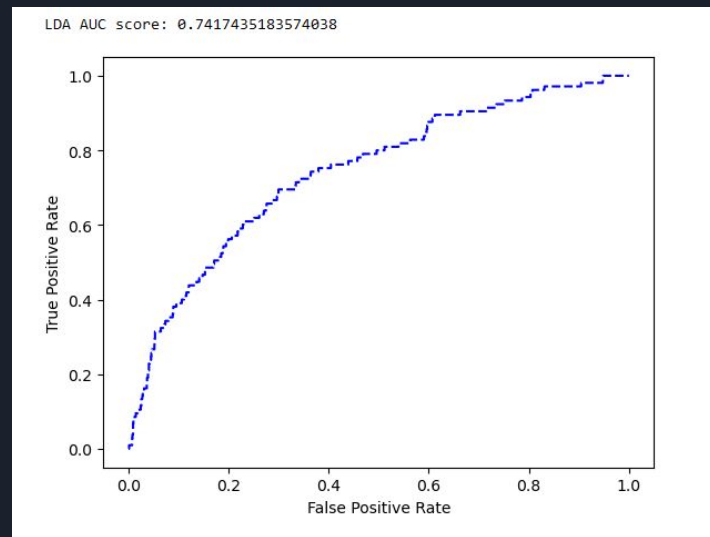
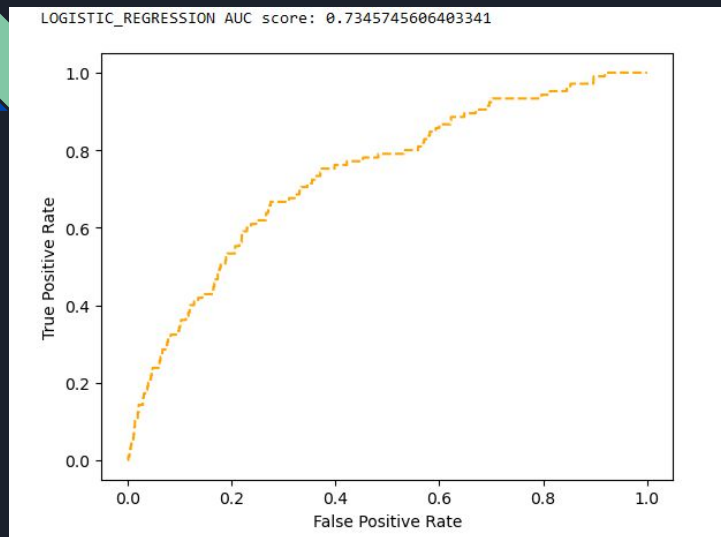
accuracy of LDA MODEL= 0.9341728677733258



accuracy of LOGISTIC REGRESSION MODEL= 0.9398969662278192



Confusion Matrix for LDA and Logistic Regression Model



LDA and logistic Regression AUC score from ROC Curve



RESULTS

From the confusion matrix we get accuracy score of LDA 93.41% and accuracy score of Logistic Regression Model 93.41% .

And From ROC curve we know the AUC value closer to 1 is the considered better model.

From ROC Curve of LDA model we have AUC score 0.734

From ROC Curve of Logistic Regression model we have AUC score 0.741

Hence the LDA Models is the better model for Caravan Policy purchasing prediction asIf you have two models and you're comparing their AUC scores:

- If Model A has a higher AUC than Model B, then Model A is generally considered better in terms of discriminatory power.



BANKRUPTCY



Input data and what to do ?

Given an Bankruptcy prediction data, we need to use logistic, KNN and Decision Tree model to predict Bankruptcy based on several parameters such as ROA(C) before interest and depreciation before interest, ROA(A) before interest and % after tax, ROA(B) before interest and depreciation after tax, Operating Gross Margin, Realized Sales Gross Margin, Operating Profit Rate, Pre-tax net Interest Rate, etc.

We will compare accuracies of each models along with ROC and AUC for each model.

Data and its Description

Descriptive analysis table of original data, but only include few column due to space limitation.

```
# Descriptive stats  
data.describe()
```

	Bankrupt?	ROA(C) before interest and depreciation before interest	ROA(A) before interest and % after tax	ROA(B) before interest and depreciation after tax	Operating Gross Margin	Realized Sales Gross Margin	Operating Profit Rate	Pre-tax net Interest Rate	After-tax net Interest Rate
count	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000	6819.000000
mean	0.032263	0.505180	0.558625	0.553589	0.607948	0.607929	0.998755	0.797190	0.809084
std	0.176710	0.060686	0.065620	0.061595	0.016934	0.016916	0.013010	0.012869	0.013601
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.476527	0.535543	0.527277	0.600445	0.600434	0.998969	0.797386	0.809312
50%	0.000000	0.502706	0.559802	0.552278	0.605997	0.605976	0.999022	0.797464	0.809375
75%	0.000000	0.535563	0.589157	0.584105	0.613914	0.613842	0.999095	0.797579	0.809469
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

8 rows × 96 columns

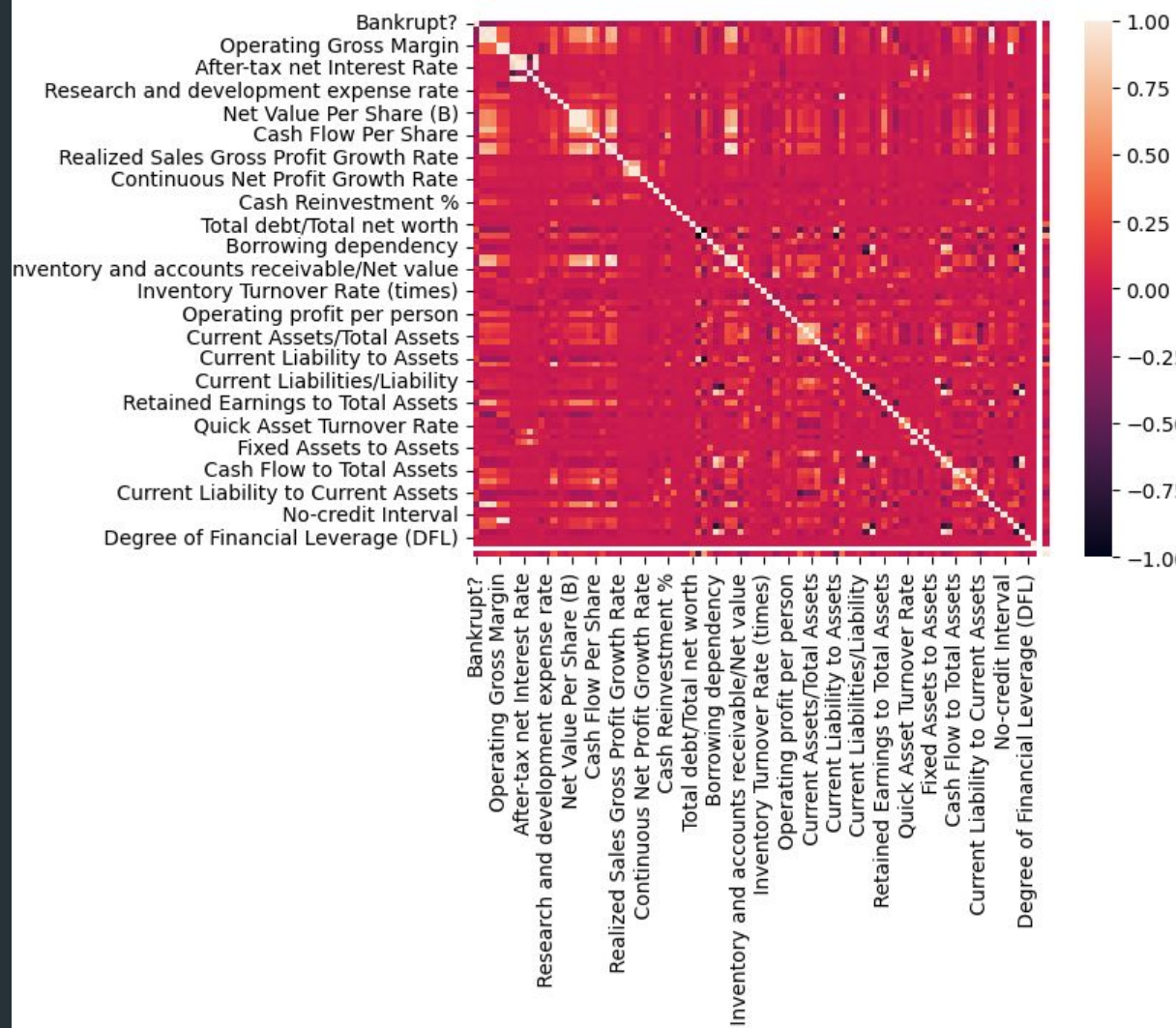
- Checking for null values in the data.

#	Column	Non-Null Count		Dtype
---	-----	-----	-----	-----
0	Bankrupt?	6819	non-null	int64
1	ROA(C) before interest and depreciation before interest	6819	non-null	float64
2	ROA(A) before interest and % after tax	6819	non-null	float64
3	ROA(B) before interest and depreciation after tax	6819	non-null	float64
4	Operating Gross Margin	6819	non-null	float64
5	Realized Sales Gross Margin	6819	non-null	float64
6	Operating Profit Rate	6819	non-null	float64
7	Pre-tax net Interest Rate	6819	non-null	float64
8	After-tax net Interest Rate	6819	non-null	float64
9	Non-industry income and expenditure/revenue	6819	non-null	float64
10	Continuous interest rate (after tax)	6819	non-null	float64
11	Operating Expense Rate	6819	non-null	float64
12	Research and development expense rate	6819	non-null	float64
13	Cash flow rate	6819	non-null	float64
14	Interest-bearing debt interest rate	6819	non-null	float64
15	Tax rate (A)	6819	non-null	float64
16	Net Value Per Share (B)	6819	non-null	float64
17	Net Value Per Share (A)	6819	non-null	float64
18	Net Value Per Share (C)	6819	non-null	float64
19	Persistent EPS in the Last Four Seasons	6819	non-null	float64
...				
94	Net Income Flag	6819	non-null	int64
95	Equity to Liability	6819	non-null	float64

We found that there are no column having null values.

- _____
- _____
- _____

Concerning which we removed columns with correlation ≥ 0.9 and as a result number of columns now reduced to 78 from 96.





K Nearest Neighbors Algorithm

- Basically in this algorithm we calculate the distance between the particular observation we need to predict the class of and all other observations.
- We take the k nearest observation from all this distance. That is we pick the k nearest neighbors of our corresponding observation.
- Among these K nearest neighbors we check in which class most of these k neighbors lies
- We predict the same class for our observation.
- K can be varied in algorithm along with how we measure distance.
- For this following project i am evaluating the model for $k = 1$ to 20 and im using euclidean distance for calculating distance between the observation.



Accuracy score for KNN model for the k-value from 1 to 20

```
0.9530791788856305
0.967741935483871
0.9638318670576735
0.9667644183773216
0.966275659824047
0.966275659824047
0.9672531769305963
0.9657869012707723
0.9657869012707723
0.9667644183773216
0.9667644183773216
0.9667644183773216
0.967741935483871
0.966275659824047
0.966275659824047
0.9657869012707723
0.966275659824047
0.9652981427174976
0.9652981427174976
```

The maximum accuracy and corresponding k value.

```
(2, 0.967741935483871)
```

Decision Tree Classification Algorithm

- Compute entropy of the dataset(S)

entropy:

$$H(S) = \sum_{c \in C} -p(c) \log_2 p(c)$$

- Calculate "Information Gain" for every attribute

Information Gain:

Attribute A.

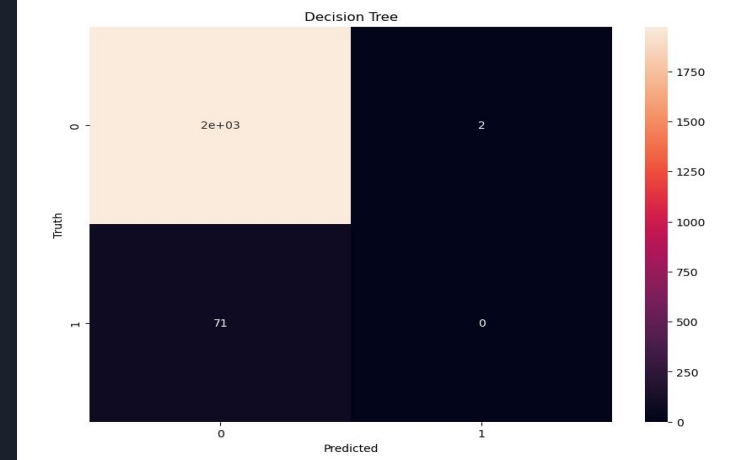
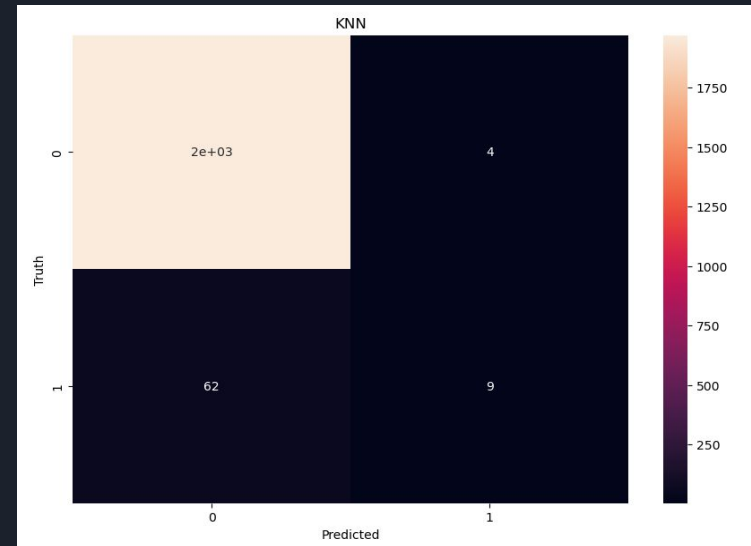
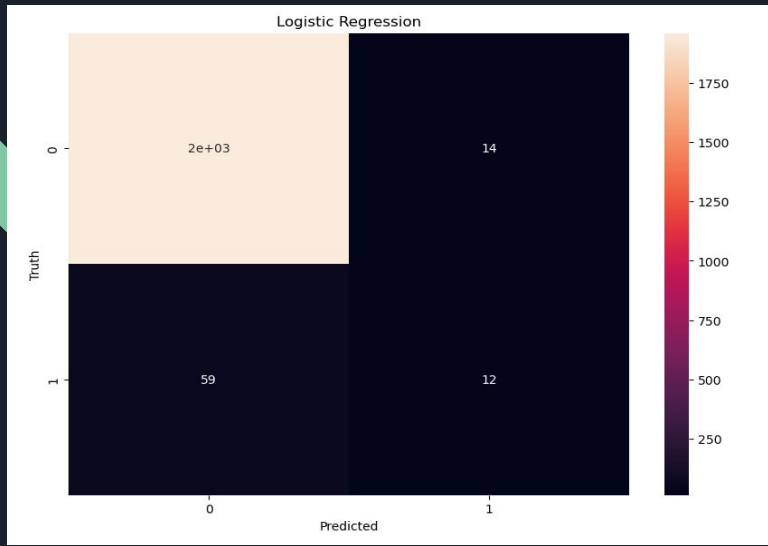
$$IG(A, S) = H(S) - \sum_{i \in I} p(S_i) H(S_i)$$

- Pick the attribute having highest Information Gain and set it as root.
- Go down the current node and repeat the process from step 1 until the sample size in current node becomes less than minimum sample split.
- Decision trees classify instances by sorting them down the tree from the root node to some leaf node. Leaf node classify the instance.



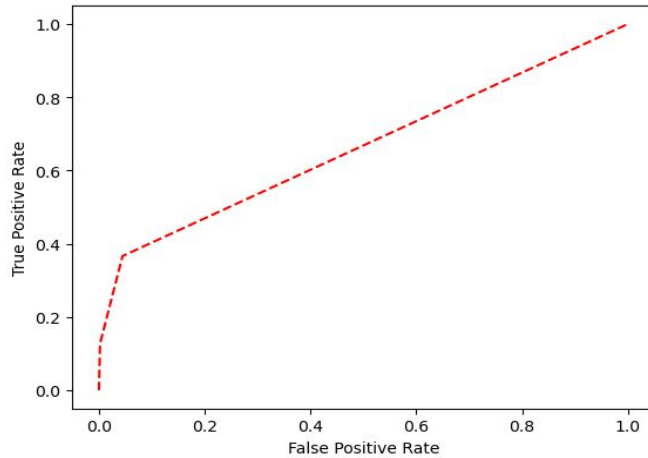
Some necessary transforms before applying the algorithms

- Using panda we read the data given in Bankruptcy.csv file and get data info, description, checked for “null value” in the columns but we found no null values.
- We then get the correlation of columns and remove the column with high correlation factor in order to reduce redundancy and the amount of noise present in the dataset.
- From the given input data we are splitting into training (70%) and testing data (30%) using train_test_split.
- We scaled the data using Standard scaler , MinMax scalar depending on our requirement .
- After these preprocessing we apply Logistic Regression, KNN , Decision Tree algorithms and then we generate the corresponding confusion matrix for comparison of the models



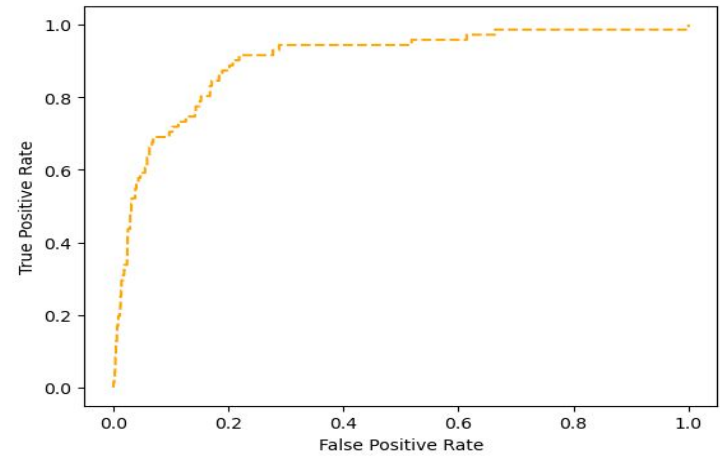
- Confusion Matrix Corresponding to Logistic Regression, KNN and Decision Tree Model

KNN AUC score: 0.6632733107505795

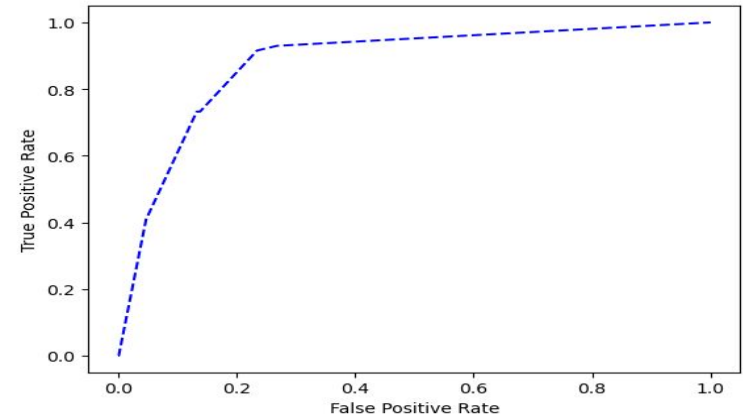


- ROC Curve and AUC score for corresponding KNN, Logistic Regression and Decision Tree Model

LOGISTIC_REGRESSION AUC score: 0.8999679087181315



Decision Tree AUC score: 0.8787662684970584





RESULTS

From the confusion matrix we get accuracy score of Logistic Regression 96.50% and accuracy score of KNN Model 96.774% and accuracy score of Decision Tree Model is 96.529%.

And From ROC curve we know the AUC value closer to 1 is the considered better model.

From ROC Curve of KNN model we have AUC score 0.663

From ROC Curve of Logistic Regression model we have AUC score 0.899

From ROC Curve of Decision Tree model we have AUC score 0.8787

Hence the Logistic Regression Model is the better model for bankruptcy prediction as If you have two models and you're comparing their AUC scores:

- If Model A has a higher AUC than Model B, then Model A is generally considered better in terms of discriminatory power.



ADVERTISEMENT



Input data and what to do ?

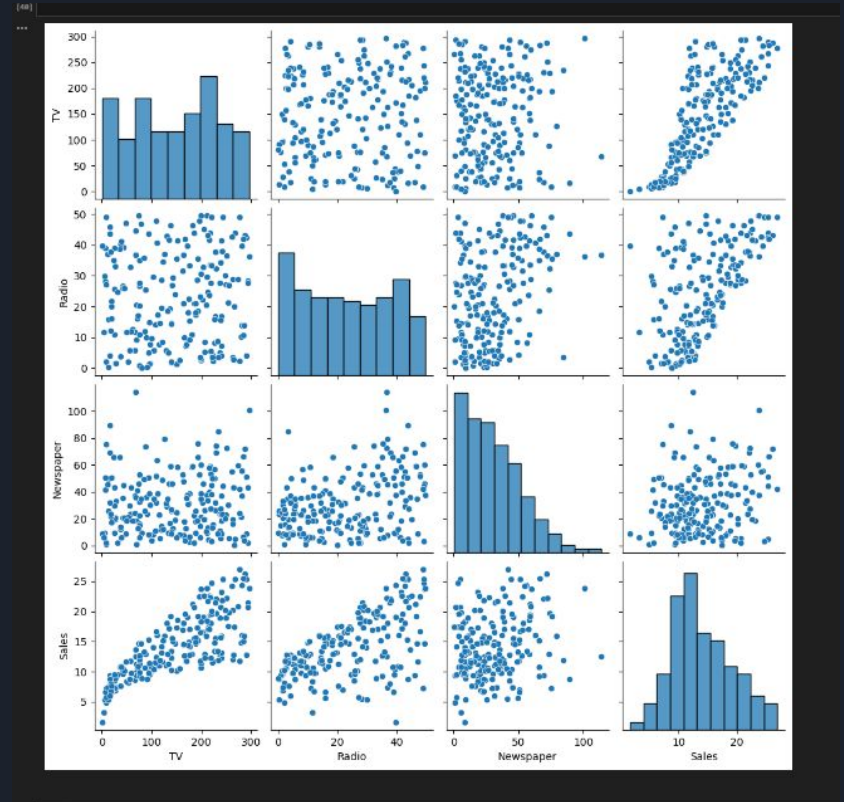
Given an Advertisement data, we need to use multiple linear regression, ridge and lasso techniques to predict the Sales. Compare different methods.

We will compare result given by different methods.

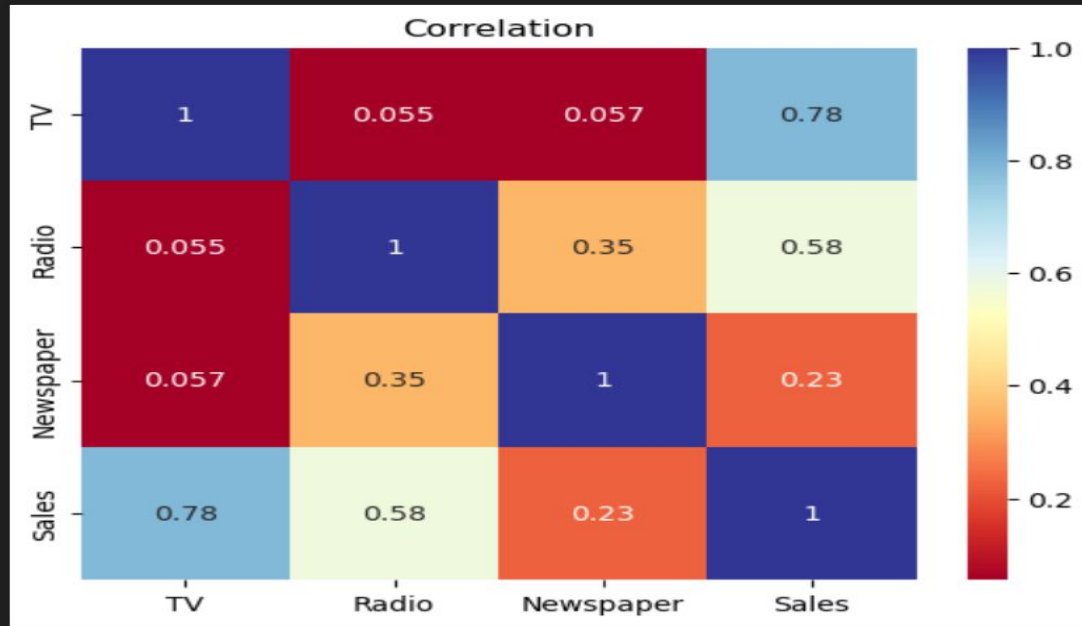
Unnamed: 0					
	TV	Radio	Newspaper	Sales	
0	1	230.1	37.8	69.2	22.1
1	2	44.5	39.3	45.1	10.4
2	3	17.2	45.9	69.3	9.3
3	4	151.5	41.3	58.5	18.5
4	5	180.8	10.8	58.4	12.9

Given data and representation

The correlation representation of all the predictor are given in the diagram



Correlation Matrix of all columns of dataset





Some necessary transforms before applying the algorithms

1. We read data using pandas library . we get data info and description and then we remove columns like index and unnamed column and other column having high correlation factor. Correlation factor are shown with the help of heat map using seaborn library .
2. From the given input data we are splitting into training (70%)and testing data (30%) using train_test_split.
3. We scaled the data using Standard scaler , MinMax scalar depending on our requirement .
4. Then we applied different models for regression and get their corresponding RMSE, MAE,MSE,R-square .



COMPARISON OF DIFFERENT MODELS

When we compare two different model we have to look for minimum error (MSE,RSE) term in different models and compare them lower the values of MSE and RMSE indicates better model .

So , we can conclude that Lasso is the best model

Ridge MODEL

R²: 0.1312094042214662

MAE: 3.689649368329414

MSE: 23.721986941416183

RMSE: 4.870522245243952

Lasso MODEL

R²: 0.15044348151224307

MAE: 3.82956309005576

MSE: 23.196807994338464

RMSE: 4.816306468066423


MLR MODEL

R²: 0.12997792345005088

MAE: 3.6897799378710037

MSE: 23.755612041549714

RMSE: 4.873972921708708



Conclusion of Advertising Sales Prediction Models

IN this part of analysis we have seen all the different kinds of models like Lasso , Ridge , multiple linear regression Lasso models is the best compared to other models results compared on the following basis:

Mean Squared Error (MSE): MSE values that are lower are preferable. Larger errors are penalised by MSE more than smaller errors.

- In terms of minimising squared errors, Model A is deemed superior if its MSE is lower than Model B's.
- The lower the Mean Absolute Error (MAE) value, the better. MAE handles every error the same way.
- In terms of minimising absolute mistakes, Model A is deemed superior if its MAE is lower than that of Model B.
- Root Mean Squared Error (RMSE): Lower RMSE values are preferable, much like MSE. Since RMSE and the target variable have the same unit, it may be interpreted more easily.
- In terms of minimising square root of squared errors, Model A is deemed superior if its RMSE is lower than Model B's.

A decorative graphic in the top-left corner consisting of two overlapping parallelograms: a blue one in the foreground and a light green one behind it, both slanted at a 45-degree angle.

SVD IMAGE COMPRESSION



Singular Value Decomposition (SVD):

SVD is a matrix factorization method that decomposes a matrix into three separate matrices. For a given matrix A , SVD is represented as:

$$A=USV(T)$$

- U is an orthogonal matrix representing the left singular vectors.
- S is a diagonal matrix containing the singular values.
- $V(T)$ is the transpose of an orthogonal matrix representing the right singular vectors.




Image Compression Using Singular Value Decomposition (SVD)

Step 1: Convert Image to Matrix

Convert the image into a matrix representation, where each element corresponds to the pixel intensity.

Step 2: Apply Singular Value Decomposition (SVD)

Apply SVD to the image matrix, decomposing it into three matrices: U , S , and $V(T)$

Step 3: Select Top Singular Values Retain only the top k singular values in the diagonal matrix S and corresponding columns in U and $V(T)$. The value of k determines the level of compression.

Step 4: Reconstruct Compressed Image

Reconstruct the compressed image using the selected components U_k , S_k , $V(T)_k$



Step 5: Visualization and Comparison

Visualize both the original and compressed images for comparison. Assess the trade-off between image quality and file size based on the chosen k value.

Additional Considerations:

Adjust k Value:

Experiment with different values of k to control the level of compression. Higher k values result in better image quality but larger file sizes.

Compression Ratio:

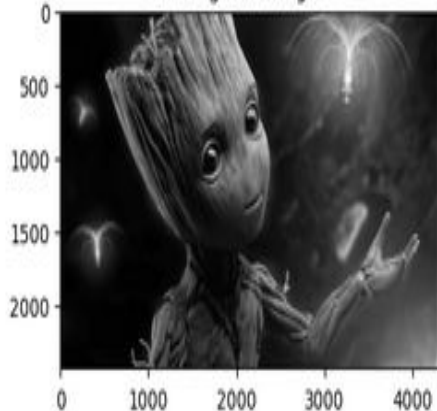
Calculate the compression ratio by dividing the original image size by the compressed image size. This provides insights into the effectiveness of the compression.

Iterative Experimentation:

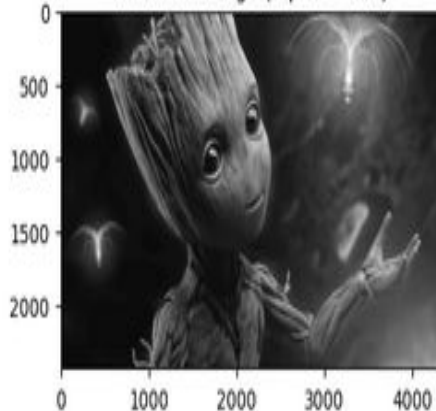
Depending on the application, iterate and experiment with different compression levels to find a balance that meets specific requirements for image quality and file size.

Results of Compression

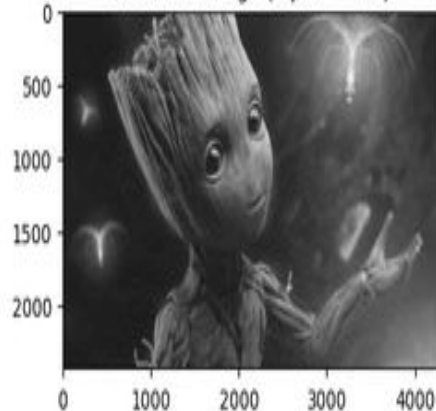
Original Image



Reduced Image (alpha = 0.9)



Reduced Image (alpha = 0.6)



Reduced Image (alpha = 0.3)

