STATISTICAL APPLICATIONS FOR DATA ANALYTICS II

REPORT-1: EXPLORING DATA ANALYTICS TECHNIQUES WITH PYTHON

DATE: 06/19/2024

SURAJ KUMAR MISHRA - 8902749

GUIDED BY:

PROF. JONATHAN PLUMBTREE

DATA EXPLORATION

- For the regression and classification analysis we use two different datasets.

  - **Diamond Dataset**
    - There are 53,940 diamonds in the dataset with 10 features (carat, cut, color, clarity, depth, table, price, length_mm, width_mm, depth_mm). The Price is shown in '$' and most variables are numeric but cut, color, and clarity are categorical variables.

Outcome/ Dependent Variable: Diamond Price in USD (Integer, Quantitative)

Independent Variables/Predictors:

Variable Name	Data Type, Type of Variable	Description
Carat	Float, Quantitative	Weight of diamond (0.2 - 5.01)
Cut	Nominal, Qualitative	Quality of the cut (Fair, Good, Very Good, Premium, Ideal)
Color	Nominal, Qualitative	Diamond color, from J (worst) to D (best)
Clarity	Nominal, Qualitative	A measurement of how clear the diamond is (I1 (worst), SI2, SI1, VS2, VS1, VVS2, VVS1, IF (best))
x	Float, Quantitative	Length of diamond in mm (0 – 10.74)
y	Float, Quantitative	Width of diamond in mm (0 – 58.90)
z	Float, Quantitative	Depth of diamond in mm (0 – 31.80)
Depth %	Float, Quantitative	Total depth percentage = z / mean(x, y) = 2 * z / (x + y) (43--79)
Table %	Float, Quantitative	Width of the top of the diamond relative to the widest point (43--95)

  - **Employee Attrition Dataset**
    - There are 1470 employee data in the dataset with 35 different variables. HR Analytics helps us with interpreting organizational data. Attrition in a corporate setup is one of the complex challenges that the people, managers, and the HRs personnel have to deal with.

Outcome/ Dependent Variable: Attrition Rate (**Nominal, Qualitative)**
Description: Attrition rate takes values as Yes and No.

Independent Variables/Predictors:

Variable Name	**Data Type, Type of Variable**	**Description**
Gender	**Nominal, Qualitative**	This denotes the Gender of the employee in the dataset and has these values: Male, Female
Department	**Nominal, Qualitative**	Department the employee belongs to and can be one of the values: 'Human resources', 'Research & Development', and ' Sales'
Education Field	**Nominal, Qualitative**	These are domains in which an employee has an education background in:'Human Resources', 'Life Science',' Marketing',' Medical', 'Technical Degrees'
Business Travel	**Nominal, Qualitative**	This extracts the frequency of travel of each employee and can take these values: 'Travel_Rarely',' Travel_Frequently',' Non-Travel'
Marital status	**Nominal, Qualitative**	This can take one of the values: 'Divorced',' Married', 'Single'
Job role	**Nominal, Qualitative**	This denotes employee designation and can be one of the following: 'Health Representative', 'Human Resources', Laboratory Technician',' Manager',' Manufacturing Director',' Research Director',' Research Scientist',' Sales Executive',' Sales Representative'
Over time	**Nominal, Qualitative**	This specifies whether a person is doing overtime or not: Yes, No
Education	**Ordinal, Qualitative**	Take values as below : 1 'Below College', 2 'College', 3 'Bachelor', 4 'Master', 5 'Doctor'
Environment Satisfaction	**Ordinal, Qualitative**	Take values as below : 1 'Low', 2 'Medium', 3 'High', 4 'Very High'
Job Involvement	**Ordinal, Qualitative**	Take values as below : 1 'Low', 2 'Medium', 3 'High', 4 'Very High'
job satisfaction	**Ordinal, Qualitative**	1 'Low', 2 'Medium', 3 'High', 4 'Very High'
PerformanceRating	**Ordinal, Qualitative**	1 'Low',2 'Good', 3 'Excellent', 4 'Outstanding'
Relationship Satisfaction	**Ordinal, Qualitative**	1 'Low', 2 'Medium', 3 'High', 4 'Very High'
Work-life Balance	**Ordinal, Qualitative**	1 'Bad',2 'Good',3 'Better',4 'Best'

❖ Python libraries used in the Analysis

- **Numpy** – Numerical computations
- **Pandas** – dataframe handling and data cleaning
- **Seaborn** – Data visualizations
- **Scipy** – Model statistics

- **Matplotlib** – Plotting and Graphs
- **Sklearn** – Regression and Classification
- **Warnings**
- **Spacy** – Text Analysis
- **Os**

DESCRIPTIVE ANALYTICS

Below are the different statistics to describe measures of central tendency for the diamond dataset for each of the variables:

	Count	Mean	Std	Min	25%	50%	75%	max
Carat	53920	0.7976	0.4737	0.2	0.40	0.7	1.04	5.01
Depth	53920	61.749	1.4323	43.0	61.00	61.80	62.50	79.00
Table	53920	57.457	2.2340	43.0	56.00	57.00	59.00	95.00
Price	53920	3930.9932	3987.280	326.0	949.00	2401.00	5323.25	18823.00
Length_mm	53920	5.731	1.119	3.73	4.71	5.70	6.54	10.74
Depth_mm	53920	5.734	1.140	3.68	4.72	5.71	6.54	58.90
Width_mm	53920	3.540	0.7025	1.07	2.91	3.53	4.04	31.80

Significance of the above statistical measures:

1. **Mean**: We use the mean to calculate the arithmetic average of all the observations of a variable in the dataset. It tells the value where most of the observations of a variable in the dataset are centered.

   One disadvantage of the mean is that it is not a good measure if there are extreme values in the dataset as the mean does not depict the true center of the observations in such situations.

2. **Median**: We use this measure to calculate the center of all observations of a variable which divides the observation into exactly two halves after sorting all values in ascending order. In other words, 50% of the values to the left are less than the median and 50% of the values to the right are more than the median.

   The median is a good measure of central tendency when there are extreme values in a variable and is the preferred statistic in such cases over the mean.

3. **Mode**: this statistic represents the most frequently occurring observation in a variable. Usually, we use a frequency distribution to fetch the number of times each observation occurs in the variable. We then sort the frequencies in descending order to get the observation is maximum occuring count.
   Mode is significant to use for categorical variables where classifying values in terms of frequency of observation is of outmost importance.

Role of central tendencies in variability :

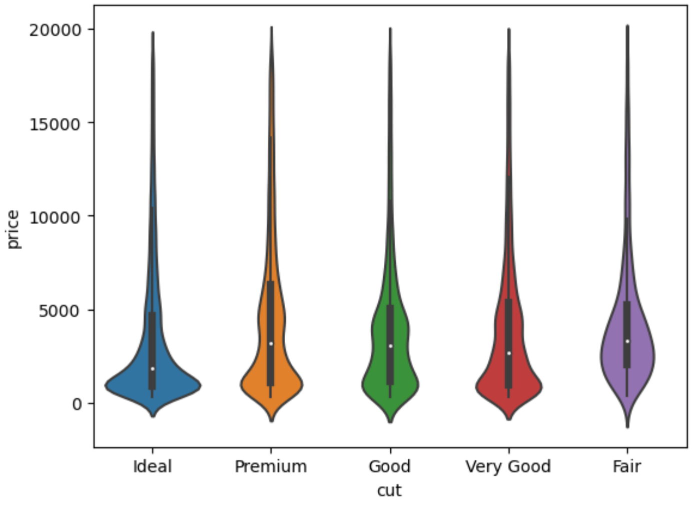
1. **Standard deviation** :
   We calculate the standard deviation of each observation with the mean of the data by summing the squares of each observation with the mean as below :

Standard deviation = sqrt ((∑ (xi – mean)²)/ n)
where, xi – all observations in a variable
n – total number of observations

Standard deviation measures by how much each observations are dispersed with respect to mean.
Low, or small, standard deviation indicates data are clustered tightly around the mean, and high, or large, standard deviation indicates data are more spread out.

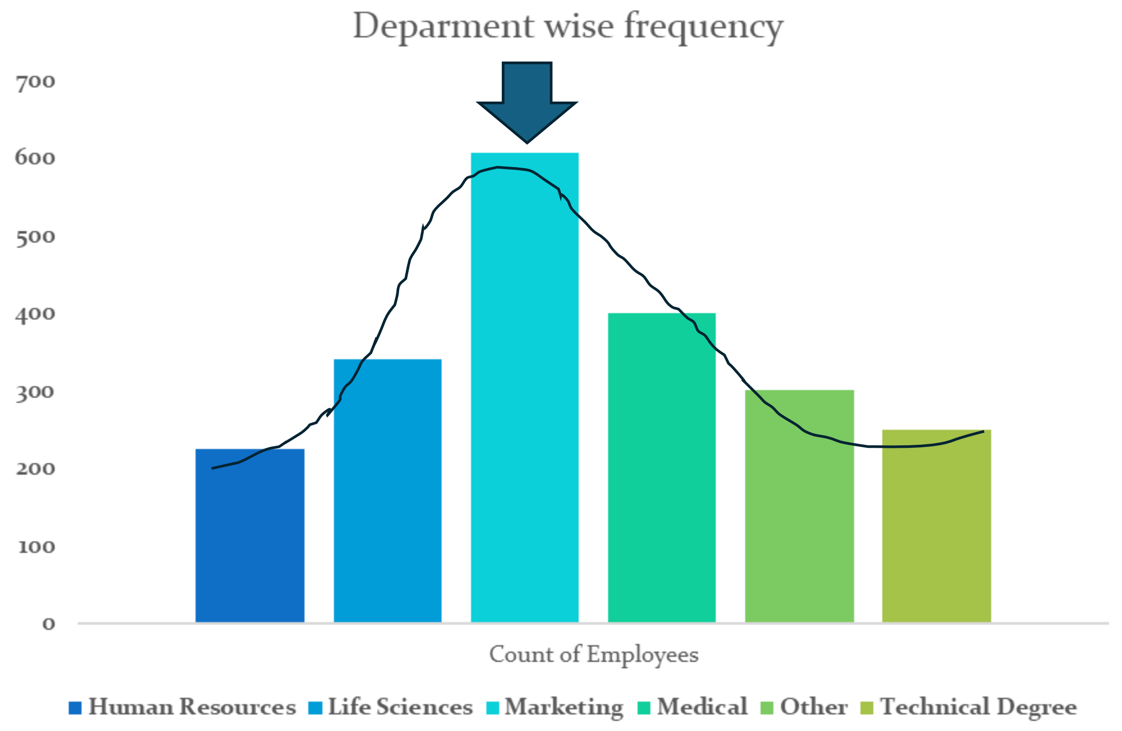
GRAPHICAL REPRESENTATION OF MEASURES OF CENTRAL TENDENCY

- ### VIOLIN PLOT



The violin Plot above is similar to a box plot which is a good representation of medina, skewness and quantiles. It also depicts distributions of numeric data for one or more groups using density curves. The width of each curve corresponds with the approximate frequency of data points in each region.

- ### Histogram:



The histogram above is a useful to tool to show dispersion amongst categorical variables (as depicted in the curve above the bars of the histogram). It also shows mode, (like above its Marketing values 600) and mean of the observations.

FREQUENCY DISTRIBUTIONS

Discrete Probability Distribution:

A discrete probability distribution describes the possibility of each potential result for a discrete random variable. In a simpler sense, it refers to circumstances in which the random variable can only accept discrete, distinct values—usually counts or integers.

- **Binomial Distribution**:
  - **Meaning**: When events have a binary result, like yes/no or success/failure, the binomial distribution represents the probability of a specific number of successes occurring across a predetermined number of independent trials.
  - **Example:** When each voter must select between two candidates, it could be used to forecast the percentage of voters who would support a specific candidate.

- **Bernoulli Distribution**:

  - **Meaning**: A single parameter, p, which represents the probability of success, characterises the Bernoulli distribution, which is a specific example of the binomial distribution in which there is only one trial and two possible outcomes: success or failure.
  - **Example:** It applies to situations where there are just two possible outcomes, such as a new-born's gender (male or female).

- **Poisson Distribution**:

  - **Meaning**: The Poisson distribution, which assumes that events occur at a constant average rate and regardless of the elapsed time since the last event, models the number of occurrences of an event within a specified span of time or space.
  - **Example:** In situations when the average rate of arrival is known, it is used to forecast how many customers will visit a store in an hour.

  - **Continuous Probability Distribution**:

  The probabilities of potential values for a continuous random variable are expressed by a continuous probability distribution. In this scenario, the random variable, which is usually represented by real numbers, can have any value within a given range, which may be infinite.

- **Normal Distribution**:

  - **Meaning**: The normal distribution, sometimes called the bell curve, is typified by a symmetric, bell-shaped curve around which data is grouped around a central mean value. The standard deviation of the distribution defines its predicted spread.
  - **Example:** It is frequently applied in statistics to simulate real-world phenomena when data tends to cluster around a mean value and most observations fall within that range, such as height, weight, or test scores.

- **Exponential Distribution**:

  - **Meaning**: In a Poisson process, when events happen continuously, independently, and at a constant average rate, the time interval between events is modelled by the exponential distribution.
  - **Example**: It's used to forecast how long it will take for the next client to show up at a service location, such a store or call centre.

- **Uniform Distribution**:

  - **Meaning**: A situation where all possible outcomes fall inside a given range have an equal probability is described by the uniform distribution.

- **Example**: Imagine a fair six-sided dice roll, where every face (number one through six) has an equal chance of showing up. With an identical chance of occurring for every outcome (number on the dice), this is a classic example of a discrete uniform distribution.

  Let X represents the result of rolling the dice. In terms of mathematics, X has a discrete uniform distribution: P(X=k)= 1/6 for k=1,2,3,4,5,6.

This indicates that since there are 6 alternative outcomes and each is equally likely, the probability of rolling any particular number, like rolling a 3, is 1/6.

DATA CLEANING AND WRANGLING

Purpose: Cleaning a new dataset is crucial to have consistent data since data quality directly affects the insights you can draw and the effectiveness of any models you build.

Steps for Data Cleaning :

1. **Handling Missing Values:**
   We have used the is.na() to check null values in all the dataset variables.   This is important because Null values will create voids in the data and make it difficult to interpret results and provides inaccurate predictions.
   After running the code in Python, we found no null values in both the dataset.

2. **Handling Duplicates**: Duplicate entries add redundancy and can distort statistical analyses, leading to misleading conclusions. Suppose the dataset contains duplicate entries for the same customer's oil consumption.
   We have used the duplicated () for detecting duplicate entries and after running the code in Python, we found no duplicate values in both datasets.

3. **Dropping Variables**: Since all columns in a dataset do not contribute meaningful information for analysis or modeling, we need to drop columns to keep only relevant features needed during modeling.
   For instance, the diamond dataset includes an index column that had no use for further data analysis, hence we dropped it as part of the data cleaning.

   Similarly, employee dataset had columns like Employee count, Over18, Stockoptionlevel, Daily rate, and Standard hours columns which are not needed for the predictive model, so these were cleaned up from the dataset.

4. **Rename Columns:** Original column names may be unclear, ambiguous, or non-descriptive. Renaming them makes the data easier to understand and work with.

   In our dats sets, the diamond dataset had x, y, and z columns that we changed it with length_mm, width_mm, and depth_mm respectively for better interpretation.

5. **Missing Value**: Missing values can introduce bias and lead to inaccurate conclusions if not addressed properly. Data cleansing techniques like imputation (filling in missing values with estimated values) or deletion (removing rows or columns with too many missing values) can help mitigate the impact of missing data.
   For example, for the diamond dataset, length_mm, width_mm, and depth_mm had minimum values as '0' for some of the observations which is practically not possible. So, we dropped the observation or the row using the drop() function in Python.

   Below is the summary of the data cleaning we did for both datasets:

S. No	Detected	Action
Diamond Dataset:		
Missing values or Null	No	NA
Duplicated values	No	NA
Drop column	Yes (Index)	dropped
Rename Column	Yes (x, y, z)	renamed
Employee Attrition Dataset:		
Missing values or Null	No	NA
Duplicated values	No	NA
Drop column	Yes (Employee count, Over18, Stockoptionlevel, Daily rate, and Standard hours)	dropped

DESCRIPTIVE DATA MINING TECHNIQUES

Data Mining techniques are used to identify patterns, relationships, and insights withing datasets without making predictions.

Here, in our case we have used one of the data mining techniques which is clustering an **unsupervised learning technique** to find the insights of the data set.

Clustering – Clustering is the task of dividing the unlabeled data or data points into different clusters such that similar data points fall in the same cluster than those who differ from the others. In a simpler way, clustering is the process of segregating groups with similar traits and assigning them into clusters.

There are various clustering types and various clustering algorithms as well. However, in our case we have used the K-Means clustering which is based on **"Centroid-based or Partition Clustering"** which works on the closeness of the data points to the chosen central value.

K-Means Clustering Algorithm –

K-Means clustering is a partition-based clustering technique that uses the distance between the Euclidean distances between the points as a criterion for cluster formation.

For instance – Let's say, there are "n" numbers of data objects, K-Means groups them into a predetermined "k" number of clusters.

About the dataset – The diamonds data set which we have used for the Regression Analysis, we have used for clustering as well.

There were 53940 rows and 11 features in the dataset itself as explained earlier in the data exploration. However, we have just taken a sample of 2500 data and three features **"carat", "cut", and "price".**

Python Libraries used for the cluster analysis –

1. Scikit-Learn – Provides machine learning algorithms and tools for data preprocessing
2. Train-Test split – To split the data set into train and test
3. K-Means – For performing K-Means clustering.
4. StandardScaler – For standardizing features
5. LabelEncoder – For encoding categorical variables
6. silhouette_score – metrics to calculate the silhouette score

Rest, the other libraries are same which we have explained above, like NumPy, matplotlib, seaborn, pandas, etc.

Data Preprocessing – In data preprocessing, we checked for the null values, duplicates, and cleaning. However, we have taken three features **"carat", "cut", and "price"** and a sample of 2,500 data points for clustering analysis. Rest other features we dropped by using drop (). Moreover, we performed the encoding of the categorical variable in this step.

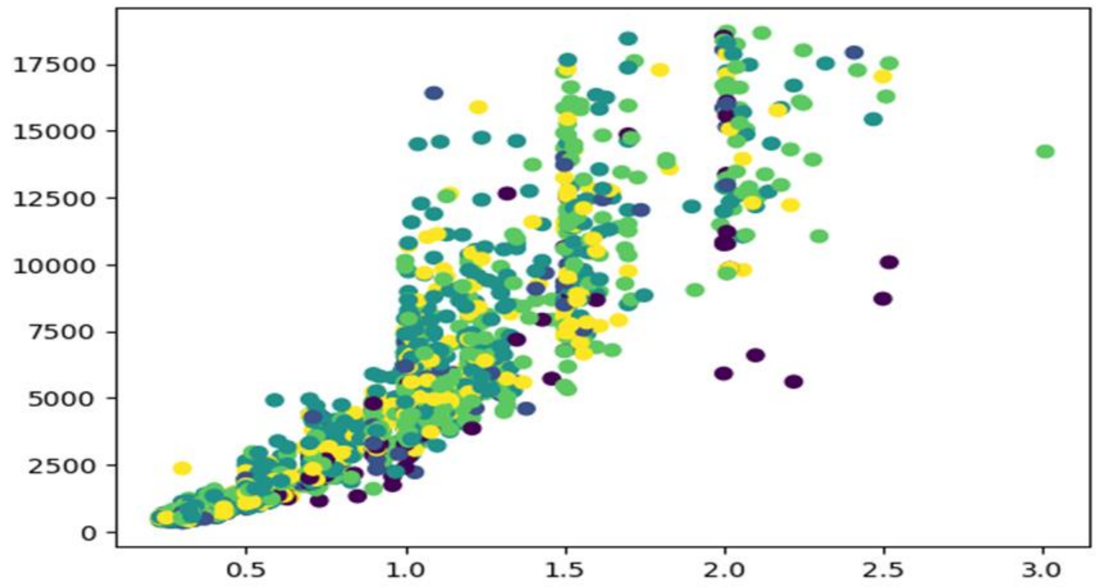
1. Encoding the categorical variable **"cut"** by using the LabelEncoder.

Data Modelling – Since we have already specified earlier, that we have used the K-Means clustering therefore we will be proceeding with following steps in the data modelling –

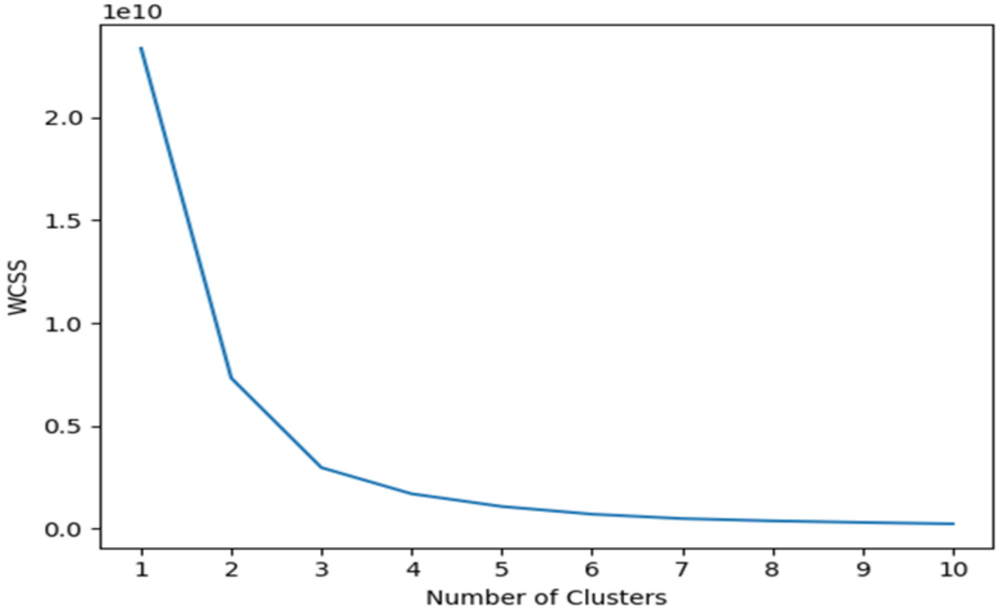
1. Divide the data – Divide the data into X and y.

2. Train-Test Split – By using the train-test split () we split the data set into train (67%) and test (33%) of the total data set.

3. Standardized the data – Standardized the input features by using the StandardScaler () as it's necessary since we are the distance based metrices algorithms. Below is the cluster plot of the observations before performing k-means clustering.



4. **Optimal K- Value –** We have used the elbow method to find the optimal K-Value as shown below. In this method, we are varying the number of clusters (K) from 1 to 10. For each value of K, we are calculating WCSS (Within-Cluster sum of squares). WCSS is the sum of the squared distance between each point and the centroid in a cluster. When we plot the WCSS with the K-Value, the plot looks like an elbow. As the number of clusters increases the value of WCSS will start to decrease. We can see from the plot below that after 5 the line in the graph moves parallel to x-axis. The K value corresponding to this point is the optimal value of K or an optimal number of clusters.

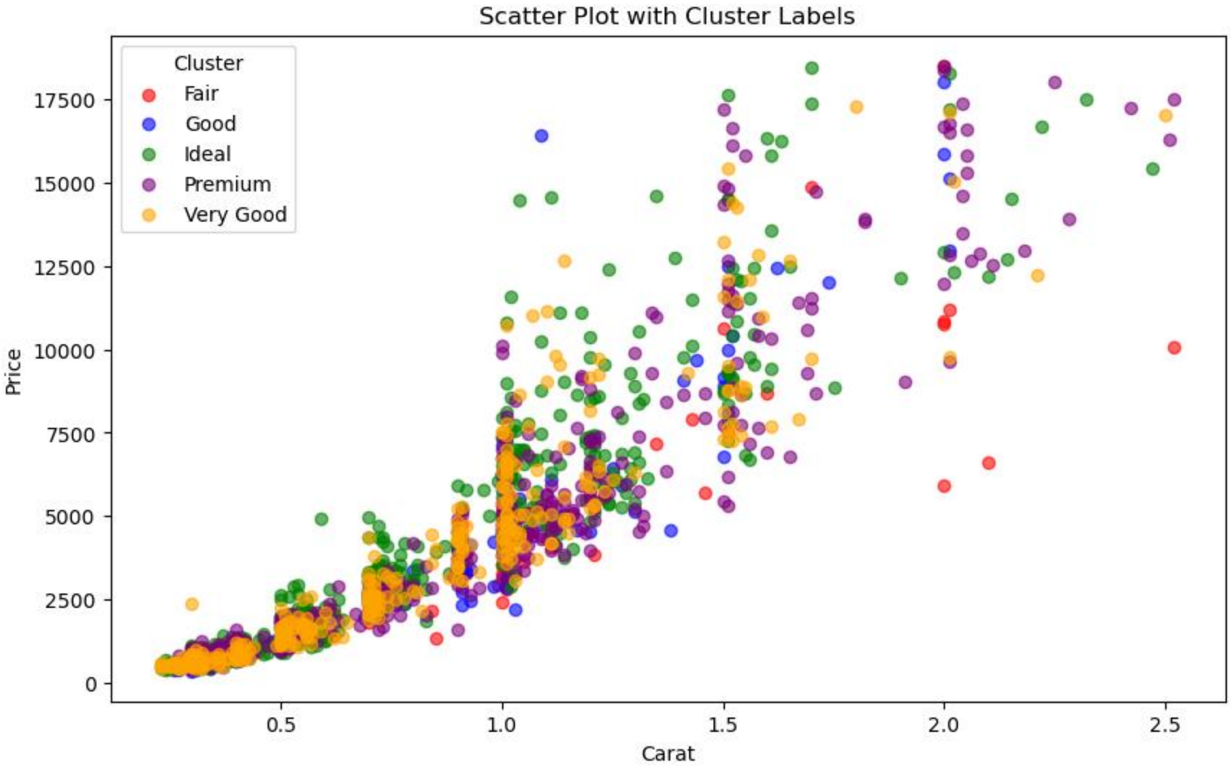


Note - We have used the **Silhouette coefficient** to calculate the goodness of the clustering. Please refer to the code file.

Silhouette Score – It is a metric used to check the performance of the clustering technique, and its value ranges between -1 to 1. In our case when we checked the score, we got .60 with 5 clusters. However, it was .46 with 3 clusters, which can be seen clearly above in the plot. Therefore, we considered 5 clusters as we know that the value of Silhouette score is toward 1 is considered better.

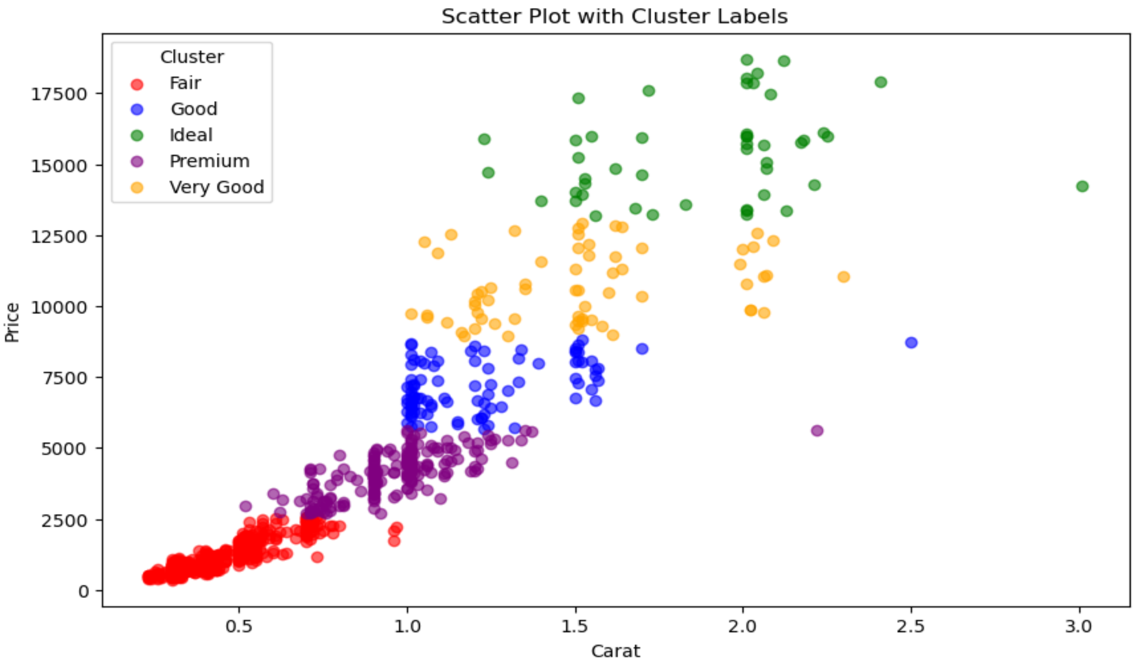
5. Train the model – After importing the K- Means from sklearn, will fit the model on scaled_X data which we standardized earlier.

Moreover, the plot below is an illustration of how the clusters are labeled into different categories against **price** on vertical axis and **carat** on horizontal axis.



6. Test the model – In this phase, we test the model on X_test data.

The plot below clearly shows how each category has formed into 5 different clusters:



TEXT MINING ANALYSIS

Text Mining – It is also one of the data mining techniques which we perform for the transformation of unstructured text data into structured format, so that we can identify the meaningful patterns or insights out the data and further can use them for various analysis as per the business needs.

Text Data Source – URL - https://www.collingwoodthejeweller.co.uk/all-what-you-need-to-know-about-diamond/understanding-the-factors-that-influence-diamond-pricing#:~:text=In%20conclusion%2C%20the%20pricing%20of,reputation%20must%20not%20be%20underestimated

Library & Functions for the Text Mining –

1. Spacy – We have used the spaCy instead of NLTK in our case for Text Analysis.
2. Wordcloud – for visual representation of the text.

1. Data Preprocessing: Text Extraction – We extracted the text and saved it in .txt format from the article related to one of our data sets that is diamond. There was another technique which could have been used, **web scrapping** by importing BeautifulSoap Library in python. However, we followed the steps we have been taught so far.

Following the extraction of the text data. The text data stored in the file "Text Mining.txt" was read and processed using spaCy's natural language processing capabilities.

2. Text Tokenization and Lemmatization –

Tokenization – It breaks down the text into individual words or tokens.

Lemmatization – It reduces the words into their base form.

Following the extraction of text data from the article related to our diamond dataset and its processing using spaCy, we performed tokenization and lemmatization to prepare the text for further analysis.

The **Counter** class from the Python Standard Library is then used to count the occurrences of each lemma, resulting in a dictionary-like object (word_counts) containing word frequencies. This stage outlines the steps taken for text preprocessing, specifically focusing on tokenization, lemmatization, and word frequency counting using the Counter class.

3. Identifying Verbs, Pronouns, and Nouns -

We utilized spaCy's linguistic annotations to identify verbs, pronouns, and nouns within the text data. Verbs represent actions or processes, pronouns are used to refer to entities, and nouns denote objects, concepts, or entities mentioned in the text.

Verbs: We filtered out tokens identified as verbs based on their part-of-speech (POS) tags.

Pronouns: Similarly, we filtered out tokens identified as pronouns based on their POS tags.

Nouns: Additionally, we filtered out tokens identified as nouns based on their POS tags.

4. Named Entity Recognition (NER)

Named Entity Recognition is a text processing technique used to identify and classify named entities such as persons, organizations, locations, dates, and more within a text document.

Here are some examples from our data for illustration for Named Entity Recognition –

Organizations: "The Gemological Institute of America" (GIA)

Products: "diamond"

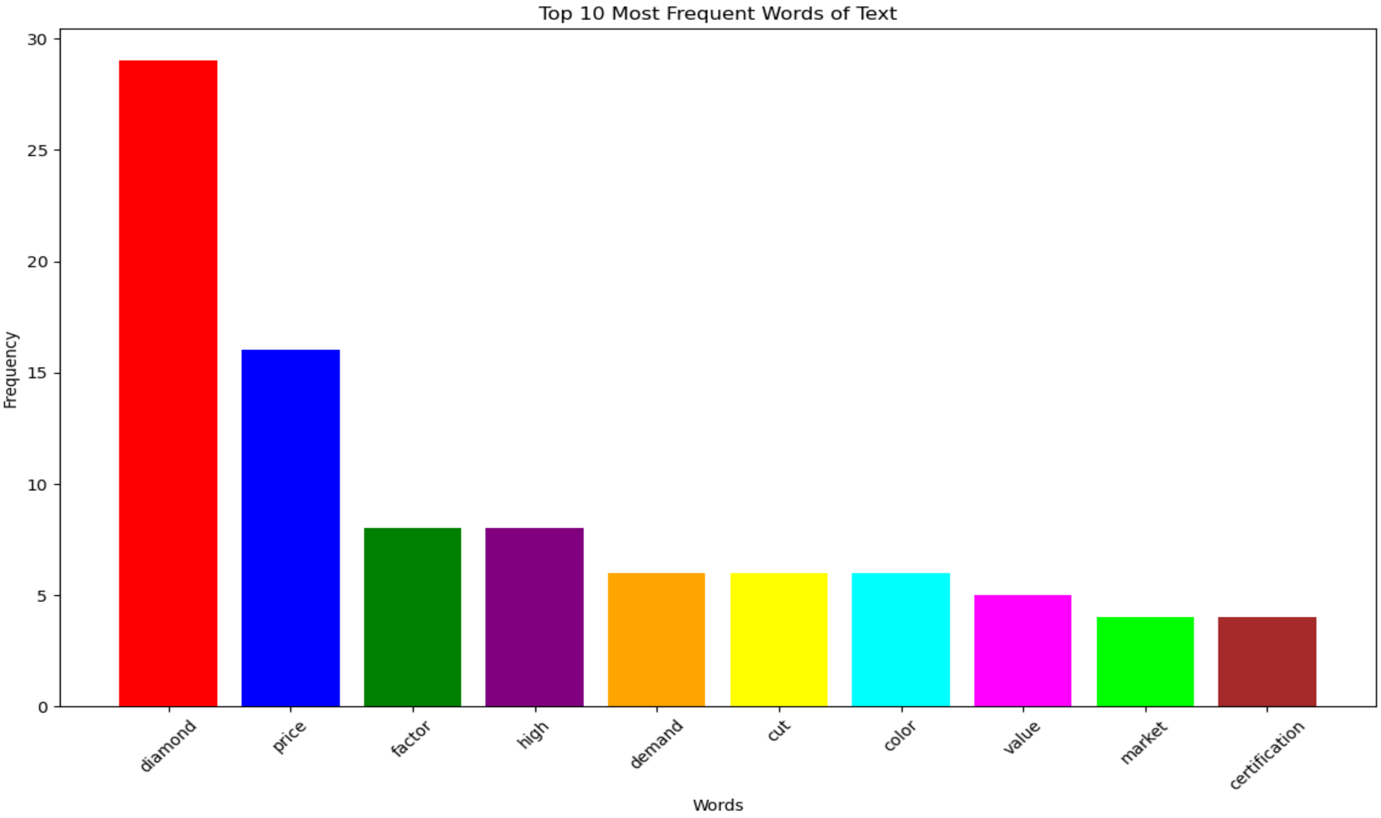
Events: "understanding the factors that influence diamond pricing"

5. Visualization –

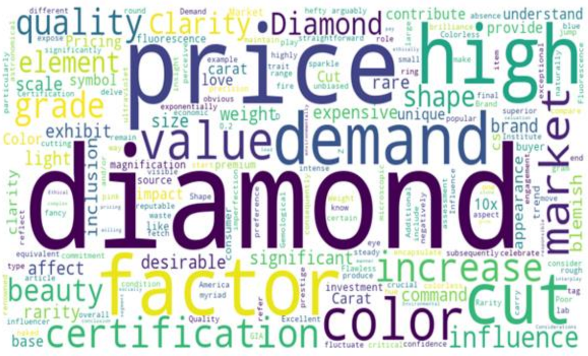
After preprocessing the text data and filtering it out, we identified the top 10 most frequent words. This analysis provides insights into the most common terms used in the text data.

Histogram and Word Cloud - To visualize the frequency distribution of the top 10 most frequent words, we created a histogram and a word cloud.

Histogram: A histogram provides a visual representation of the frequency distribution of words. Each bar in the histogram represents the frequency of a specific word.



Word Cloud: A word cloud visually represents the frequency of words in a text document, where the size of each word corresponds to its frequency. The more frequently a word, the larger it appears in the word cloud.



PREDICTIVE DATA MODELING

There is a wide array of data mining techniques used in data science and data analytics. Your choice of technique depends on the nature of your problem, the available data, and the desired outcomes. Predictive modeling is a fundamental component of mining data and is widely used to make predictions or forecasts based on historical data patterns. You may also employ a combination of techniques to gain comprehensive insights from the data.

Here, we use different types of data mining techniques

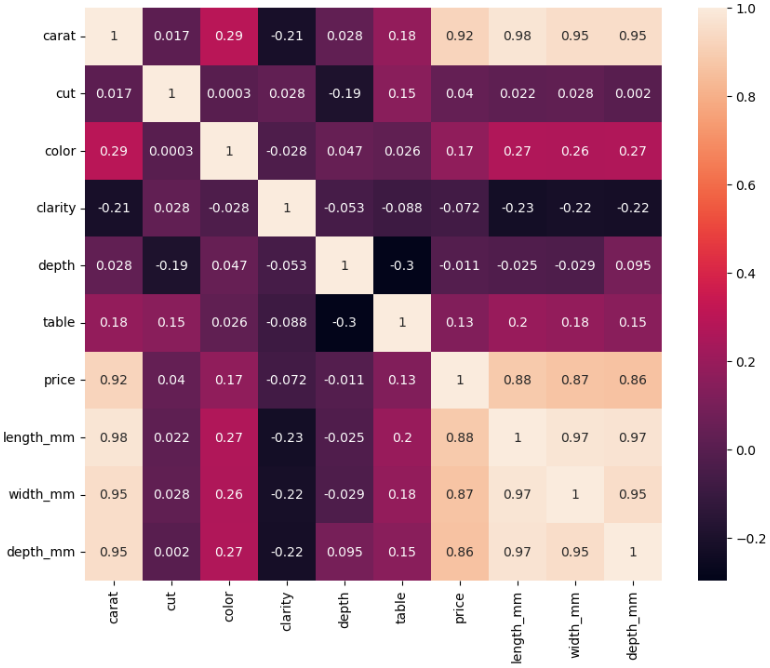
1. Regression Analysis

2. Classification

Regression Analysis:

Linear Regression: Linear Regression is a part of Supervised Learning Technique which is first most basic part of Machine Learning. So, Linear Regression can be called as first most Machine Learning algorithm. It is mostly used to quantify the correlation among the single or multiple predictive and responsive variables.

Model Selection: Linear Regression Model For Diamonds dataset.



DIAGNOSTIC PLOTS:

Interpretation :
The above correlation matrix shows Price vs. length_mm, width_mm, and depth_mm are highly correlated. In other words, If any changes made in diamond dimensions are highly affect in diamond price.

Model Assumptions:

1. There is a linear association between Price and Predictors (carat, cut, color, clarity, table, depth, length, and width.
2. Diamond dimensions (x, y, z) have a better explanation of diamond price and seems to have a larger impact on low vs high diamond price.

The linear fit model can be expressed as follows:

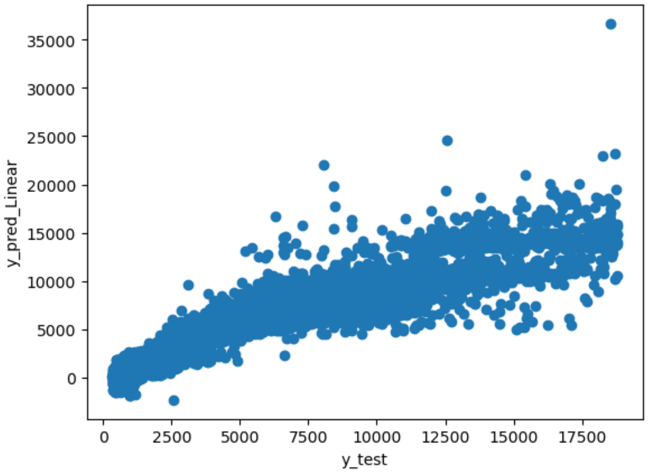
Price = β0 + β1·carat + β2·cut + β3·color + β4·clarity + β5·table + β6·depth% + β7·length_mm + β8·depth_mm + β9·width_mm + ϵ

Where:

- β0 is the intercept term.
- β1, β2, β3, … are the coefficients of the respective independent variables.
- ϵ is the error term.

Let's talk about model building process:

- Splitting Dataset into a Training dataset and Test dataset. To do this, we'll need to import the function train_test_split from the model_selection module of scikit-learn. For diamond dataset we split our data into 75:25 ratio in training and test dataset.
- Building and training the model, The first thing we need to do is import the LinearRegression estimator from scikit-learn.
- Making Prediction for our model, scikit-learn makes it very easy to make predictions from a machine learning model using the predict() function. Using the test dataset and predicted model, draw a scatterplot to make it linear.



- Testing the performance of our model, Using metrics from sklearn we can find 1. Adjusted R^2 2. Mean Absolute Error 3. Root Mean Squared Error.

  Adjusted R^2: Adjusted R^2 is a corrected goodness-of-fit (model accuracy) measure for linear models. It identifies the percentage of variance in the target field that is explained by the input or inputs.

R^2 tends to optimistically estimate the fit of the linear regression. It always increases as the number of effects are included in the model. Adjusted R^2 attempts to correct for this overestimation. Adjusted R^2 might decrease if a specific effect does not improve the model.

Adjusted R^2 is always less than or equal to R^2. A value of 1 indicates a model that perfectly predicts values in the target field. A value that is less than or equal to 0 indicates a model that has no predictive value. In the real world, adjusted R^2 lies between these values.

Mean Absolute Error (MAE): Mean Absolute Error (MAE) is a simple, yet powerful metric used to evaluate the accuracy of regression models. It measures the average absolute difference between the predicted values and the actual target values.
Unlike other metrics, MAE doesn't square the errors, which means it gives equal weight to all errors, regardless of their direction. This property makes MAE particularly useful when you want to understand the magnitude of errors without considering whether they are overestimations or underestimations.

Root Mean Squared Error (RMSE):

The root mean square error (RMSE) measures the average difference between a statistical model's predicted values and the actual values. Mathematically, it is the standard deviation of the residuals. Residuals represent the distance between the regression line and the data points.

RMSE quantifies how dispersed these residuals are, revealing how tightly the observed data clusters around the predicted values. As the data points move closer to the regression line, the model has less error, lowering the RMSE. A model with less error produces more precise predictions.

RMSE values can range from zero to positive infinity and use the same units as the dependent (outcome) variable.
After Evaluating our model, we get

Metrices	Value
Adjusted R^2	0.8874
Mean Absolute Error	858.27
Root Mean Squared Error	1337.57

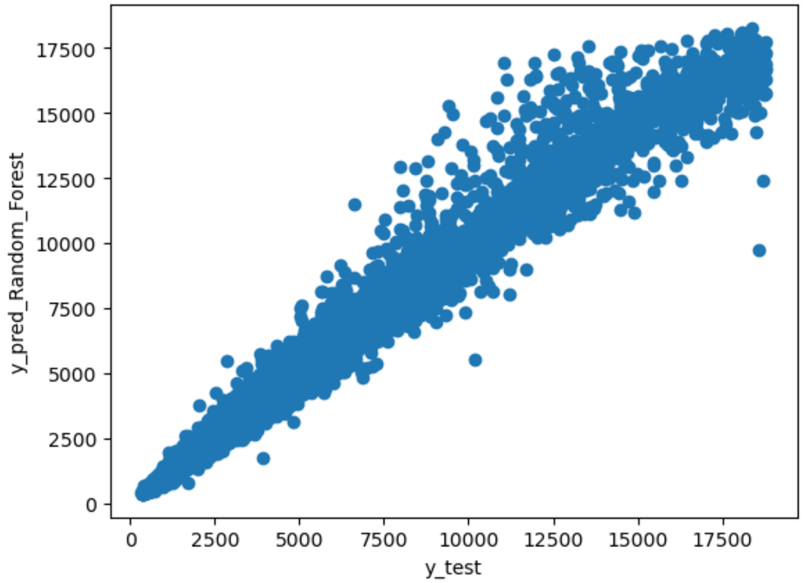
Random Forest Regressor: Random Forest Regressor is a powerful tree-learning technique in Machine Learning. It works by creating several Decision Trees during the training phase. Each tree is constructed using a random subset of the data set to measure a random subset of features in each partition. This randomness introduces variability among individual trees, reducing the risk of overfitting and improving overall prediction performance.

In prediction, the algorithm aggregates the results of all trees, either by voting (for classification tasks) or by averaging (for regression tasks) This collaborative decision-making process, supported by multiple trees with their insights, provides an example of stable and precise results. Random forests are widely used for classification and regression functions, which are known for their ability to handle complex data, reduce overfitting, and provide reliable forecasts in different environments.

Model Selection: Random Forest Regressor

Let's talk about model building process:

- Splitting Dataset into a Training dataset and Test dataset. To do this, we'll need to import the function train_test_split from the model_selection module of scikit-learn. For diamond dataset we split our data into 75:25 ratio in training and test dataset.
- Building and Training the model, The first thing we need to do is import the RandomForestRegressor estimator from scikit-learn.
- Making Prediction for our model, scikit-learn makes it very easy to make predictions from a machine learning model using the predict() function. Using the test dataset and predicted model, draw a scatterplot to make it linear.



- Testing the performance of our model, Using metrics from sklearn we can find 1. Adjusted R^2 2. Mean Absolute Error 3. Root Mean Squared Error.

  After Evaluating our model, we get

Metrices	Value
Adjusted R^2	0.9803
Mean Absolute Error	268.84
Root Mean Squared Error	558.26

- **Model Comparison:**

  Adjusted R^2:

Higher Adjusted R² indicates a better fit of the model to the data. The random forest regressor has a significantly higher Adjusted R², suggesting it explains more of the variance in the target variable than the linear regression model.

Mean Absolute Error (MAE):

Lower MAE indicates better performance as it measures the average magnitude of the errors in a set of predictions, without considering their direction. The random forest regressor has a much lower MAE, indicating better predictive accuracy.

Root Mean Squared Error (RMSE):

Lower RMSE indicates better performance as it measures the square root of the average of the squared differences between predicted and observed values. The random forest regressor has a much lower RMSE, indicating better predictive accuracy.

Based on the given evaluation metrics, the **Random Forest Regressor** outperforms the linear regression model in all aspects. It has a **higher** Adjusted R², and significantly **lower** MAE and RMSE values, indicating that it provides a better fit to the data and more accurate predictions.

Therefore, the random forest regressor is the better model among the two based on the provided metrics.

Classification Modeling

Classification is a type of data analysis where our outcome variable is qualitative based on features that relevant for classification. As discussed earlier , we have two type of qualitative variables :

1. **Nominal variables**: these types of variables do not have any natural ordering of the classes it identify based on categories. Example, Gender(Male/Female) , Yes/No type variables, etc.

2. **Ordinal variables:** variables that have natural ordering similar to assigning a rank based on the severity of the observations in a variable. Example, “Education” has ranked values 1 -5 , which 1 being pass out from high school and with 5 being a highly educated person.

As part of this report, we are using two classification models :

1. **Logistic Classifier**: this type of classification is done in terms of log odds of an event happening or in other words, it’s a classification with the log of outcome is predicted based on the conditional probability (using maximum likelihood function) for classifying outcome in given categories given input observations of the predictors are true.

2. **<u>Ridge Classifier:</u>** This type of classification is used for multi-class classification problems by using a method called L2 regularization. In statistical terms, L2 regularization is method used to penalize the model for using an increasing number of predictors by adding a mean square loss function as shown below :

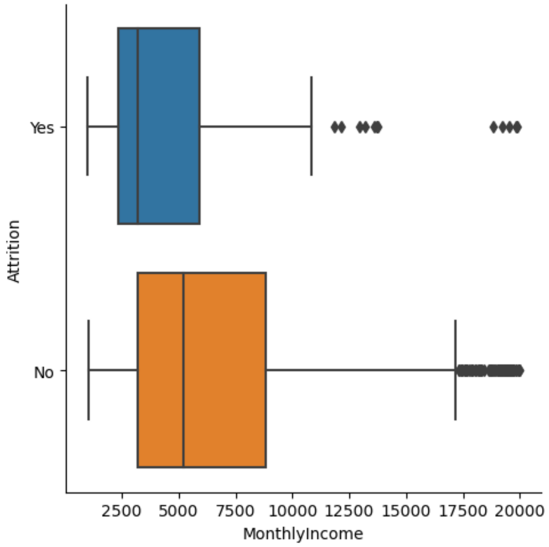
$$\mathbf{Lossridge = LossOLS + \lambda \sum(j=1\ to\ p)\ \beta j2}$$

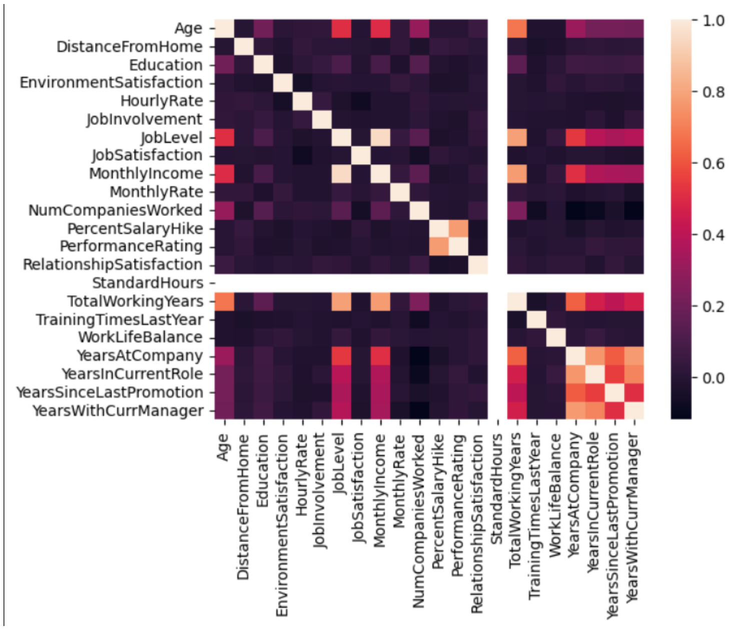
Here, **λ (**lambda) is what we call a hyperparameter, which is chosen to keep a balance between bias and variance by controlling the strength of regularization. It controls overfitting a **βj^2 term** which is nothing but sum of squared coefficients that we use similar to in a linear regression problem.

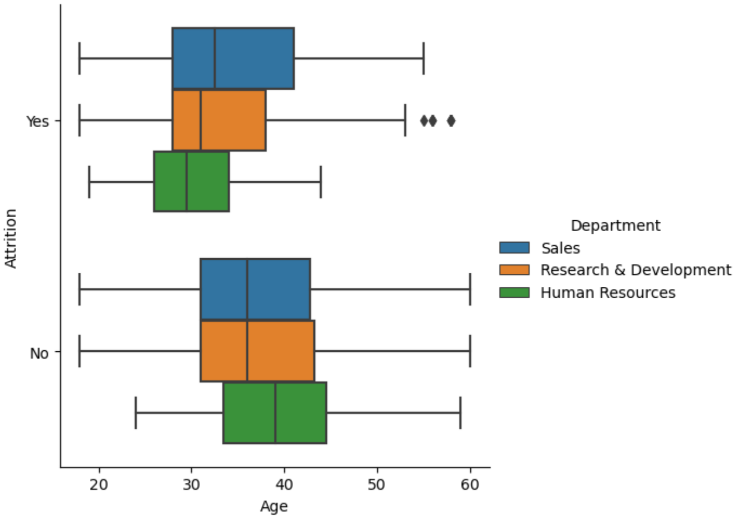
<u>Below are the steps for classification that we follow</u> :

a. Perform diagnostics to remove to keep relevant features that impact the classification outcome
b. Split the dataset in the training and testing set with 60% of the observations as the training test and rest 40% as the testing set.
c. Convert any categorical variables as dummies or ranked-based categories to make them suitable for classification as ordinal values do not support model fit for the classifier.
d. Fit the classification model with outcome and predictors.
e. Run predictions with the corresponding classification model fitted in step c)
f. Create a confusion matrix to determine the accuracy measures of the model.
g. Compare which model is better based on the accuracy and F1 scores

DIAGNOSTIC PLOTS:







Model Assumptions and Data interpretation:

- From the heatmap, we can remove one of the variables from the pair which are highly correlated since it would not make sense to keep both in pair to reduce bias in the modeling
- Total working years-
  We will choose Age and remove Total working years since it's evident that people with more age in the organization will have more working years as well, especially in case of managers or people at leadership positions.

- Job level - We will remove the job level and take monthly income since income increases with an increase in job level and job level might not impact Attrition since there can be cases where employees are in good positions but are not satisfied with their salary and hence salary becomes more contributing to Attrition.

- Years in Current Role, Years since last promotion, and Years with Current Manager. We will remove all these since all of these are highly correlated to "Years at Company" which should be sufficient to include as our predictor instead of keeping all four.

Below are the outcome and predictors used in the classification:

Outcome Variable: Attrition

Predictors: ['Age', 'BusinessTravel', 'DailyRate', 'Department', 'DistanceFromHome',
'Education', 'EducationField', 'EmployeeCount',
'EnvironmentSatisfaction', 'Gender', 'HourlyRate', 'JobInvolvement',
'JobLevel', 'JobRole', 'JobSatisfaction', 'MaritalStatus',
'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked', 'Over18',
'OverTime', 'PercentSalaryHike', 'PerformanceRating',
'RelationshipSatisfaction', 'StandardHours', 'StockOptionLevel',
'TotalWorkingYears', 'WorkLifeBalance', 'YearsAtCompany',
'YearsInCurrentRole', 'YearsSinceLastPromotion',
'YearsWithCurrManager']

After splitting the datasets and creating dummy variables for the qualitative predictors, below is how we fit classifiers on the training set:

1) LogisticRegression(random_state=0).fit(X_train, y_train)
2) RidgeClassifierCV(alphas=[0.05,0.1,0.5],cv=5).fit(X,y)

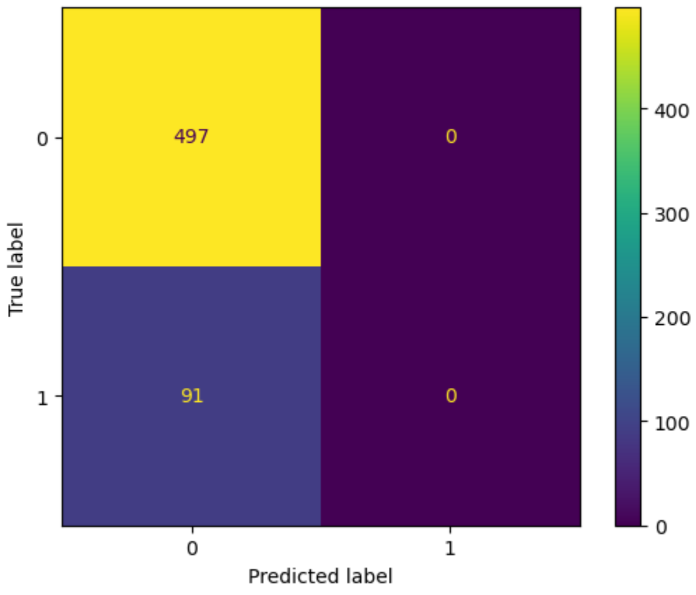
Interpretation of Ridge Classifier:

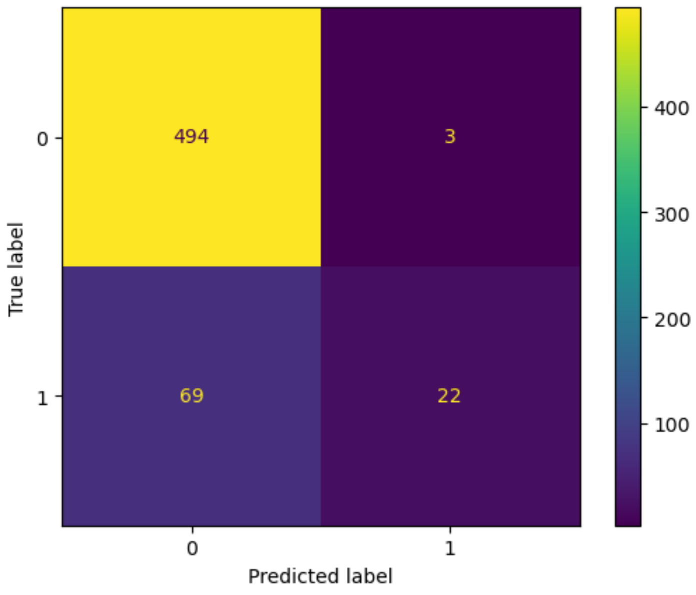
- **Choice of hyperparameter** alpha as per syntax**:**

Here we have considered different values of alpha that the fitted model will take and choose the best regularization strength out of these alpha values to perform data conditioning. This will also reduce the variance amongst the estimates using a suitable regularization strength.

- **Cross-validation of train and validation sets**

We used 5-cross validation by using cv=5 inputs that will choose the optimal fitted model and compare with all alpha values to select the final optimal model. The cross-validation technique helps to reduce overfitting data by dividing the dataset randomly into 5 folds and iteratively selecting one of the folds as the validation set and the rest k -1 (4) folds as the training set. Based on the number of observations in the data, usually cv value of 5 does a decent job.





Logistic Confusion matrix

Ridge Confusion matrix

INTERPRETATION OF CONFUSION MATRIX PLOT

- **True Label** – Here true label signifies the actual observations that have 0 and 1 values (corresponding to **No** and **Yes** responses) of the Outcome variable, **Attrition**.

- **Predicted Label -** Here predicted label signifies the actual observations that have 0 and 1 values (corresponding to No and Yes responses) of the Outcome variable, **Attrition**

➢ **Below is the meaning of different results in the above grids of the confusion matrix plots for both classification models:**

1. **True Label = 1 and Predicted Label = 1**
   This represents the total number of True Positives, or in other words, the number of observations that have the actual value of Attrition as “Yes” that the model has predicted outcomes as “Yes” as well.

2. **True Label = 0 and Predicted Label = 0**
   This represents the total number of True Negatives, or in other words, the number of observations that have the actual value of Attrition as “No” that the model has predicted outcomes as “No” as well.

3. **True Label = 0 and Predicted Label = 1**

This represents the total number of False Positive, or in other words, the number of actual observations with Attrition as "No" that the model has predicted outcomes as "Yes" as well.

4. **True Label = 1 and Predicted Label = 0**
   This represents the total number of False Negative, or in other words, the number of observations that have the actual value of Attrition as "Yes" that the model has predicted outcomes as "No" as well.

Logistic confusion matrix summary

```
              precision    recall  f1-score     support

          No       0.85      1.00      0.92       497
         Yes       0.00      0.00      0.00        91

    accuracy                           0.85       588
   macro avg       0.42      0.50      0.46       588
weighted avg       0.71      0.85      0.77       588
```

Ridge confusion matrix summary

```
              precision     recall  f1-score   support

          No       0.88      0.99      0.93       497
         Yes       0.88      0.24      0.38        91

    accuracy                           0.88       588
   macro avg       0.88      0.62      0.66       588
weighted avg       0.88      0.88      0.85       588
```

INTERPRETATION OF CONFUSION MATRIX SUMMARY TERMS

1. **Accuracy**

   Accuracy is the metric in classification prediction is the ratio of the number of observations that were correctly classified by a classifier to the total number of observations in the test data.

2. **Precision**

   Precision is the metric in classification prediction is the proportion of observations predicted to be "Yes" for Attrition by a classifier that were actually as "Yes".

3. **Recall**

Recall is the metric we use to describe the ability of a classifier to correctly predict “Yes” as an outcome for Attrition for (positive) observations.

In other words, Recall can be expressed as, ratio of below :

(no. of obs correctly classified as Yes) and ((no. of obs correctly classified as Yes) + (no. of true obs misclassified as No))

4. **F1-score**
   F1 score is the metric that is used to combine and recall and precision and take the average score to calculate the actual accuracy of the classification algorithm

CLASSIFICATION PREDICTIONS CONCLUSION :

For the HR dataset, **Ridge Classifier algorithm has the best accuracy rate of 88% and also has the best F1 score of 85%** which is also the true accuracy of the Ridge classification model.

On the other hand, the Logistic Classifier algorithm has an accuracy of 85% and has an F1 score of 77%.