A picture containing icon

Description automatically generated

**CCT College Dublin**

|  |  |
| --- | --- |
| **Module Title:** | Data preparation and Visualization, Programming for Data Analytics, Statistics for Data Analytics, Machine learning for Data Analytics |
| **Assessment Title:** | MSc DA\_CA2 |
| **Lecturer Name:** | Sam Weiss, John o Sullivan, Muhammad Iqbal, David |
| **Student Full Name:** | Muhammad Ahsan |
| **Student Number:** | 2023132 |
| **Assessment Due Date:** | 26/05/2023 |
| **Date of Submission:** | 14/05/2023 |

# Group ID - MSc in Data Analytics

Author: Muhammad Ahsan

E-mail: [2023132@student.cct.ie](mailto:2023132@student.cct.ie)

Student ID: 2023132

**Table of Content**

[Group ID - MSc in Data Analytics 1](#_Toc136013425)

[Abstract: 3](#_Toc136013426)

[Introduction: 3](#_Toc136013427)

[Dataset Information: 4](#_Toc136013428)

[**USA\_construction\_cost\_of \_healthcare\_project:** 4](#_Toc136013429)

[**Total Production in Building and Construction Sector:** 4](#_Toc136013430)

[**Hotel dataset:** 5](#_Toc136013431)

[Project Management Framework: 6](#_Toc136013432)

[Exploratory Data Analysis: 9](#_Toc136013433)

[**Data Cleaning:** 9](#_Toc136013434)

[Data Visualization: 12](#_Toc136013435)

[Graphical User Interface Graph with Tkinter: 17](#_Toc136013436)

[Hypothesis testing: 20](#_Toc136013437)

[**Two-sample T-test:** 21](#_Toc136013438)

[**Two-sample KS-test:** 22](#_Toc136013439)

[**Two pair T-test:** 23](#_Toc136013440)

[**Chi-square test:** 24](#_Toc136013441)

[**ANOVA (Analysis of Variance):** 24](#_Toc136013442)

[Model Building and Evaluation: 26](#_Toc136013443)

[Machine Learning Algorithms: 28](#_Toc136013444)

[**Random Forest Regressor:** 28](#_Toc136013445)

[**Linear regression:** 30](#_Toc136013446)

[**Decision Tree:** 32](#_Toc136013447)

[Hyperparameter Tuning: 34](#_Toc136013448)

[**Grid Search:** 34](#_Toc136013449)

[**Random Search:** 34](#_Toc136013450)

[Sentimental Analysis on Hotel Reviews: 38](#_Toc136013451)

[Conclusion: 41](#_Toc136013452)

[References: 41](#_Toc136013453)

# Abstract:

This study is actually about constructional dataset which contains information related to various factors affecting construction costs, such as construction type, location, number of projects, and other relevant variables. I used two datasets “USA\_construction\_cost\_of \_healthcare\_project” and “Total Production in Building and Construction Sector”. The dataset contains a wide range of construction projects by counties, including residential, commercial, and industrial. During the EDA process, we hire a combination of statistical techniques, data visualization approaches, and exploratory data analysis tools to expansion a inclusive understanding of the dataset. Descriptive statistics, such as measures of central tendency and dispersion, are calculated to summarize the distribution and characteristics of the cost variables and also apply the hypothesis test for relate the distributions of the features. I execute Exploratory Data Analysis (EDA) on my two datasets that plays a vital role in understanding and acquisition insights from datasets in numerous domains trends that can deliver valuable insights for decision-making in building projects. Moreover, data visualization techniques, such as histograms, box plots, scatter plots, and correlation matrices, are exploited to classify patterns and relationships between the variables. Through visual exploration, we analyse the impact of different factors on constructional costs, such as, total number of projects in different counties of USA, and what the mean cost of different counties etc. Overall, this exploratory data analysis on the constructional cost dataset highlights the importance of understanding the underlying patterns and relationships in construction cost data. The insights gained from this analysis can assist project managers, contractors, and other industry professionals in optimizing cost estimation, resource allocation, and decision-making in construction projects.

# Introduction:

This project analysis is based on three different datasets related to the constructional information about USA and Ireland counties. The purpose of accompanying the EDA on the constructional cost dataset is to gain insights, expose patterns, authenticate assumptions, ensure data quality, and inform decision-making processes in the construction industry. An analyst must analyse the data to properly understand given attributes and related information to create ease in data preparation and visualizations. The research explores the key factors influencing construction cost of different countries like Ireland and USA and their impact on the real estate market. The ultimate purpose of this research is to provide treasured insights and information that can inform decision-making in the construction industry. The results from the analysis can help project managers, contractors, and other stakeholders in enhancing cost estimation, resource allocation, project planning, and budgeting. By considerate the cost drivers and variations, stakeholders can make more knowledgeable decisions to control and accomplish constructional costs effectively. This project consents us to validate expectations and hypotheses related to constructional cost analysis. It supports in confirming or disproving initial beliefs or expectations about the dataset and its variables. This can prevent potential biases and ensure that subsequent analyses are based on reliable insights. It also helps as a quality assurance step by classifying data quality issues, missing values, outliers, or inconsistencies in the constructional dataset. By talking these issues, we can guarantee the reliability and accuracy of the consequent analyses and decision-making processes.

Let’s talk about the datasets which I used in my project.

# Dataset Information:

I used three datasets for this project.

## **USA\_construction\_cost\_of \_healthcare\_project:**

This dataset has six different features and one of the columns is unusual which are not useful for EDA so I drop that column in my next steps so I take only five features and for the sake of easily accessing I change the names of the columns and the sequence you can check it on my jupyter notebook file. So here are the names of the column which I take for my project 'year', 'county', 'no\_project', 'project\_status', 'cost'. The shape of this dataset is “4544 x 4“. It means in this dataset we have 4544 rows and 6 columns/features. The five features of this dataset are independent and only one feature is dependent on the other features and we called it target feature or variable. Here is the explanation of the dataset features:

* **County**: The datatype of this column is object or categorical. This column/feature holds nominal values. It means this feature has categorical values that represent different categories without any integral order. The values in this column denotes a definite county, such as "Alameda", "Contra Costa", "Ventura" and "Yolo". These values do not have a numerical relationship between them.
* **Year:** The datatype of this column is object but I changed it into integer because it has discrete values. This column gives the idea to represent the date when the data was produced or recorded. It probable contains historical information in a date format, permitting analysis based on different time periods. But I changed this date format into only year because there is only a few unique dates in the dataset which is not really useful for time series analysis so I just get the year from this column.
* **Number of Projects (no\_project):** The datatype of this project is integer. It also has a discrete value. It represents the number of projects recorded for the specific county. It contains numerical values indicating the count of projects.
* **Project Status:** The datatype of this column is categorical. It has nominal values because values do not have a natural ordering or numerical relationship between them.This column indicates the status of the project. It has categorical values such as "In Review," "Pending Construction," "In Construction," or "In Closure". Each value represents the current stage or status of the project.
* **Cost:** The datatype of this column is float because it has continuous values. This feature signifies the total costs related with the construction projects. The values of this feature in a monetary format, such as dollars.

## **Total Production in Building and Construction Sector:**

This dataset has eight different features and three of the columns is unusual which are not useful for EDA so I drop that column in data analysis which you are going to see in the EDA section so I take only five features and for the sake of easily accessing I change the names of the columns. So here are the names of the column which I take for my project 'statistic', 'stats label', 'year', 'construction type', 'price'. The shape of this dataset is “1840 x 5“. It means in this dataset we have 4544 rows and 5 columns/features. The four features of this dataset are independent and only one feature is dependent on the other features and we called it target feature or variable. Here is the explanation of the dataset features:

* **Statistics**: The datatype of this column is object or categorical. This column/feature holds nominal values. It means this feature has categorical values that represent different categories without any integral order. The values in this column denotes a statistic. The values do not have a numerical relationship between them so that’s why its nominal values
* **Year:** The datatype of this column is integer and it has discrete values like years 2010,2011 etc. This column gives the idea to represent the year when the data was produced or recorded.
* **Statistic Label:** The datatype of this project is categorical. It has a nominal value. The values do not have a numerical relationship between them so that’s why its nominal values. This column provides the production information. I rename the values of this column because it has very unusual characters and very long description so I changed it for easily accessing.
* **Construction type:** The datatype of this column is categorical. It has nominal values because values do not have a natural ordering or numerical relationship between them.This column indicates the type of the construction like residential building, non-residential building, and civil engineering etc. Each value represents the type or status of the construction.
* **Price:** The datatype of this column is float because it has continuous values. This feature signifies the total price related with the construction projects. It shows the construction cost of the different projects

## **Hotel dataset:**

Actually, I used this dataset for sentimental analysis because I’m unable to find the reviews dataset related Ireland construction. I want to scrap reviews of construction cost related to Ireland country but I found only 5 to 10 reviews which is very less for Machine learning modelling so I found this dataset and apply sentimental analysis on it. This dataset also has six features but I only used description column and the target variables for my project. I only explain those features which I used for my task:

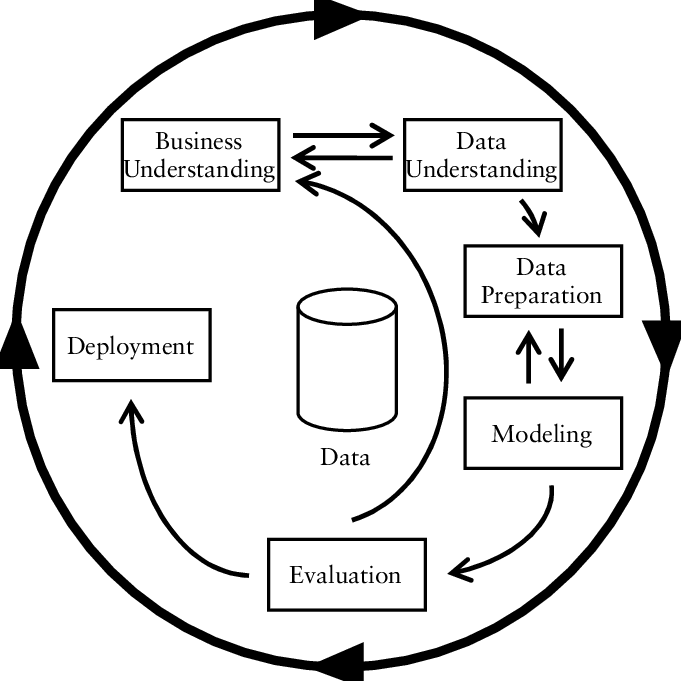
* **Description:** The datatype of this column is object (String). This column encompasses reviews related to hotel visits, expressing the sentiments of the guests. I have applied the sentimental analysis on the reviews of customer and train the machine learning model.
* **Response:** The datatype of this column is object (String). It comprises categorical values, either "happy" or "not happy" representing the general sentiment of the customer.These values do not have a natural ordering or numerical relationship between them. Each value represents a distinct category or response without any inherent rank or order.This is the target variable of the dataset. It specifies whether the customers overall experience was positive or negative.

# Project Management Framework:

I choose the CRISP (Cross Industry Standard Process) framework for my assessment because the procedure of the CRISP framework is very easy and efficient. Here are the reasons why I choose CRISP Framework:

* **Business Understanding:** This step involves analyzing and understanding the trends and patterns of constructional costs in different areas of USA. You can see in the dataset that cost of construction were very high in Los Angeles, San Francisco, Santa Clara, etc. and you can also check the cost by yearly in different counties of the USA. On behalf of that, we can make recommendations for different type of construction like residential, non- residential, civil engineering, etc. in different counties of USA and how much it cost.
* **Data Understanding**: This step includes exploring and analyzing the dataset better to understand its structure and content. It contains data visualization and statistics to identify data quality issues, missing values, or outliers. This stage will help identify relevant variables and understand the data patterns and relationships. In USA dataset, there is not no missing values or bad quality values but the names and the datatypes of the feature need work to do so I work on them and you can see the year column have dates but it’s not useful because it has very less values so I just fetch the year from that column. And I perform same action for the other datasets and make them useful for my project.
* **Data Preparation**: In this stage, I perform Exploratory data analysis and data transformation on all three datasets. I also do feature engineering on them and used only that features which have information for the modeling. I remove unusual columns from the dataset and remove the nan values from the data. Also check whether the feature from the all datasets is useful for modeling or not. I Clean and preprocess the dataset to guarantee its quality and correctness for analysis. I also transform the data types, addressing outliers, normalizing or scaling data. For example, I convert the "cost" column from a string to a numerical format. I also change the year column from string to integer. Just get year from the date column because there are less unique values of dates so that’s why I fetch only year from it.
* **Modeling**: In this step, I choose different machine learning model for training and testing on different datasets. I used linear regression, decision tree regressor, random forest regressor, hyperparameter tuning with random forest regressor, ridge regression, lasso regression on the USA dataset and get some insights of constructional cost related to different counties of USA. I also apply Sentimental Analysis on the hotel dataset which trains on the customers reviews and predict the sentiments of the customers. I used these types of techniques in modeling procedure.
* **Evaluation**: This step comprises the model performance and accuracy using fitting evaluation metrics. After training and testing the different models on different dataset, I got good accuracy of the models and also know the precision and recall. With the help of hyperparameter tuning I got good training and testing accuracy and after applying validation on them I generates the average accuracy of the model. With the help of confusion matrix, you will know the true positive and true negatives from the model predictions which shows the actual performance of the model.
* **Deployment:** In this step, I will find out the insights derived from the analysis in a clear and understandable manner. I also create visualizations, reports, or dashboards to effectively communicate the results. I have made sure that the insights are legal and can be used to make knowledgeable decisions or take suitable actions. You can also deploy the trained model and make the relevant industry's final reports, actions, and results.

Here is my workflow diagram:



I’m using Jupyter Notebook for Python code. I create a jupyter notebook file name CA\_1 which means Class Assessment 1. After that, I imported all libraries which I will need to do all the Python code, so here are all the **Python libraries** that I used in my code:

* **import numpy as np** (used for mathematical functions)
* **import pandas as pd** (used for data manipulation and analysis like data cleaning, merging, filtering etc.)
* **import matplotlib. pyplot as plt** (used for graphical plots and charts)
* **import seaborn as sns** (same as matplotlib, but
* it’s efficient from matplotlib)
* **import plotly. express as px** ( used for high-level interface visualization, which shows all details in the graphs)
* **from sklearn.preprocessing import StandardScaler** ( used for Standardization, which helps to rescale values into mean and standard deviation)
* **from sklearn.model\_selection import train\_test\_split** (used for splitting the dataset into a training set and testing set)
* **from sklearn.linear\_model import LinearRegression, LogisticRegression, Ridge, Lasso** (used for predicting a continuous target variable and linear relationship between a dependent variable and independent variable)
* **from sklearn. ensemble import RandomForestRegressor**
* **from sklearn. Tree import DecisionTreeRegressor**
* **import tkinter as tk (** It is a Python GUI toolkit. I used it for creating graphical user interface that provides the detail of my work**)**
* **from tkinter import ttk (** It provides additional widgets and styling in GUI**)**
* **from matplotlib.backends.backend\_tkagg import FigureCanvasTkAgg(** Itallows you to entrench matplotlib figures into tkinter applications and display them as cooperative plots.)
* **from sklearn.model\_selection import cross\_val\_score** (helps to ensure that the model is not overfitting to the training data and provides a more reliable estimate of the model's performance on unseen data)
* **from sklearn.model\_selection import RandomizedSearchCV**
* **import pymongo (**It used for connecting a high-level API with databases, query, and manipulate data easily)
* **from pymongo.mongo\_client import MongoClient (**itis a class provided by pymongo that signifies a connection to a MongoDB server.)
* **from pymongo.server\_api import ServerApi (**It enables you to postulate the server API version and features to use when connecting to a MongoDB server**)**
* **from sklearn.feature\_extraction.text import TfidfVectorizer (** The main purpose of using the library is to convert text data into a numerical data that can be used as input for ML algorithms. It takings a corpus of text documents as input and performs the following steps:

Tokenization: It breakdowns each string into individual words.

Counting: It counts the frequency of each word in each string.

Vectorization: It converts the text data into a matrix representation, where each row represents a text and each column represents a word.**)**

* **from sklearn.model\_selection import GridSearchCV**

(these two are used for Hyperparameter tuning, which is used for improving model performance)

* **from sklearn.metrics import r2\_score** (used to calculate the accuracy of models)
* **from sklearn.metrics import mean\_squared\_error** (used to calculate the mean square error of predicted outcomes)
* **from sklearn.metrics import mean\_absolute\_error** (used to calculate the mean absolute error of expected results)
* **from sklearn import metrics** (evaluating the performance of the machine learning model)
* **from SciPy. Stats import norm** (used for normal distribution)

I used all these libraries in my project and I mentioned the use of all these libraries with them.

# Exploratory Data Analysis:

Exploratory Data Analysis (EDA) is an important technique used to understand and analyze the characteristics of a given dataset. Initially, I created a data frame of the “USA\_construction\_cost\_of \_healthcare\_project” dataset as df and “Total Production in Building and Construction Sector” as df2. I also attach screenshot of the datasets first and last rows in below.

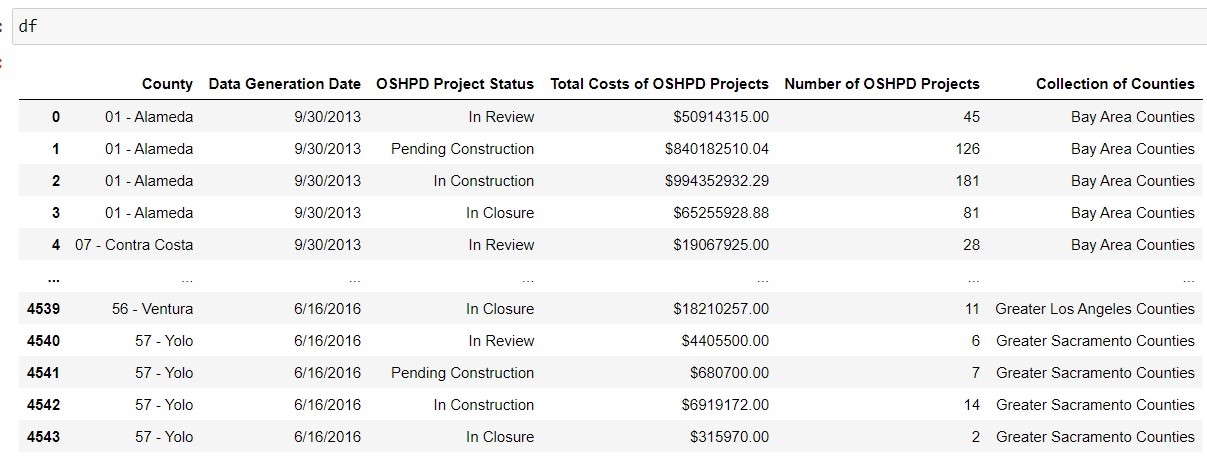
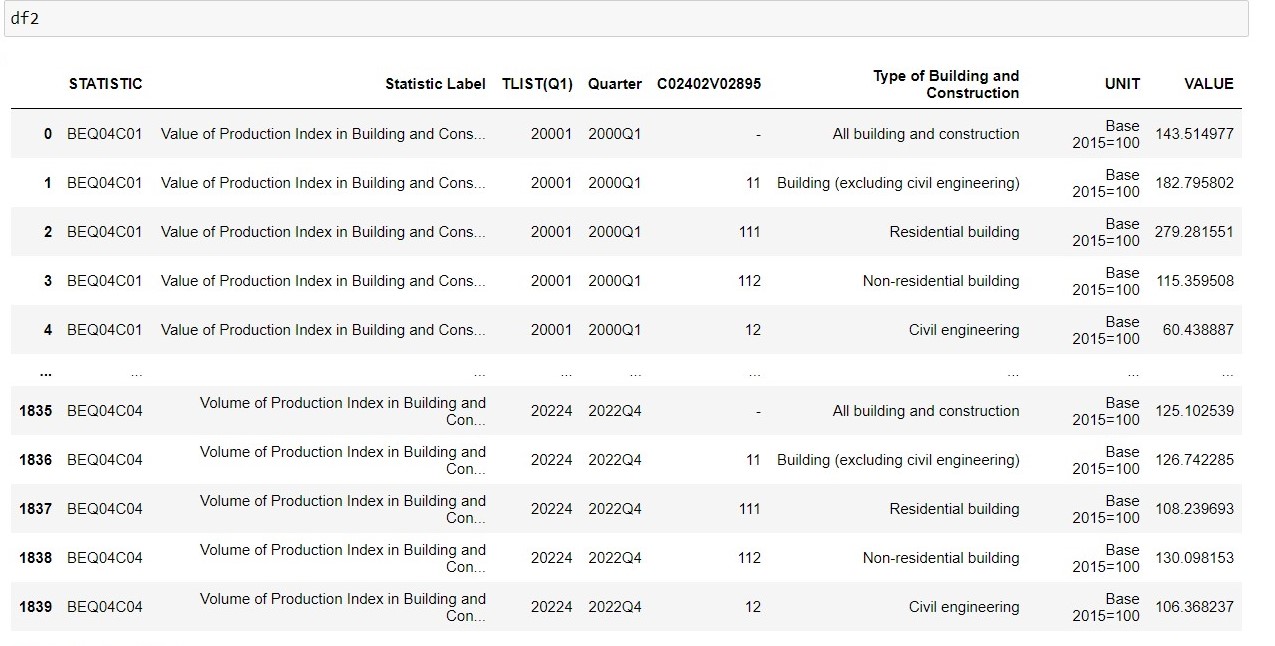


Figure 1



You can see these two different dataframes. So I have to apply data analysis steps on them to clean and arrange for the machine learning modelling. Also have to understand the insights of the datasets I apply the steps of data cleaning on both of the datasets.

## **Data Cleaning:**

Let’s first perform data cleaning on the first dataframe. There are six different features of this dataset. But not all features are useful for data analysis so that why I’m performing EDA on it to get some useful or knowledgeable insights for modelling. You can see that the name of the features is very difficult to type and access so that why I renamed them and also change the sequence of the columns.

Figure 2

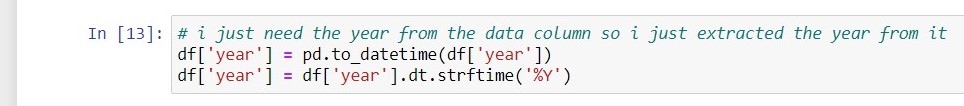
After that I just fetch the year from the year column because there are very less unique dates in that column so that’s why its not good approach to apply time series on it so I just only fetch year from that column.

Figure 3

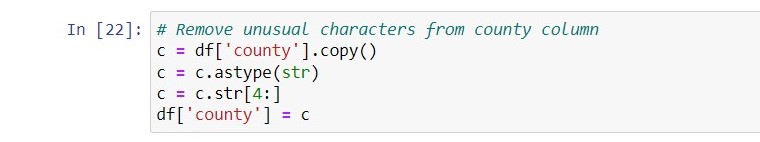
Then you can see that the county column values have unusual characters in all values so I have to remove them too so I removed them and just fetch the name of the county.

Figure 4

Now the main thing is that I check the datatypes of all columns so the cost column datatype is string so I also change it into float. And change the datatype of year into integer. So here are the datatypes of the all column in below figure.

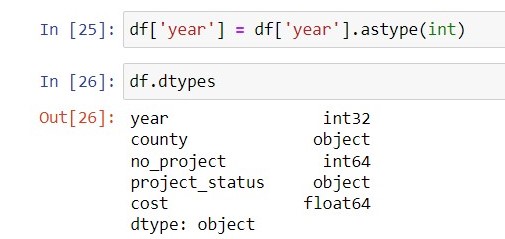


Figure 5

The dataframe one is almost cleaned but there is some changes I have to do so I did all these changes on the dataset then my dataset is ready for the other steps of analysis like visualization and machine learning modelling.

I did the same cleaning process on the second dataframe. First, I drop 'Quarter', 'C02402V02895', 'UNIT', these column from the dataset because it not useful in my analysis and modelling. After that I rename the column names and then one of the column names as “stats label” have categorical values which are not shown properly in the view so I renamed them as well for properly readable from the outputs. Then I also change the datatypes of the columns.



Figure 6

Now the both datasets are cleaned and ready to perform some visualization and train machine learning algorithms for on it. The data analysis of third dataset will be shown in the sentimental analysis section.

After the cleaning both datasets look like that:

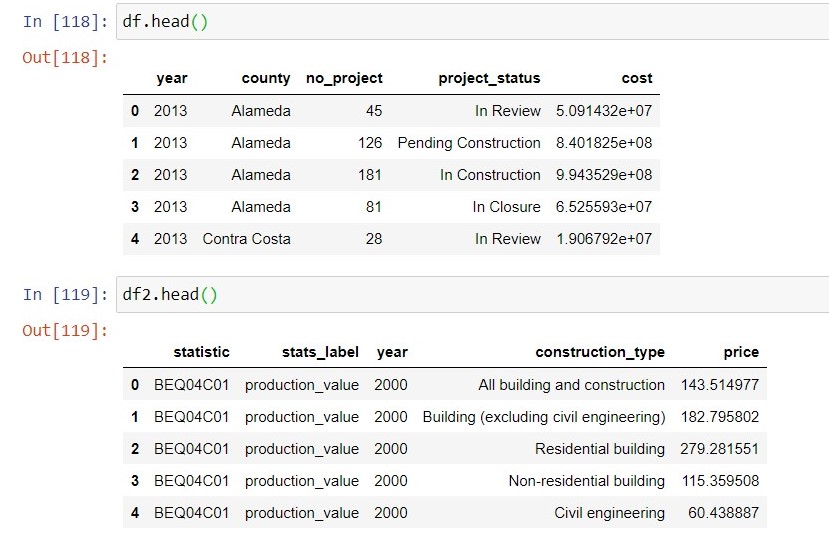


Figure 7

# Data Visualization:

After that I apply some graphical representation on the datasets for understanding the datasets and help us to gain an initial understanding of the dataset by visually investigative the patterns, trends, and distributions of the data. It helps us identify outliers, clusters, correlations, and other important features of the data that may not be deceptive from the rare numbers alone. The main thing of performing visualization is that it helps us to understand the distribution of the dataset like its distributed normally or what type of distribution it used so that’s why the visualization is the main part of the exploratory data analysis.

**Bar plots:**

Here are the distribution graphs of both dataset side by side

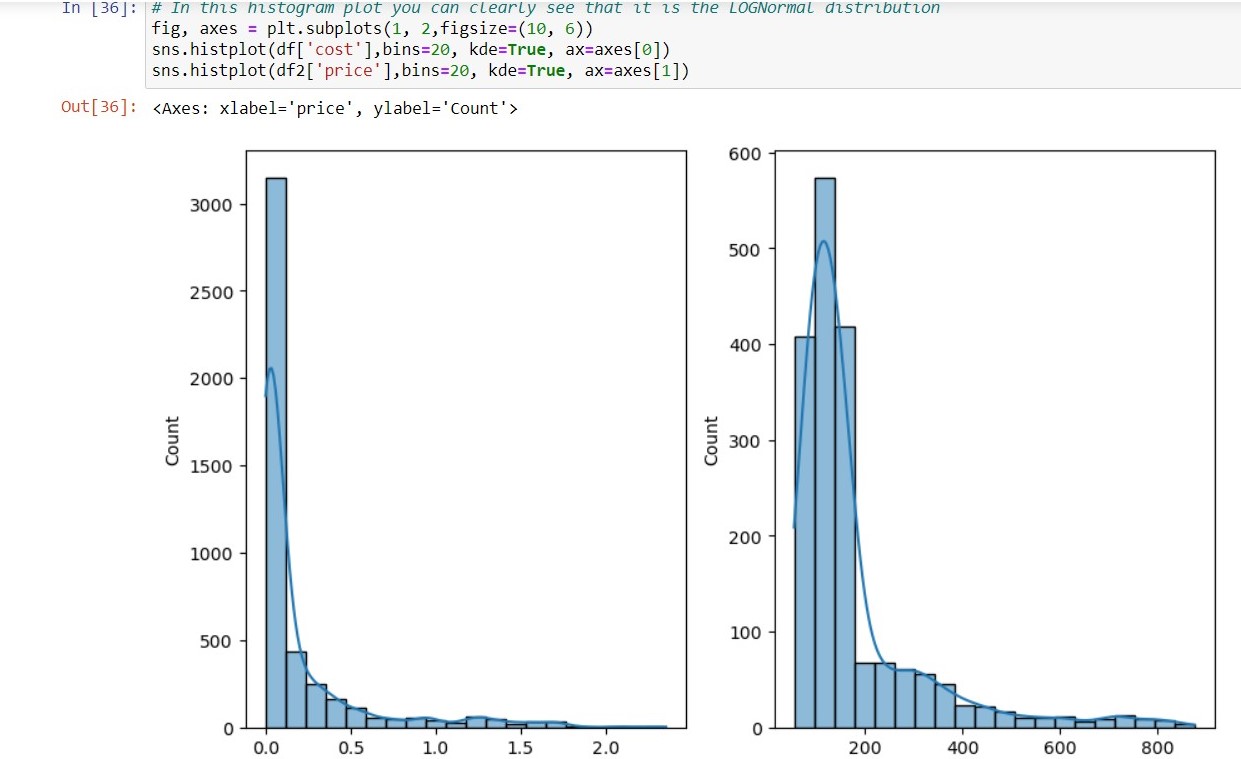
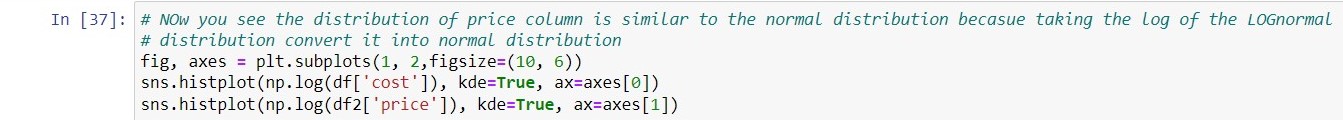


Figure 8

This graph clearly shows that the distribution of cost column and the price column from the datasets are log normally distributed which mean that both are right skewed meaning it has a long tail on the right side. This skewness occurs because taking the logarithm of a positive value compresses the data towards zero. For converting the log-normal distribution to normal distribution we have to take log of both column and then plot it. I have done this to and distribution converts to normal. You can see code in the figure below:



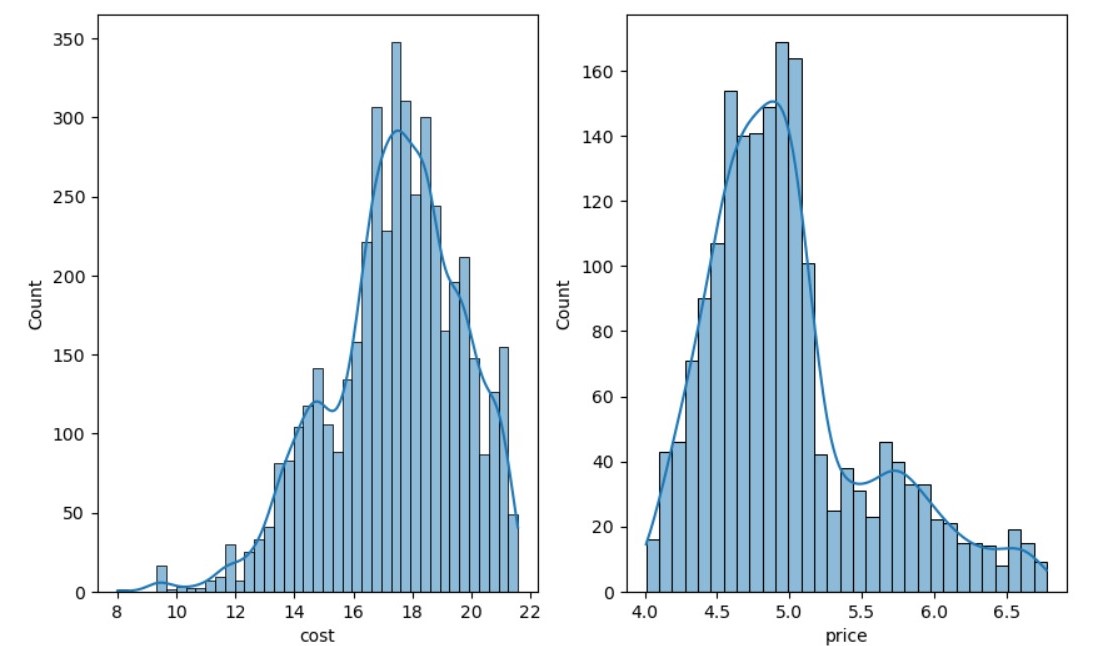


Figure 9

Now you can see it converts to normal distribution by taking the log of both columns.

After that I will show the cost by the counties in the graph, you can visualize easily in which county the constructional cost expensive and where is the lowest cost of construction. You can clearly see in the graph; in Los Angeles the constructional cost is much expensive than from the other counties. After that San Francisco, Santa Clara and Alameda are also expensive but cheaper than Los Angeles. And the other counties have a average cost of construction in the USA.

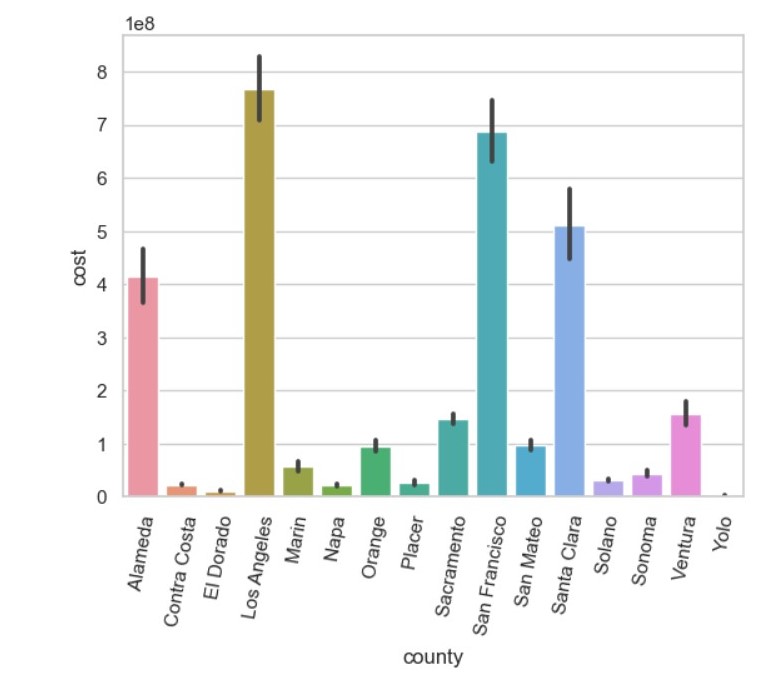


Figure 10

I also draw the graph of the Ireland construction dataset in which graph shows the cost of the construction by constructional type. It shows residential buildings are much expensive than the civil an non-residential.

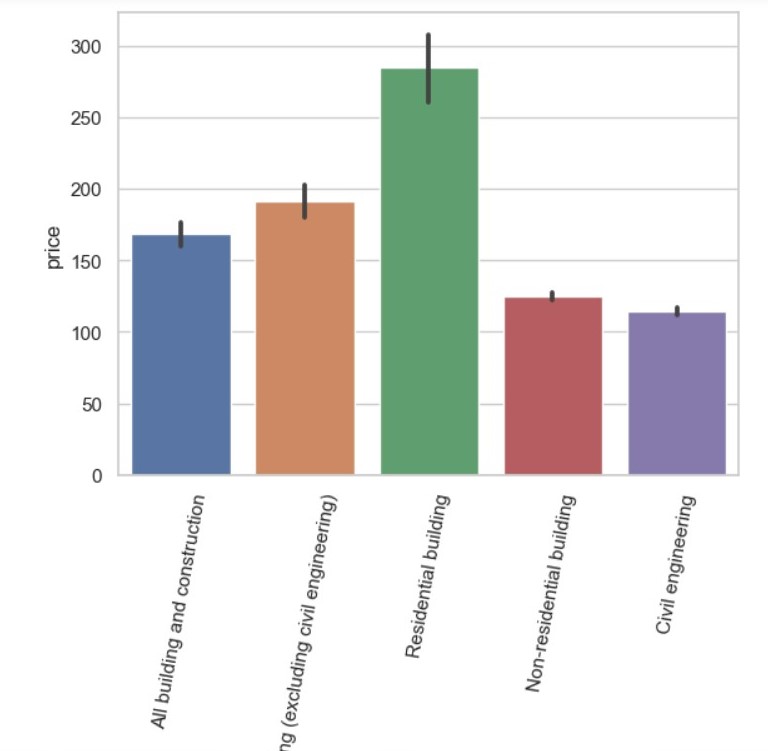


Figure 11

**Box plots:**

Now I will show you the boxplot of the USA constructional dataset. In this dataset, it shows all statistical computation of dataset one. When you hover, you mouse on the boxplot then it will show the mean value, median, maximum, minimum and quantiles as well. It’s the best approach for visualize the statistical information of dataset. Boxplots are actual for comparing the distributions of multiple groups or categories within a dataset. Plotly allows you to create grouped or categorical boxplots, where each box represents the distribution of a specific group or category. By visually comparison the boxes, you can classify differences in medians, quartiles, and the spread of data between the groups. Now I will show you the USA dataset boxplot which has log distribution.

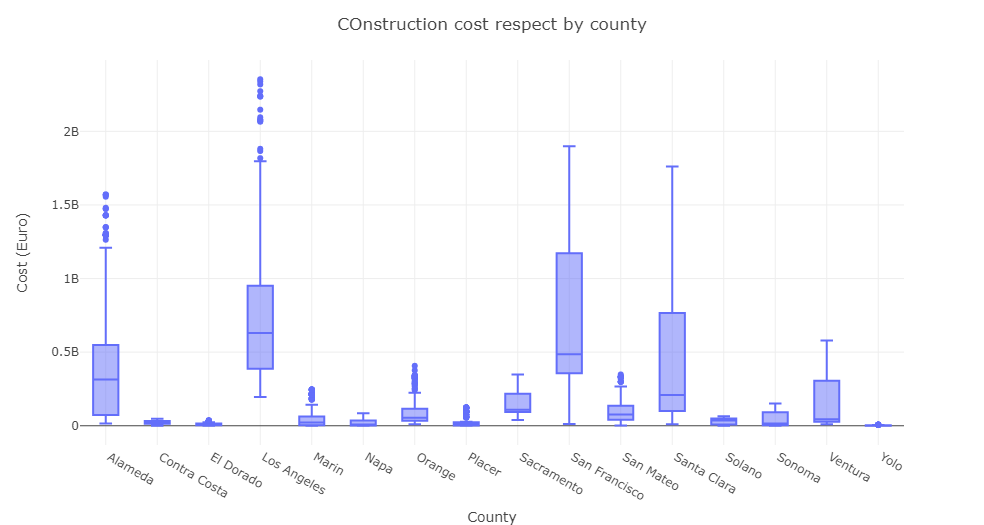


Figure 12

In this plot you have noticed that this graph not represented completely and feel difficult to understand. The reason behind is that the values of cost are so high so that’s why graph is not perfectly shown. For solving this problem, I do take log of cost column and then plot the graph. After that graph shows the actual values and explain all the detail clearly. Hers is the same graph with normally distribution. Here is the code and graph after applying log on cost column

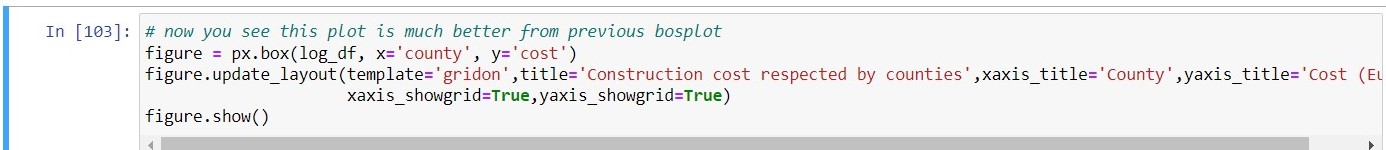




Figure 13

Now its easy to understand and more informative than before. Boxplots help assess the skewness of a dataset. By exploratory the position and the relative positions of the quartiles, you can conclude whether the data is symmetrically distributed or skewed to one side. This interactivity improves the understanding and exploration of the dataset.

I have also plotted the boxplot of Ireland construction dataset. I will show you the graph of second dataframe.

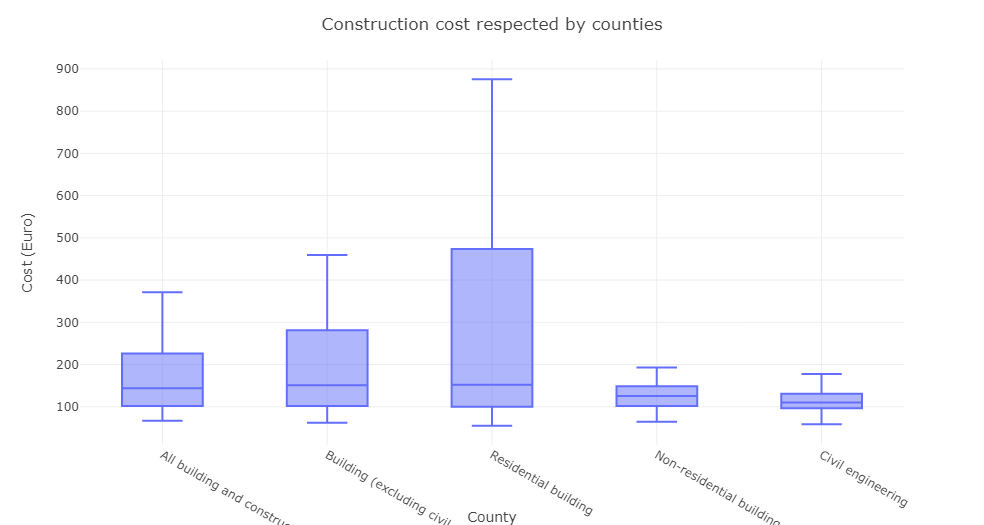


Figure 14

This plot display summary statistics such as the median, quartiles, and the minimum and maximum values. Plotly delivers options to customize the display of these statistics, letting you to show additional information such as the mean, standard deviation, or confidence intervals of the second dataset.

I have also shown you the how many numbers of projects are in working by yearly. USA do 37615 projects in 2013, 134370 projects in 2014, 143225 projects in 2015 and 56304 projects in 2017. I also plot graph for this representation.



Figure 15

Now I will show you the number of projects by counties of USA.

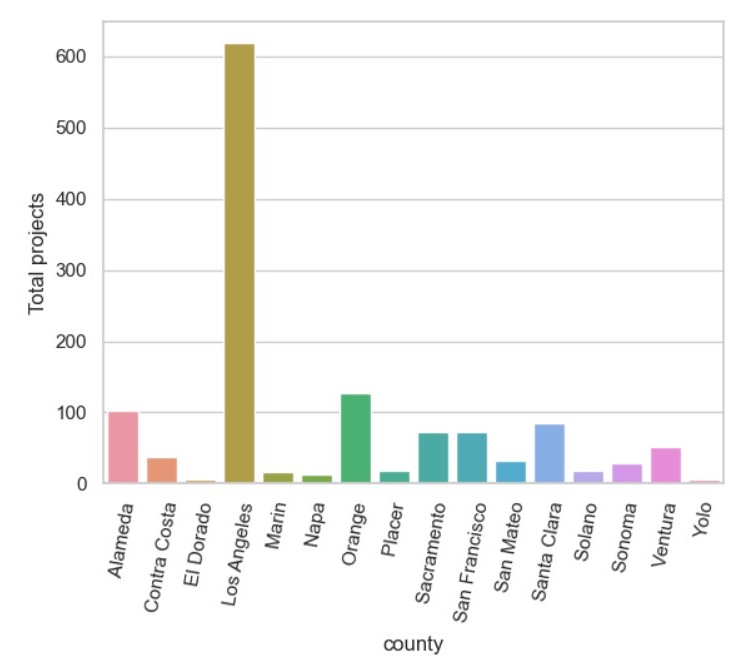


Figure 16

This shows that in Los Angeles more projects are done and lowest has done in Contra Costa.

# Graphical User Interface Graph with Tkinter:

I practice Tkinter Python library for a modest and influential way to generate graphical user interfaces (GUIs) for desktop applications. I container design and develop windows, dialogs, buttons, menus, and various other interactive components that make up the user interface of your request.

Here is the code of my GUI graph app:



Figure 17

And the output of this piece of code is here

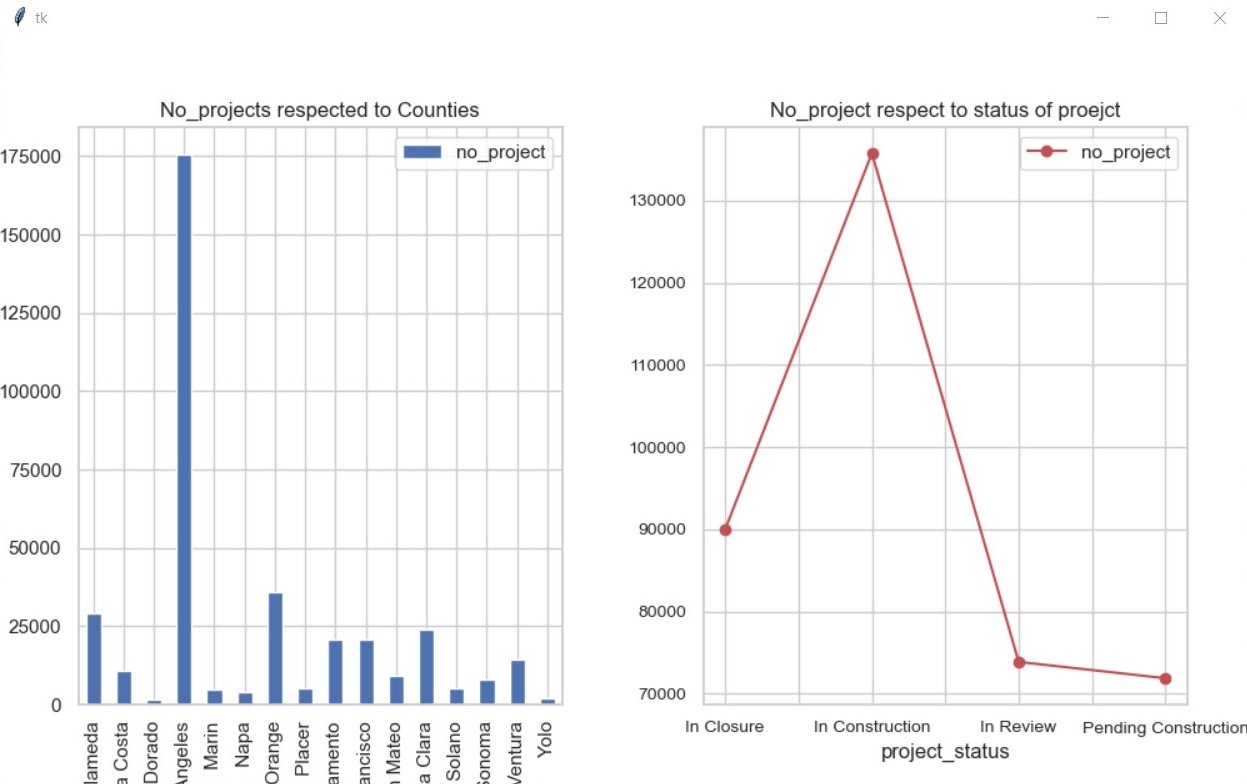


Figure 18

I will explain you all my code here

root = tk.Tk(): This line creates a Tkinter window, which serves as the main window for the GUI.

bar1 = FigureCanvasTkAgg(figure1, root): This line creates a FigureCanvasTkAgg object, associating the Figure with the Tkinter window.

bar1.get\_tk\_widget().pack(side=tk.LEFT, fill=tk.BOTH): This line places the FigureCanvasTkAgg widget in the Tkinter window, stipulating its position and filling the available space.

line1 = FigureCanvasTkAgg(figure2, root): This line creates a FigureCanvasTkAgg object for the second Figure and associates it with the Tkinter window.

line1.get\_tk\_widget().pack(side=tk.LEFT, fill=tk.BOTH): This line places the second FigureCanvasTkAgg widget in the Tkinter window.

inter2.plot(kind='line', legend=True, ax=ax2, color='r', marker='o', fontsize=10): This line creates a line chart using the plot() function from pandas, specifying the kind as 'line'. It uses the grouped Dataframe inter2 as the data source and plots the data on the subplot ax2. It also sets the line colour to red, adds markers at data points, and adjusts the font size of the plot.

root.mainloop(): This line initiates the Tkinter event loop, allowing the GUI window to be displayed and interacted with.

After that I create a desktop app which get input of x-axis and y-axis and then plot the bar chart in your machine. Here is the graphical output of that piece of code is:

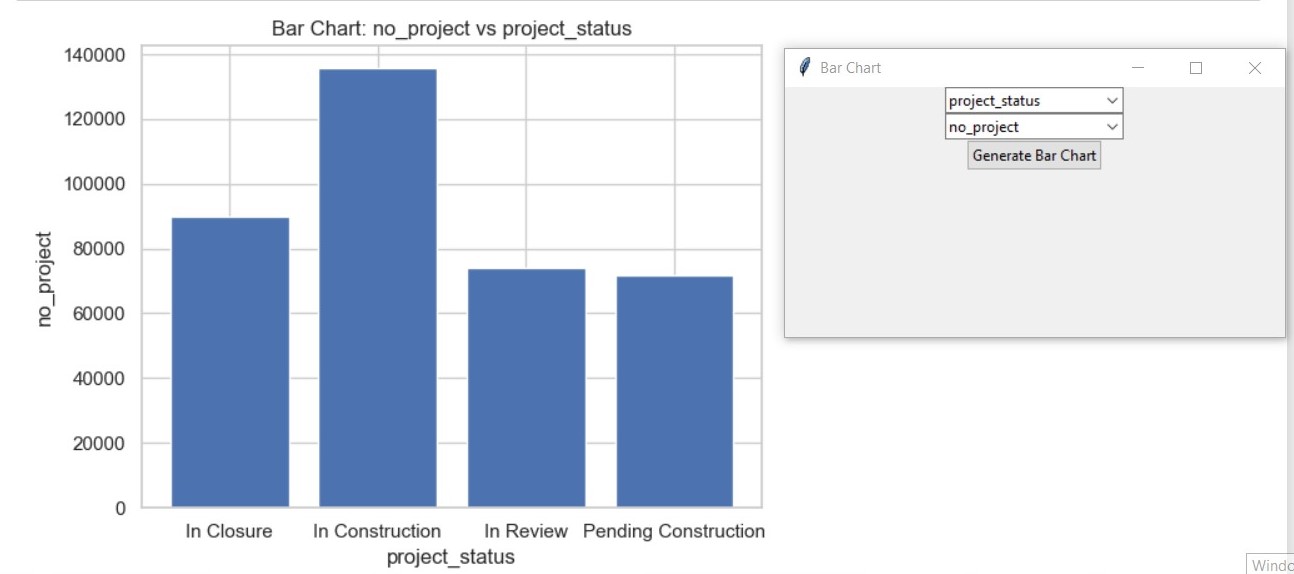


Figure 19

This graph shows number of projects with the project status like seventy thousand and something project are in review. So that kind of graphical interface are really attractive and responsive approach to explain your insights of the datasets. Now jump to the next step of my project that is hypothesis testing. Which proof that the dataset features used same distribution or not.

Let’s talk about hypothesis testing in detail

# Hypothesis testing:

These are the tests that I apply on my datasets:

**One-sample T-test:** It Used to compare means between two groups whether a sample mean different from a population mean. I apply one-sample t-test on USA dataset for proof that Null hypothesis (H0) have to be accepted or rejected. Here is the code

Figure 20

You can see first I take the mean of cost column then take the sample from cost column and then take mean of the sample set. After that I apply one-sample test on it and get p-value 0.746958 which is compared to the significance level (alpha) of 0.05. The null hypothesis (H0) states that there is no significant difference between the sample mean and the population mean. The p-value denotes the probability of perceiving a sample mean as extreme as the one obtained if the null hypothesis is true. When p-value is less than from 0.05 it means we rejecting the null hypothesis but in our case p-value is greater from 0.05 so there is not enough evidence to reject the null hypothesis. I also plot the graph of means.

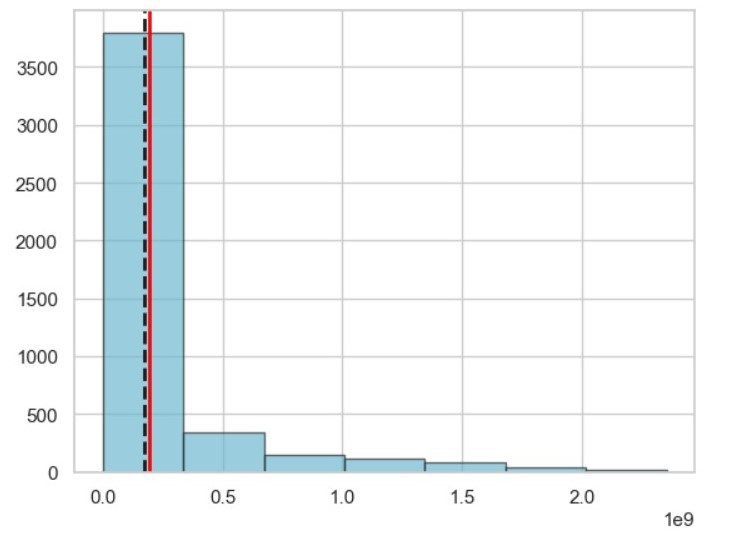


Figure 21

**Two-sample T-test:** The two-sample t-test is used to compare the means of two independent groups to regulate if there is a significant difference between them or not. I apply this test on my dataset you can see the figures shown below

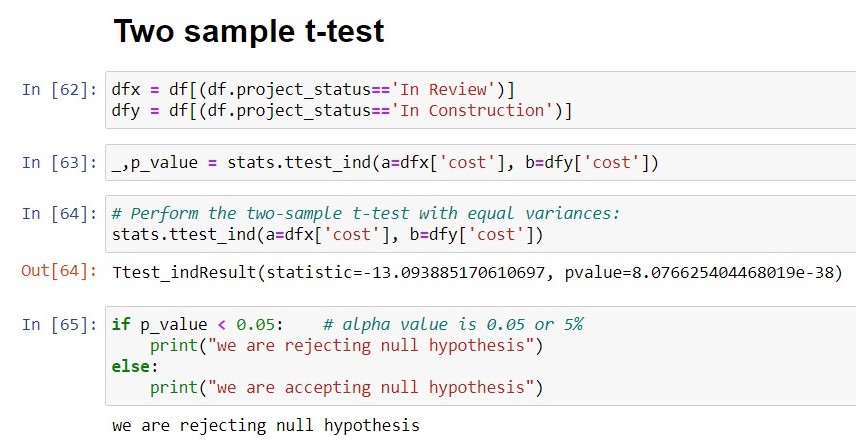


Figure 22

I take cost of two different project types and I apply two-sample test on it for checking there is difference between on both distribution or they follow the same distribution. Then I get p-value is 8.076 which directly reject the null hypothesis. It means we have not get enough evidence to accept the alternative hypothesis (H1).

Here is the graph of the test

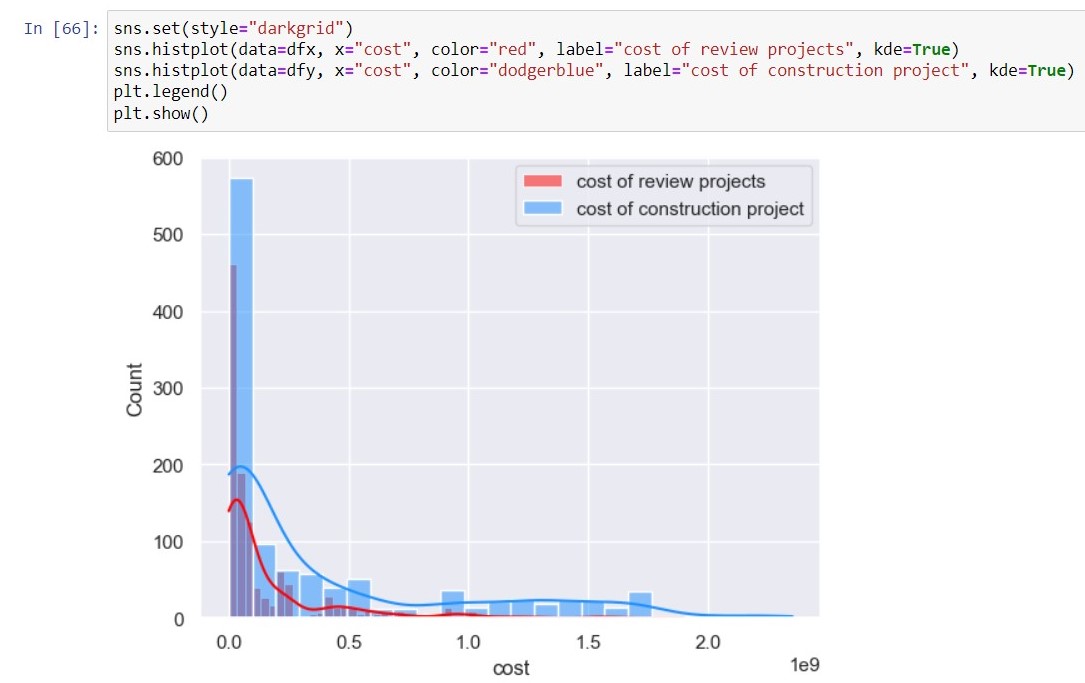


Figure 23

**Two-sample KS-test:** The two-sample KS test is a nonparametric test used to compare the distributions of two independent samples. It determines whether the two samples are drawn from the same distribution or if they significantly differ. I also applied this test too in my dataset and the code are shown in below:

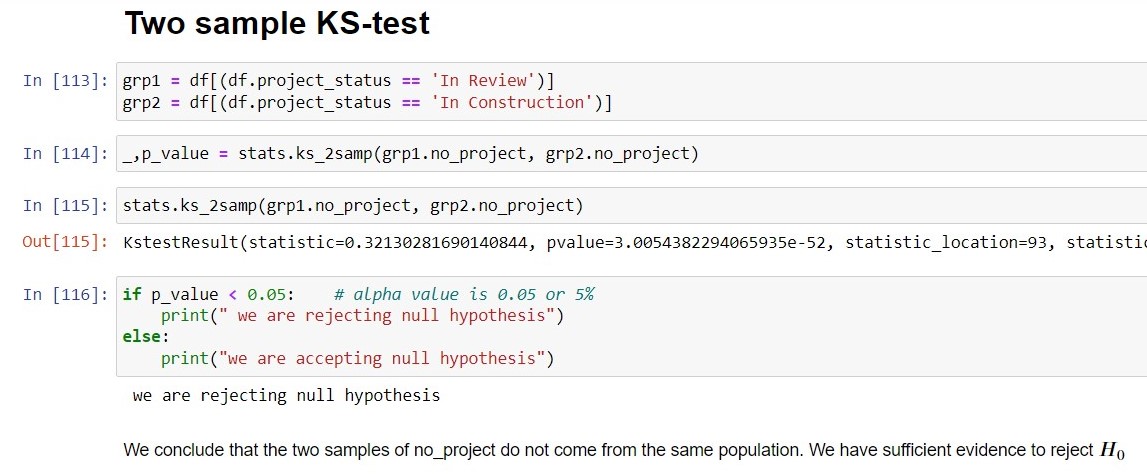


Figure 24

I applied two-sample KS-test on two different groups and we get the p-value is 3.00543 which means it rejecting the null hypothesis and accepting the H1. It also proofs that both groups do not come from the same population so we have sufficient evidence to reject the null hypothesis. I also plot the graph for this test shown below

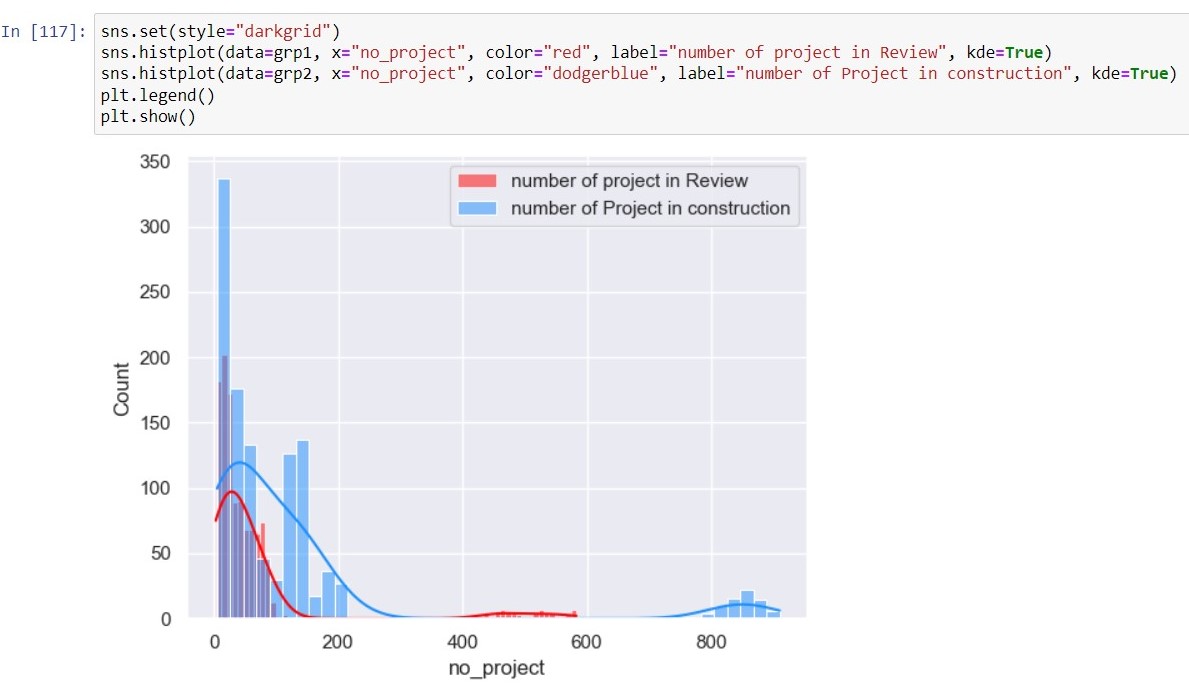
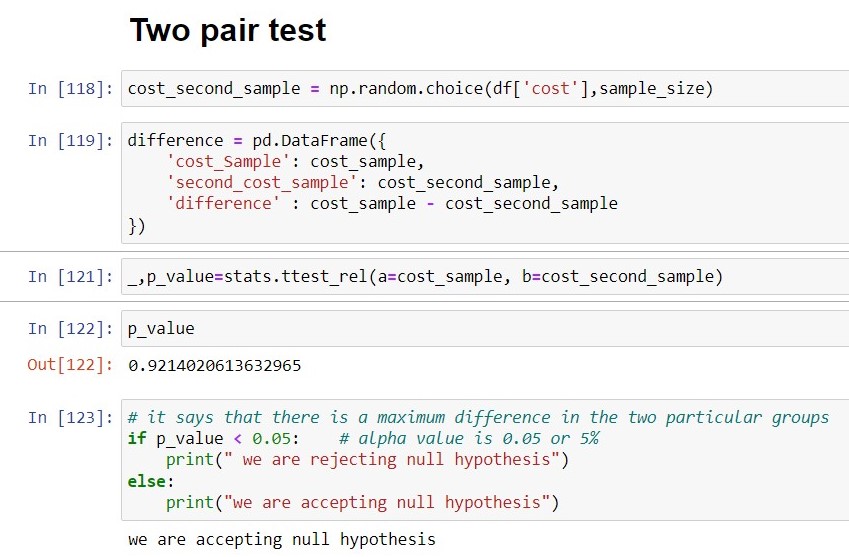


Figure 25

**Two pair T-test:** The paired t-test is used to compare the means of two related or paired groups. It is naturally applied when the data points in the two groups are corresponding or paired in some way. Let’s apply it on my dataset and find out the results.



A paired t-test is accomplished to compare the means of two related samples like cost\_sample and cost\_second\_sample. The null hypothesis (H0) states that there is no significant difference between the means of the two samples. After execution the paired t-test, the resulting p-value is related to the significance level (alpha) of 0.05. The p-value represents the probability of observing a difference in means as extreme as the one obtained if the null hypothesis is true. The p-value is less than the significance level (alpha), it means that the observed difference in means is significantly different from zero, and the null hypothesis is rejected. This means that there is evidence to suggest that the means of the two related samples are significantly different.

## **Chi-square test:**

A chi-square test determines if there is a relationship between two categorical variables: 'county' and 'no\_project' from USA dataset. The null hypothesis (H0) assumes that there is no relationship between the variables. The contingency table, observed values, is formed using the 'pd.crosstab()' function, which represents the observed frequencies of individually combination of categories.

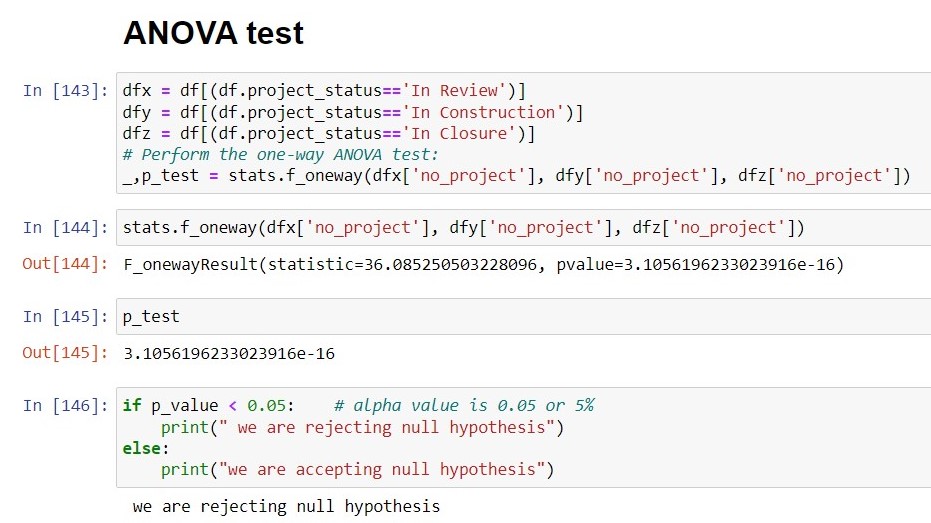
The 'stats.chi2\_contingency()' function returns multiple values, including expected frequencies stored in the expected values variable. The degrees of freedom are calculated created on the dimensions of the contingency table. The chi-square test statistic is computed by summing the squared differences between observed and expected frequencies, divided by the expected frequencies. The critical value is gained using the 'chi2.ppf()' function, which corresponds to the elected significance level (alpha) and the degrees of freedom. Then p-value is calculated using the 'chi2.cdf()' function, representing the probability of obtaining a chi-square statistic as exciting as the observed value. After that the output displays the calculated chi-square statistic, critical value, p-value, significance level (alpha), and degrees of freedom.

The first if statement checks if the chi-square statistic is greater than or equal to the critical value. So the chi-square value is greater from the critical value then it shows there is a relation between two categorical variable and we are rejecting H0.

Then the second if statement checks if the p-value is less than or equal to the significance level (alpha). So the p-value is less than from the alpha that’s why we reject the H0 which means there is a relation between two categorical variables.

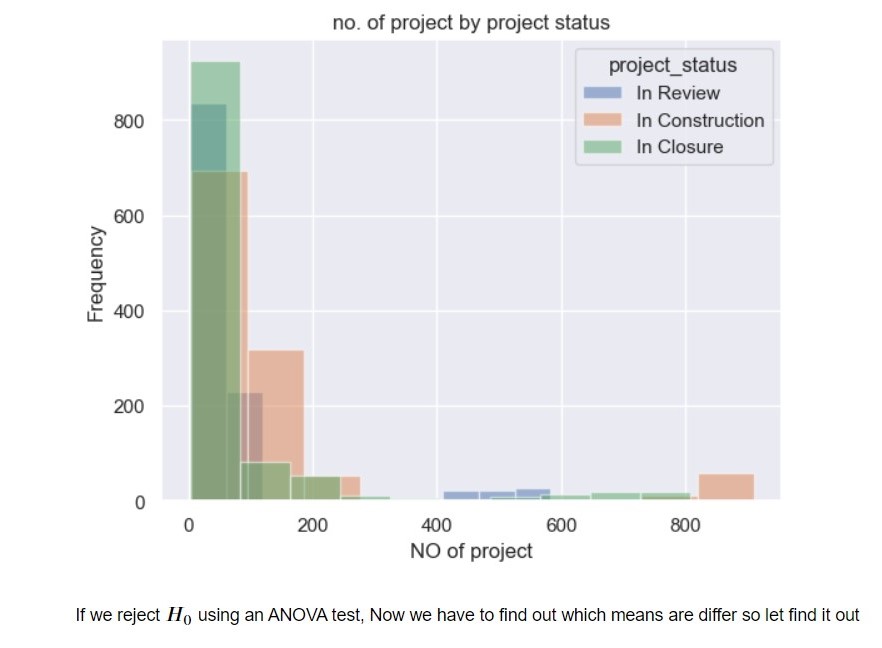
Rejecting the null hypothesis suggests that there is evidence to support the presence of a relationship between the 'county' and 'no\_project' variables in the dataset.

**ANOVA (Analysis of Variance):** It used to compare means among more than two groups. So, I create three different groups and trying to check there is difference between the means of these groups or not.



You can that I apply test on the three different groups which have different project status and the number of projects then we get p-value is 3.105619 which is less than from the 0.05, it means there are significant differences between the means of the groups.

Here is the graph of the output:



If we reject the H0 using an ANOVA test, Now we have to find out which means are differ so let find it out by perform a Tukey-Kramer analysis to see which of the means differ from each other.

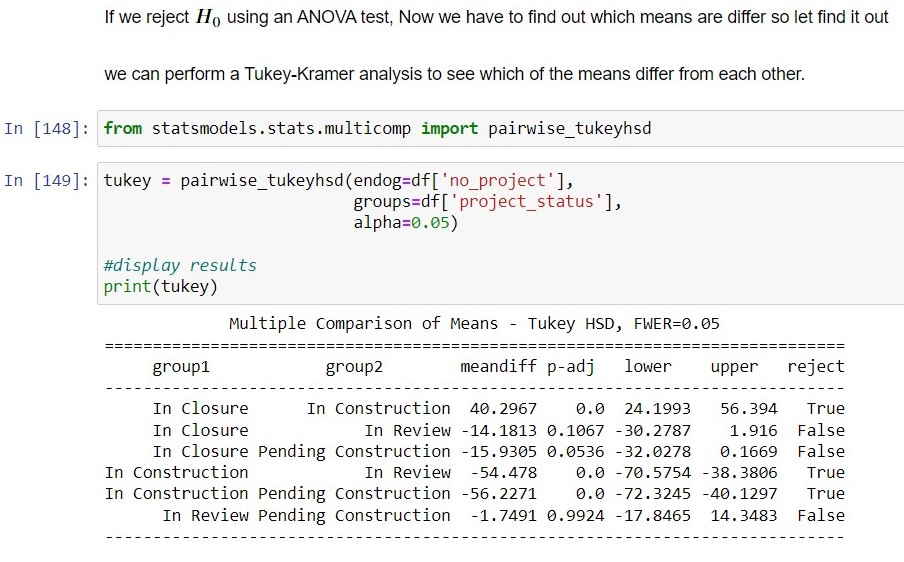


Figure 26

The first-row comparisons "In Closure" group and "In Construction" group. The mean difference is 40.2967, and the adjusted p-value is 0.0 (very small). The confidence interval for the mean difference ranges from 24.1993 to 56.394. Since the p-value is below the significance level 0.05, the null hypothesis is rejected, representing a significant difference between the means of these two groups.

The second row compares "In Closure" group and "In Review" group. The conversant p-value is 0.1067, which is larger than the significance level (0.05). Accordingly, the null hypothesis is not rejected, telling no significant difference between the means of these two groups.

The third row compares "In Closure" group and "Pending Construction" group. The adjusted p-value is 0.0536, which is close to the significance level (FWER=0.05). It falls in the range where one might consider rejecting the null hypothesis. However, the decision depends on the chosen significance level and the specific requirements of the analysis.

The remaining rows follow a similar interpretation, comparing different groups and determining if there are significant differences in their means based on the adjusted p-values.

Now jump to the next step of this report.

# Model Building and Evaluation:

Machine learning is a part of artificial intelligence that includes the development of algorithms and models that enable computers to learn from and make predictions or decisions based on data. It allows computers to automatically learn from experience or historical data without being clearly programmed for each specific task. In our case, we use machine learning algorithms for training models on historical data of house prices in Ireland. As you know, our dataset has continuous values, so we have to use regression algorithms like we have to predict the prices of the new house and second-hand houses in Ireland.

There are some reasons why I choose a supervised learning approach for training a model on my dataset:

**Predictive modelling**: A supervised learning method would be suitable for my dataset because I want to build a model that can predict construction cost in different counties of USA and Ireland based on historical data. I have to use the labelled data to train a model that learns the patterns and relationships between the features of the construction and their matching prices and then use the trained model to make predictions on testing data.

**Availability of labelled data:** A supervised learning method would be a good choice because I have a large amount of labelled data for construction cost in USA and Ireland. Labelling data allows the model to learn from given target values or labels, resulting in a more accurate and reliable model. I have a cost column as label data and price column in second dataset, which is useful for training my model for predicting construction expenses.

**Interpretability**: Supervised learning models are frequently more explainable than other machine learning methods, such as unsupervised or deep learning. This can be beneficial when understanding and explaining the predictions of the model, which may be necessary in a housing market situation.

**Task complexity**: A supervised learning approach can be operative if the task of predicting constructional cost in USA and Ireland is comparatively straightforward and can be represented as a plotting between input features and output prices. Supervised learning models are most suitable for tasks that can be defined as a plotting between inputs and outputs based on labelled data.

However, it's important to note that the choice of machine learning approach depends on various factors, including the specific problem I’m trying to solve. It's always important to carefully analyse the dataset and problem requirements before choosing the most appropriate approach for our use case.

I used some supervised learning algorithms to train the models, like Linear regression, Lasso Regression, Ridge Regression, Decision tree regressor, Random Forest Regressor, etc. These algorithms are used for training supervised learning protocols. And also apply sentimental analysis on hotel dataset and predict the reviews of customers like its positive or negative.

Let’s start by explaining the code I did for training the machine learning models.

As you know, I should have to clean my dataset for the training model, which I have done before this part. All the features are dependent variables as they depend on the target variable, and the target variable is independent because it was not dependent on any variable. Now I have to split my dataset into two parts. One is for features, and the other one is for targets or labels. So, I assign all the features of my dataset to the **x** variable and target values to the **y** variable, as you see in the picture below.

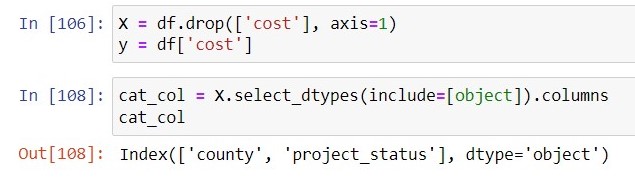
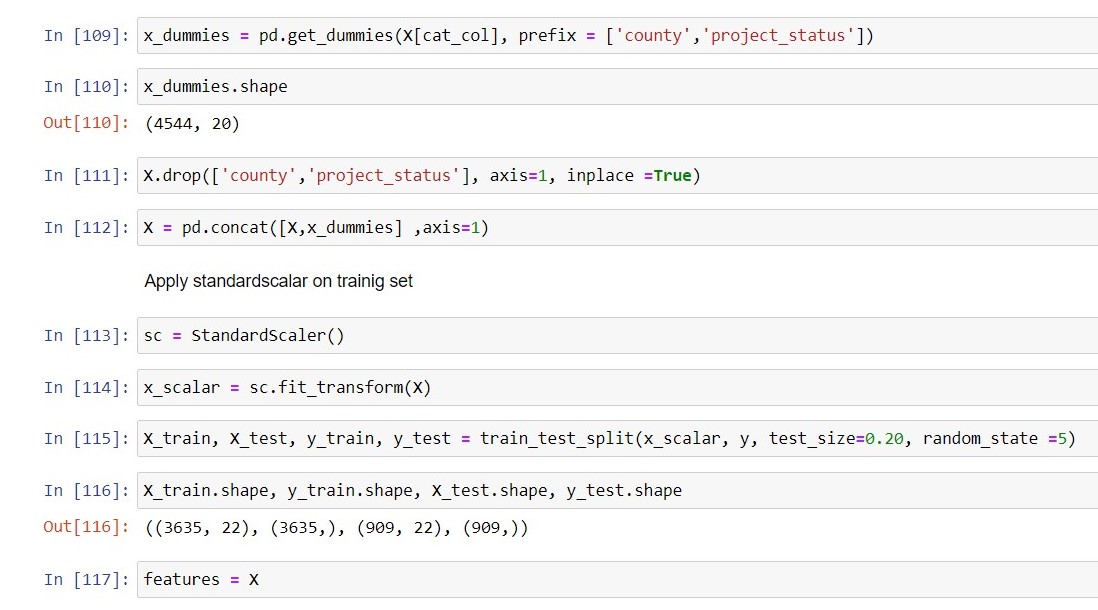


Figure 27

As you know, we have two categorical columns project\_status, and county. For training a machine learning model contains only integer values like 0 to 1 form, so we have to convert our categorical column into an integer.

For that we have **One-hot encoding approach** which convert generates a binary (0 or 1) indicator column for each unique category in the categorical column. It signifies each category as a separate column and assigns a value of 1 to the matching column if the category is present and 0 if it is not. I used a pandas **get dummies** method to convert my categorical columns into integers for feeding them into the machine learning model. Here are the steps that I do in my code: 

After doing this, our dataset shape will change and now it has 22 columns. You can see in the picture above that the column is increased. Now I have to concate these columns with our dataset and drop the categorical columns.

After that, we have to standardize the columns because of the different scale of values in the dataset. Here are the steps for standardizing the columns. Now our dataset is ready for the training of the models.

I split the dataset into two parts for training the model. I apply the **train\_test\_split** method from sklearn to my dataset. I choose 20% data for testing and 80% for training the model.

# Machine Learning Algorithms:

I applied five different algorithms for training the model and also do cross validation for getting the actual results. Let’s talk about the algorithms and code

## **Random Forest Regressor:**

The Random Forest Regressor is a supervised machine-learning algorithm that is used for regression tasks. An ensemble learning method combines multiple decision trees to make predictions. It builds a group of decision trees, where each tree is trained on a random subclass of the data and a random subset of the features. This randomization benefit reduces overfitting and improves the accuracy of the model.

The Random Forest Regressor builds an ensemble of decision trees by training multiple trees on different subsets of the data. This ensemble approach helps improve the model’s predictive performance, reducing overfitting and providing more robust and accurate predictions compared to a single decision tree. It selects a subclass of data points and features for training each decision tree. It makes predictions by taking a common vote or being an average of the predictions of all the trees in the ensemble. It can provide information about the importance of, unlike features in making accurate predictions.

I set the criterion of random regressor as a **squared error** because I have a continuous target variable. Then I used for loop to try the different values of the n-estimator from 50 to 200. I check the effect of different values of the n-estimator and plot the graph of the accuracy. You can see the code in the notebook. I also apply the **cross-validation technique** in all the algorithms to check how it affects the accuracy of all the models. Here is the table of accuracy with different n-estimators.

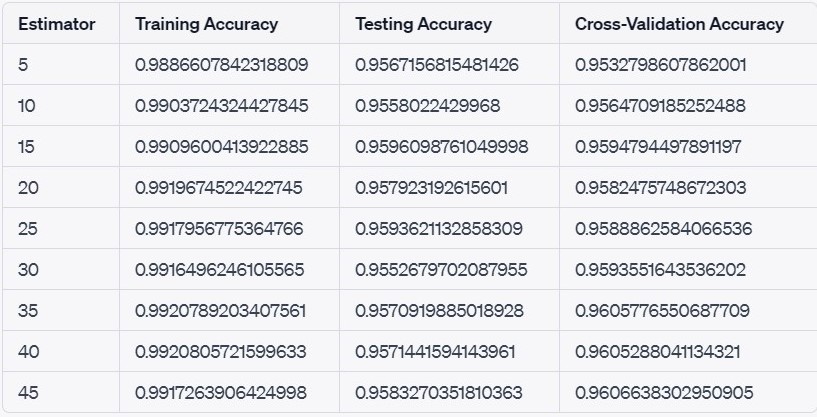


Figure 28

This is the table of random forest regressor performance with different values of n-estimators. I also draw the graph of the performance.

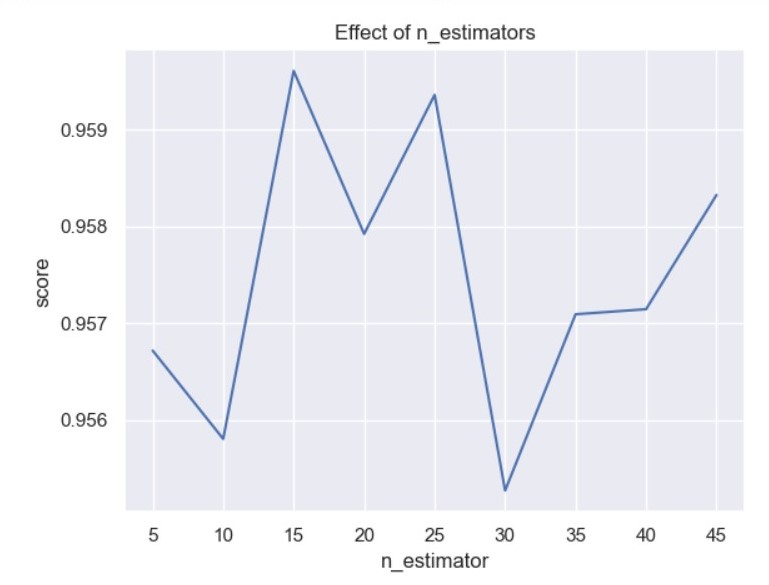


Figure 29

## **Linear regression:**

Linear regression is a technique used for the model connection between a dependent variable and one or more independent variables. It accepts a linear relationship, a sense that a straight line can signify the relationship. Linear regression is normally used for prediction and estimation tasks and is one of the simplest and most extensively used regression techniques. I trained a linear regression model on my dataset and get good accuracy. I also try the cross-validation technique on linear regression; you can see the code in the notebook file. Here are the results of the linear regression model.

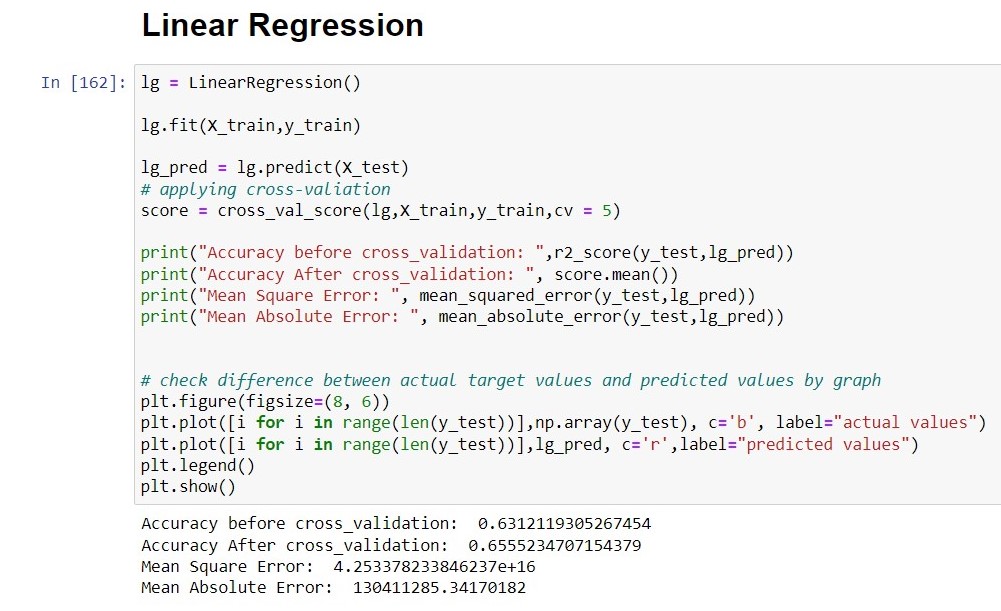


Figure 30

The accuracy of linear regression is not quite good but it acceptable but you noticed that after cross validation accuracy increased. I also plot the graph of accuracy.



Figure 31

You can see the predicted values shown in red lines and actual values show in blue lines and the difference will be shown clearly. Now I create a instance of RFE class for check the features importance. It takes two arguments:

rf: The estimator to be used for feature selection. In this case, it is rf, the model that was previously fitted.

n\_features\_to\_select: The number of features to select from the data. In this case, it is set to 10, indicating that the RFE algorithm will select the 10 most important features.

I access the selected features using the rfe.support attribute, which returns a boolean mask indicating which features were selected. The rankings of the features can be accessed through the rfe.ranking\_ attribute. Remember to replace X\_train and y\_train with your actual training data and target variable. Additionally, ensure that the rf model is instantiated and defined properly before fitting it. Here is the code:

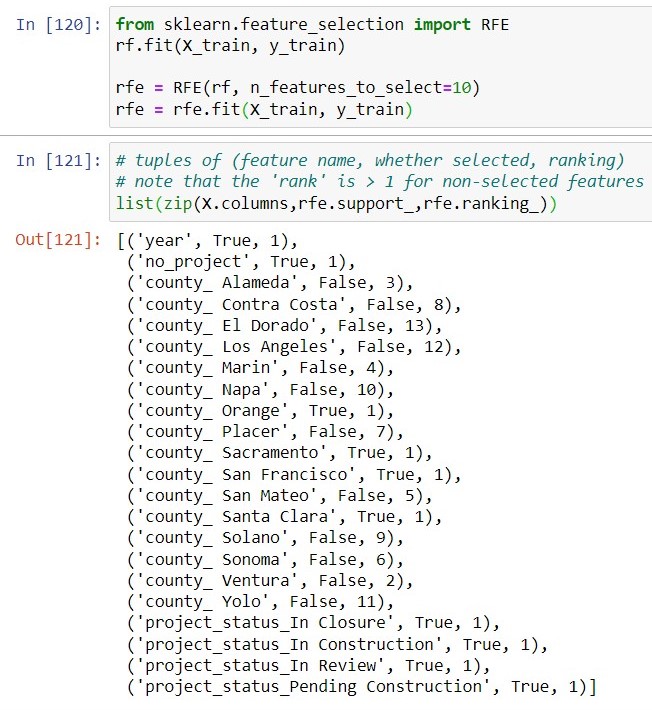


Figure 32

These selected features are considered to be the most important for the given task, based on the RFE algorithm's ranking and selection process.

## **Decision Tree:**

The decision tree algorithm is a well-liked machine learning method for classification and regression applications. It is a kind of supervised learning algorithm that creates a structure resembling a tree to make judgments or predictions based on the input information.

The fundamental of the decision tree method is to recursively split the data into branches or nodes according to the values of the input features. The splits are created to produce pure or homogeneous data subsets in the tree’s leaf nodes by minimizing the impurity or uncertainty of the data at each node. By moving along the tree from the root to the relevant decision-making node, the decision tree can then be utilized to make predictions or decisions.

I build a decision tree regressor model using the squared error criterion, and the splitter is random. I will show you the results of the decision tree regressor model.

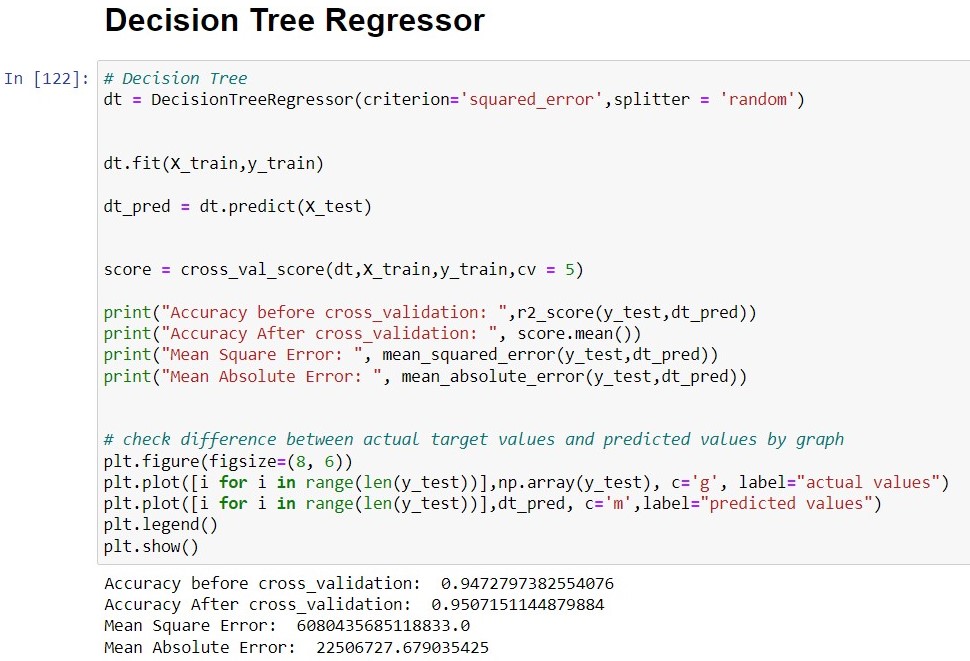


Figure 33

You can see the accuracy of the model is very good and after applying cross validation accuracy increased. I also plot the graph of the actual accuracy and predicted accuracy.

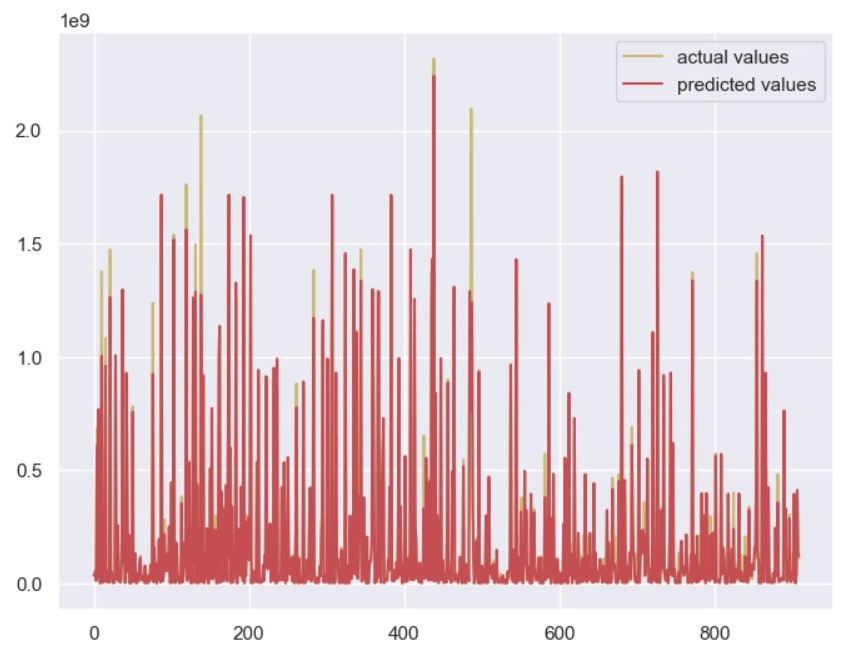


Figure 34

# Hyperparameter Tuning:

Hyperparameter tweaking is essential since selecting hyperparameter values may considerably impact a machine learning model's performance. A model's accuracy and speed may all be increased by selecting the right hyperparameter values, but electing the wrong ones can lead to overfitting or underfitting.

For hyperparameter tweaking, a variety of methods are available, such as:

**Grid Search:** The model is trained and tested for all possible combinations of a predetermined set of hyperparameter values in this method. The ideal set of hyperparameters is chosen as the set of values for the hyperparameters that produce the maximum performance metric, such as accuracy or F1 score.

**Random Search:** This method is comparable to grid search, except for training and assessment, a random selection of hyperparameter values is utilized rather than trying every conceivable combination. Due to the fact that it does not involve attempting every combination, this may be more effective in terms of computational resources.

I did hyperparameter tuning with both grid search cv and random search. There is no big difference between both results. They are mostly similar. In hyperparameter tuning, we can just change the parameters and check the performance of the model. Then we check which parameters’ accuracy will change for that; I just create a variable that stores the values as a list, then check which parameters’ random forest regressor model accuracy will perform better from before.

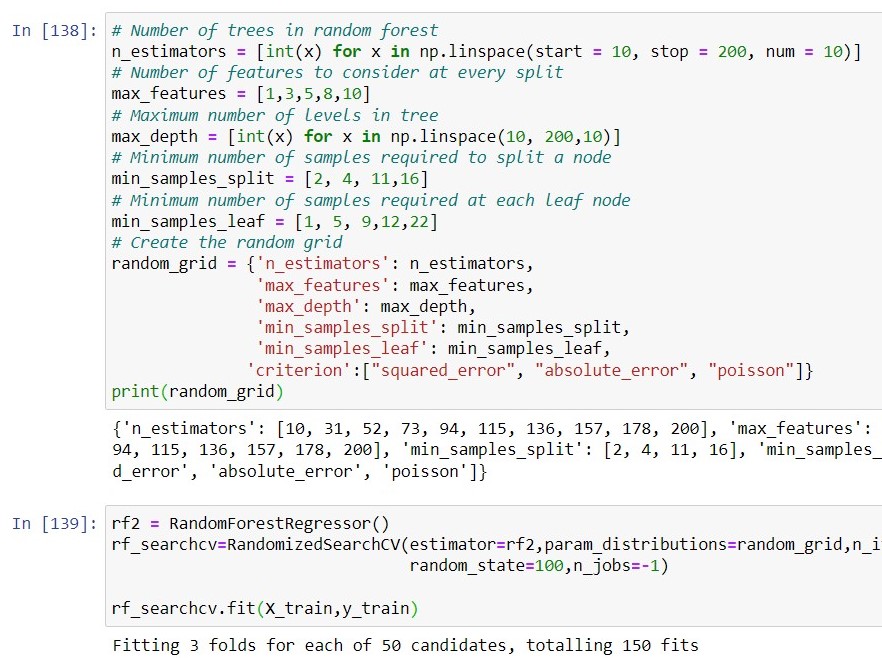


Figure 35

There are the parameters used for random forest regressor:

* n-estimators: It denotes the number of trees used for hyperparameter tuning. I randomly assign 10 values between 10 to 200.
* Max-features: It determines the number of features to study when looking for the best split at each node in the decision trees.
* Max-depth: It controls the random forest's maximum depth of individual decision trees. I randomly assign 10 values between 10 to 200.
* Min-samples-split: It outlines the bare minimum of samples needed to separate an inner node during decision tree development.
* Min-sample-leaf: It establishes the basic minimum of samples that must be present in a leaf node. It is used to control how well the tree can detect noise or outliers in the data.

After that, I assign all parameters in the parameter grid, then fit the model and get the best parameters for the random forest regressor.

I also check the table of random search cv scores in table. Table is very large but I’ll show you some examples with picture below.

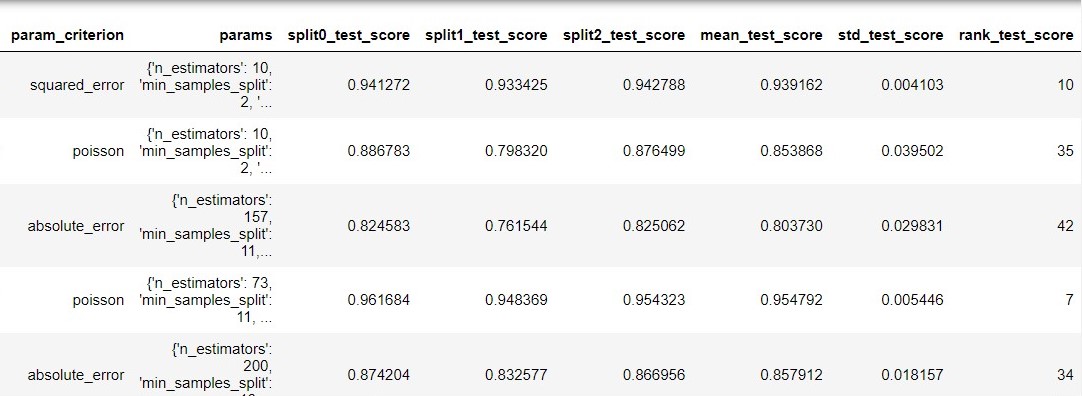


Figure 36

This table shows accuracy with different parameters but the best accuracy used squared error criterion and 10 estimators.

After that I do hyperparameters tuning with Grid Search CV. I used the same parameter for grid search cv and add values on them then apply it.



I had done the same steps as for the random search cv. But there is the same results for both methods with the different parameters.



Figure 37

In the starting you see model is overfitted and the accuracy is very low but with the depth the accuracy changed and go to the 87% which is very good.

I also used Lasso Regression and Ridge regression with different values of alpha. Ridge regression and Lasso regression are regularization techniques in linear regression. They are actually used for addressing overfitting in the linear regression model. The accuracy of both models is quite good. The ridge regressor gets 63% testing accuracy, and the lasso regressor gets 63% testing accuracy.

Here I attach the code of Ridge regressor and the Lasso regressor.

**Ridge Regressor:**

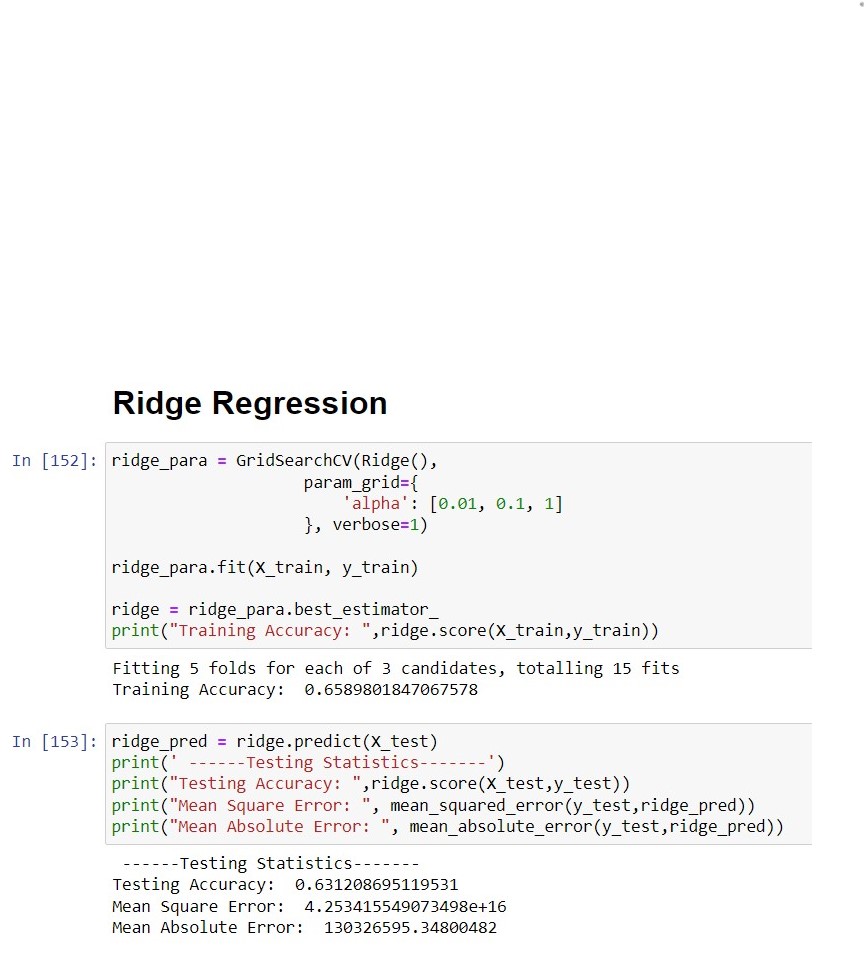
****

Figure 38

**Lasso regressor:**

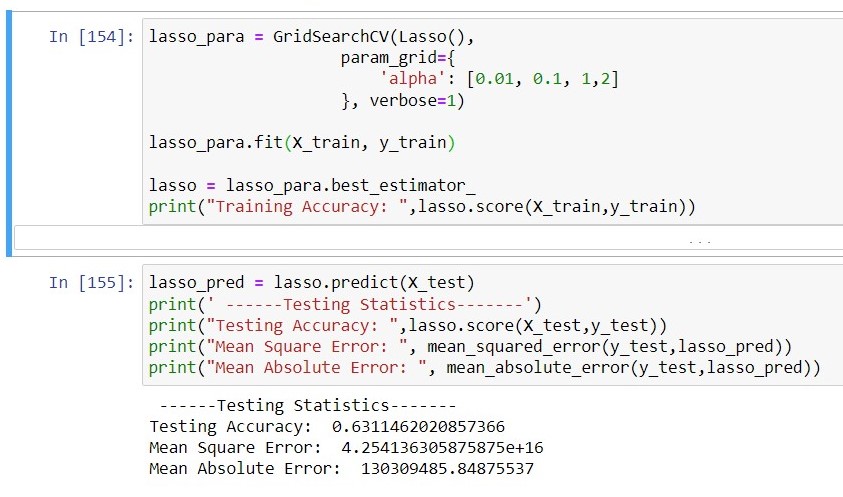
****

Figure 39

# Sentimental Analysis on Hotel Reviews:

In the last I apply sentimental analysis on the hotel dataset. I want to apply on the constructional reviews of Ireland country and compare with the other country but I there is not dataset in which I perform sentimental analysis so that’s why I search about other dataset on which I can apply sentimental analysis. So, I do sentimental analysis on hotel review dataset.



Figure 40

First, I upload the hotel dataset into my database by using the MongoClient library and then I create the client and connect with a server. I also apply check whether server is connected or not. When server is connected then I create database and collection which create on my mongoDB application. After that I fetch the dataset from the database and apply data analysis techniques on it. The dataset has unusual features or column so I drop them from the database. When I apply data analysis techniques then there are no null values in the dataset. We have one independent variable and one is dependent which is a target variable. You can the dataset in the figure below. I drop rows which is not useful and have null values. In target variables we have response of customers.

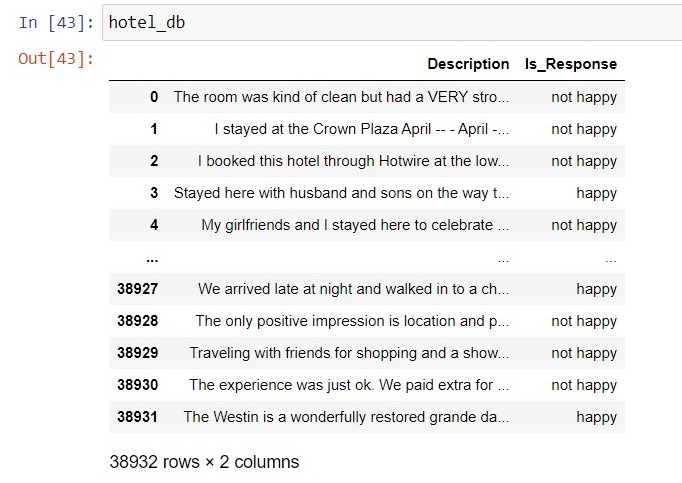


Figure 41

Response variable have two values Happy and not happy so I have to check the distribution of it so I draw a bar plot for that. Figure in the below.

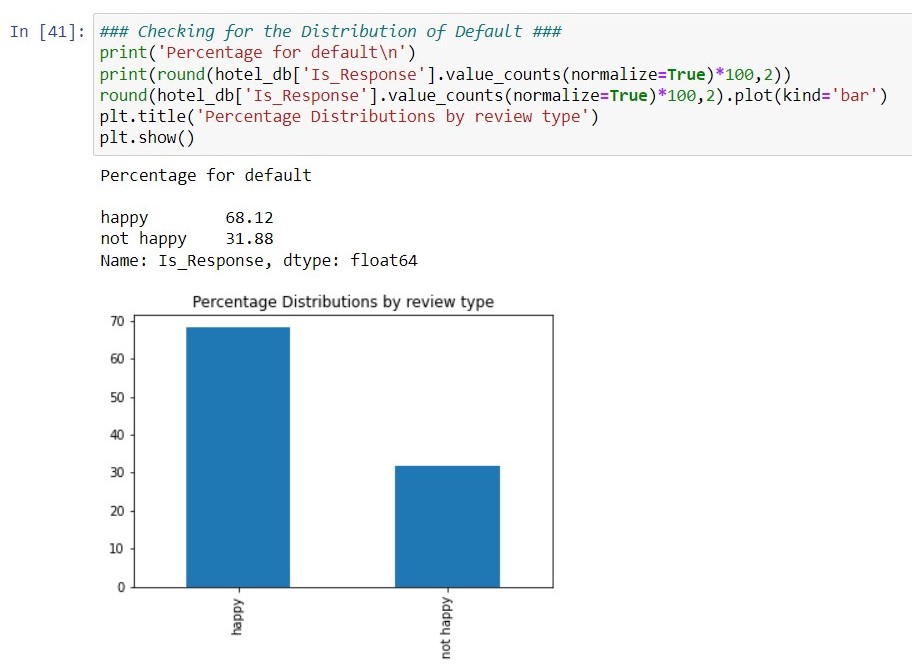


Figure 42

After that I have to clean the text for sentimental analysis so I creates a function which converts the text into lower-case, and removes square bracket, removes numbers and punctuation as well. Then I removed the stop words from the description. After doing all cleaning process my dataset look alike:



Figure 43

I create column of cleaned text because it very easy to access them. Then I used TfidfVectorizer library that is used to convert a collection of raw documents into a matrix of TF-IDF (Term Frequency-Inverse Document Frequency) features. Then I train the logistic regression model on this dataset description. Then check the **confusion matrix** of the predicted values and the actual values.



Figure 44

The 4988 signifies the count of true negatives (correctly predicted negatives). The 550 represents the count of false positives (incorrectly predicted positives). The 324 represents the count of false negatives (incorrectly predicted negatives). The 1925 represents the count of true positives (correctly predicted positives).

Then I check the accuracy of the model which is quite very good. The model accuracy is 88% which makes very good prediction on it. Then I check my model by give it a random input of the text related to the hotel then model predict it very good.

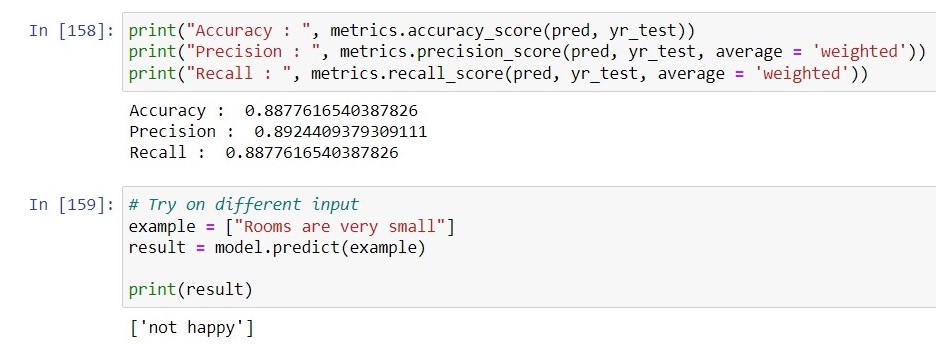


Figure 45

# Conclusion:

In order to determine constructional cost by counties in USA, information on the project type, cost by different counties, and the yearly total number of finished projects. I applied many algorithms of machine learning, but Ridge and lasso regression got low accuracy which demonstrates that the data set is insufficient for finding a solution to the issue.

On the current data set, extra features are extracted using feature engineering techniques, nevertheless the independent variable features are insufficient for more accurate estimation. The success of the model may be greatly increased by counting more specific details, such as what material used in construction, material quality. Number of people engaged in the construction area, project duration. Also have not enough dated dataset which helps us to find the duration of project. I want to compare the both dataset but the data are really different from each other and its not possible to compare them.

Furthermore, the constructional dataset is not available at wide range for analysis and Machine learning model. The main part of the research is we should find out the best data or collect dataset from the other resources.

# References:

**Datasets reference:**

USA Dataset: <https://data.world/healthdata/ca-healthcare-project-cost>

Hotel Dataset: <https://www.kaggle.com/datasets/anu0012/hotel-review>

Sanyal, S., 2022. Analytics Insight. [Online]

Available at: https://www.analyticsinsight.net/10-reasons-why-python-is-one-of-the-best-programming-languages/

Sharma, N., 2018. Towards Data Science. [Online]

Available at: <https://towardsdatascience.com/ways-to-detect-and-remove-the-outliers-404d16608dba>

K, D., 2018. Medium. [Online]

Available at: https://medium.com/@k\_dasaprakash/linear-regression-af99714eea8b

Malato, G., 2021. Towards Data Science. [Online]

Available at: <https://towardsdatascience.com/feature-selection-with-random-forest-3e1979b3a2dc>

Ronaghan, S., 2018. Towards Data Science. [Online]

Available at: <https://towardsdatascience.com/the-mathematics-of-decision-trees-random-forest-and-feature-importance-in-scikit-learn-and-spark-f2861df67e3#:~:text=The%20sum%20of%20the%20feature's,T%20%3D%20total%20number%20of%20trees>

CUEMATH, n.d. cuemath.com. [Online]

Available at: <https://www.cuemath.com/binomial-distribution-formula/>

DATA.GOV.IE, n.d. DATA.GOV.IE. [Online]

Available at: <https://data.gov.ie/dataset?theme=Housing>

Sanyal, S., 2022. Analytics Insight. [Online]

Available at: <https://www.analyticsinsight.net/10-reasons-why-python-is-one-of-the-best-programming-languages/>

PSRA, 2023. The Property Services Regulatory Authority. [Online]

Available at: <https://www.propertypriceregister.ie/>

Sharma, N., 2018. Towards Data Science. [Online]

Available at: <https://towardsdatascience.com/ways-to-detect-and-remove-the-outliers-404d16608dba>

w3schools, n.d. w3schools.com. [Online]

Available at: <https://www.w3schools.com/python/numpy/numpy_random_binomial.asp>

"Data Visualization with Python and JavaScript: Scrape, Clean, Explore & Transform Your Data" by Kyran Dale

"Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow" by Sebastian Raschka and Vahid Mirjalili

"Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems" by AurélienGéron

"Data Visualization with Python and JavaScript: Scrape, Clean, Explore & Transform Your Data" by Kyran Dale

"Python Plotting with Matplotlib: A Beginner's Guide to Plotting Data with Python" by Benjamin Root

"Python Data Science Handbook: Essential Tools for Working with Data" by Jake VanderPlas

"Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython" by Wes McKinney

"Data Science from Scratch: First Principles with Python" by Joel Grus

"Python Data Science Essentials: A Practitioner's Guide to the Top Five Python Libraries" by Alberto Boschetti and Luca Massaron

"Practical Statistics for Data Scientists: 50 Essential Concepts" by Peter Bruce and Andrew Bruce

"Applied Predictive Modeling" by Max Kuhn and Kjell Johnson