SCHOOL OF COMPUTER SCIENCE

BACHELOR OF COMPUTER SCIENCE (HONORS)

SEPTEMBER 2024 – SEMESTER 6

MODULE CODE:	**ITS 69304**
MODULE NAME:	**DATA ANALYTICS AND MACHINE LEARNING**
MODULE LEADER:	**MR. ANMOL ADHIKARI**
DEADLINE:	**09TH NOVEMBER 2024, 23:59 PM (NPT)**

INDIVIDUAL ASSIGNMENT TASK 2 (40%) **ARTIFICIAL INTELLIGENCE**

STUDENT DECLARATION: We Declare That –

- ✓ I confirm my awareness about the University's regulations, governing cheating in tests and assignments, and form the guidance issued by the School of Computing and IT concerning plagiarism and proper academic practices, and the assessed work now submitted is in accordance with this regulation and guidance.
- ✓ I understand that, unless already agreed with the School of Computing and IT, that the assessed work has not been previously submitted, either in whole or in part, in this or any other institution.
- ✓ I recognize that should evidence emerge that my work fails to comply with either of the above declarations, then I may be liable to proceeding under regulation.

CASE STUDY ON INDUSTRY

ALL SCALE ENTERPRISE IN NEPAL

STUDENT FULL NAME	UNIVERSITY ID	SIGNATURES	SCORES
SUJAL RATNA TULADHAR	**036 2483**	Sujal Ratna Tuladhar	_____ / 100

TABLE OF CONTENT

TABLE OF FIGURE

TABLE OF TABLE

TABLE OF CODE

TABLE OF EQUATIONS

CASE STUDY ON SCALED INDUSTRIES: STATISTICS OF NEPAL

Abstract:

This paper demonstrated the use of python language to empower and help the different types of scaled industries present in Nepal. The dataset is extracted from the official governmental site and advanced data analysis and visualizations is presented to help stakeholders and whom ever might be researching further on the topic. Further identifying the need of sustainable growth and business development. This research helsp future entrepreneur to leverage the wonders of python and help in combating numerous challenges and limitations found in current system of the country.

Keywords:

Industry, Enterprise, Artificial Intelligence, Machine Learning, Linear Model, Logistic Regression, Ensemble, Random Forest Classification, Accuracy, Precision, Recall, F1-Score,

1. SECTION A

1.1. Problem Identification

Economy of Nepal recently have seen improvement and incremental growth of the first half of 2024 than previous years. This is supported by the different sectors that has significant impact on helping the declining economic growth to rebound in form of services and industries. With services like accommodations, foods, tourism, financial and insurance, wholesale trades and retail shops, with industry production of hydroelectricity, agriculture, with availability of materials and favorable but declining weather conditions. Increase in domestic demands due to public consumption and inflow of more goods than remittance. Investment in private sector has been sluggish, evident from lack of capital gain. And budget has been adjusted to reduce by the government as by revisiting the previously stated revenue spending, finding lack of efficiency in execution.

1.1.1.Economic Significance

Nepal's economy has been shifting gradually from majority of population focusing on agriculture to more and more of the newer generation prioritizing the modern service industries. As the subsistence industry has decreased to half of total in last two decades. But a greater number of men have entered informal blue-collar jobs for temporary income which has seen more productivity. With half of the labor in agrarian not enough surplus is produced to accommodate in rural areas while excess waste can be found around the busier urban areas.

From the early 1980s Nepal has given emphasis to economic planning and manufacture industry as in the mid-60s such were nonexistent although efforts were made in the 30s by the government to building public enterprises industrial base. Which were later handled to private conglomerates on lease to reform the declining and preventing it from shutting down.

1.1.2.Challenges Faced

Although claims have been made that women are also included, not a lot of significant number has transitioned overall. Policies should be enforced to prepare and connect women and youth to better job options including entrepreneurship. Identify new market while integrating value-chains to increase productivity. Strengthening the guidance for all youngling to enhance living standards.

Privatization leading to slowdowns also lead to revolution and resistance from worker union as they feel their jobs were being threatened due to lack of commercial success and lacking media and updates made it look like the

market viability wasn't possible for the investor to pump money into a failing industry with overstaffed and underqualified, a serious liability in healthcare and machinery

1.2. Proposed Solution:

1.2.1.Development initiative

Together with established business leaders and cooperation with government of Nepal, different sectors and partner. Enterprise can effectively conduct activities in multi-national levels to achieve their objective that has been significantly hampered by the difficult situations, trying to revive the economy and rejuvenate the interest inside the country also that the government bodies will put forward a plan that could save the business with medium- and long-term strategies and plans. Empowering with business development. Digital support and transformation are also required with sustainably growing and optimizing the operations. With state head and province headquarter in domestic and forging diplomatic relation with international ambassadors in

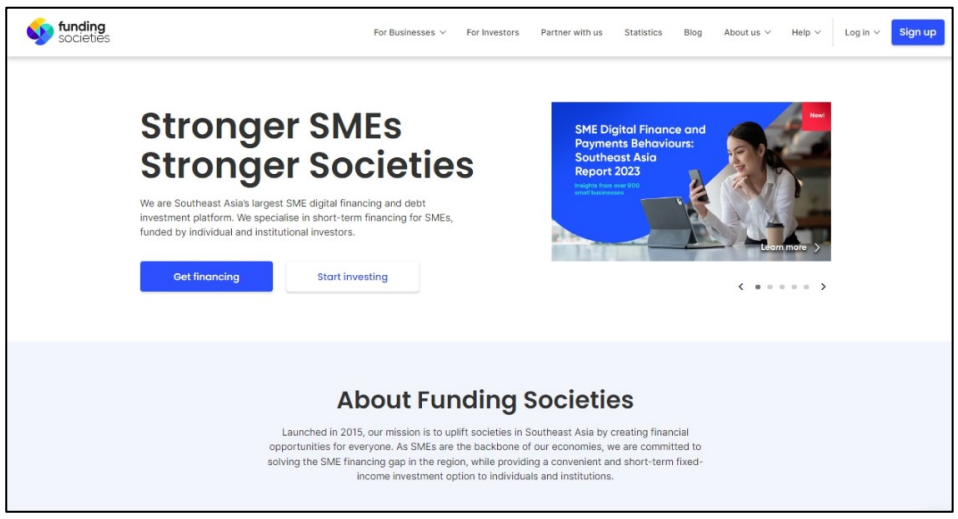
1.2.2.Solution Components

working together with government for policy and legal reform, advocating for quick and decisive actions to issues that hamper the development to flourish. Expansion to robust network of support across multiple levels ranging from international to regional/district facilitating knowledge sharing and mentorship for scalability and developing a structure approach. This will help in creating better future with a synergy between todays need and solve contemporary issues of tomorrows expectations in the public and private corporate sectors. Creating a widespread belief that Nepali industry society require urgent cleaning of plague with modern well-trained professional that can fully support and keep up with well-trained technological advancements in sectors irrespective of their investment and nature of business.

1.3. Comparison with Recent Solutions

1.3.1.Solution A: funding societies

Funding societies describe itself as the largest digital financing and debt platform for Small and Medium Enterprises although it mostly operates in South-East Asia. Similar approach can be used in South Asia market especially Nepal. SME are backbone of countries that are lacking in opportunities. Uplifting societies and region with financial gap with short-term fixed-income to individuals/institutions to grow their portfolio. This allows services in remote areas then increasing and providing access to resource, the business is able to be more authentic and succeeding without any turmoil that hamper togetherness in teamwork. Solution vendor is received upfront and tailored. Supply chain as distributor is paid earlier and payment are also in terms.

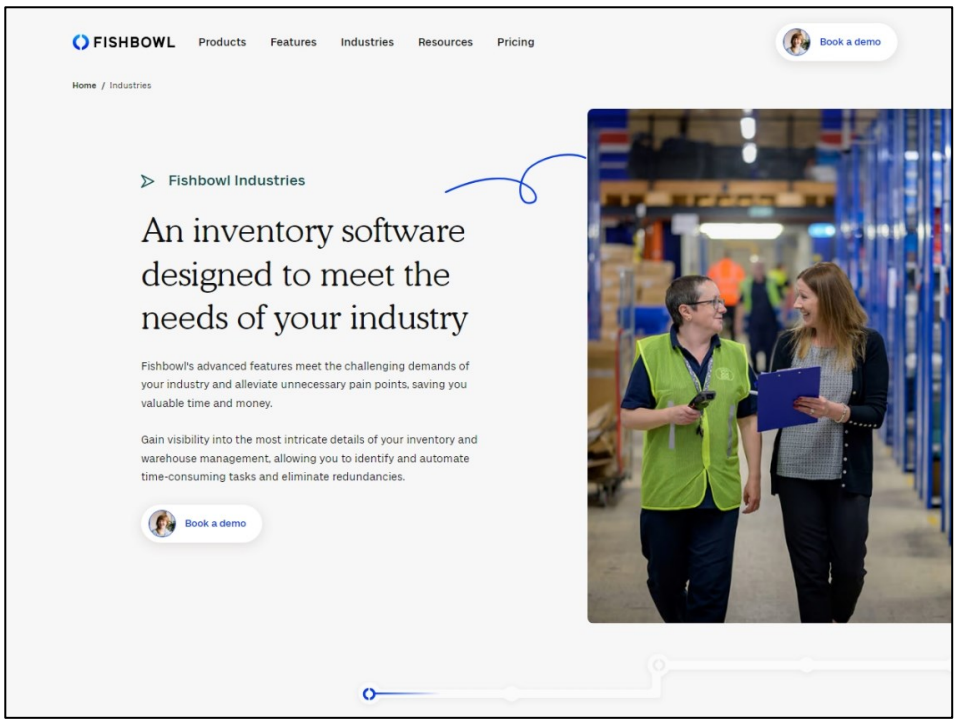


Images 1. funding societies

1.3.2.Solution B: FISHBOWL

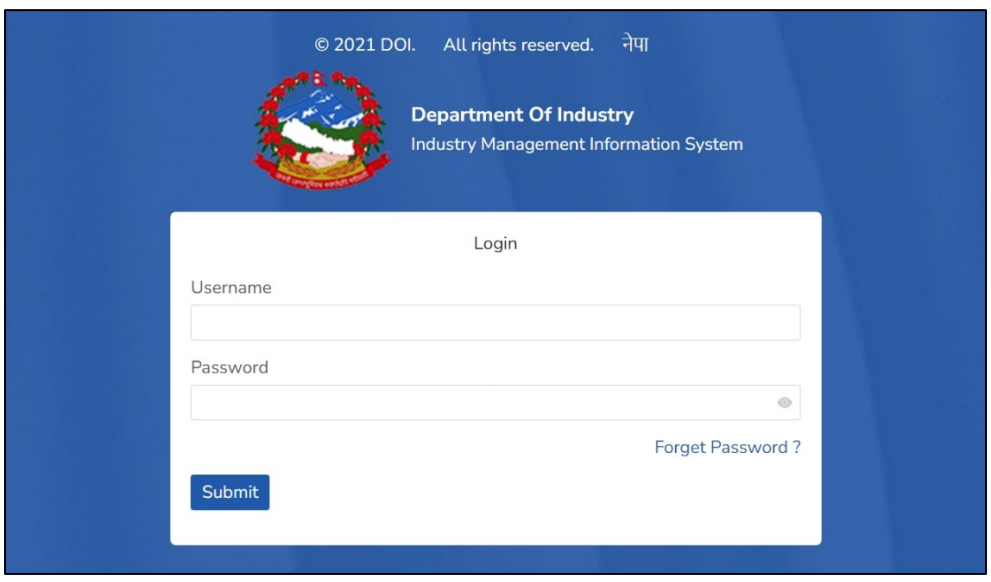
Fishbowl is a one easy inventory management software that is used by thousands of businesses for tracking their inventory, while effectively managing multiple warehouse's locations. they are integrated a data assistant named "Athena" which use artificial intelligence to get instant access to crucial business analytics and vital information is tailored with data and information provided. Challenges related to supply chain, accounting, distribution of complex workflows has been identified early and the money saved with streamlined operations

are quite efficient. Also integrates well with popular software's in a single spot



Images 2. fishbowl industry

1.3.3.Solution C: Industry Management Information System



Images 3. IMIS

official site from department of industry for management, policies and intellectual properties. this helps to manage and look over the industrial development and providing information to industry. This online service was introduced and online since June of 2022 and has been updating since then. but the site requires login information and verification

2. SECTION B

2.1. Data Modeling

Data modeling would refer to showing types of data used in this particular dataset. To show some kind of visual representation like diagrams or tables to understand relationship of data points required for business and how it can be grouped, modified and organized.

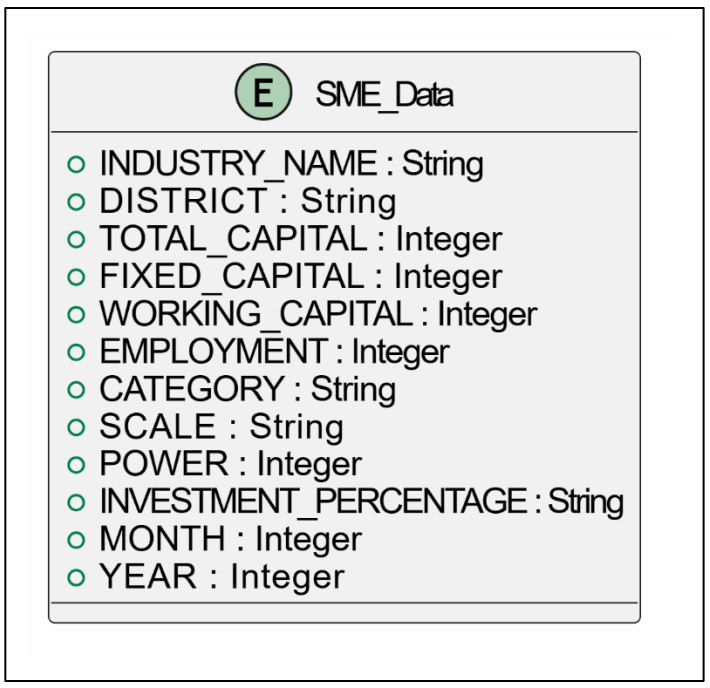
2.1.1.Identifying Key Entities

Table 1. Column's Description

COLUMN	DESCRIPTION
S. NO.	Displays the order in which industry was registered to the database of Department of Industry of Government of Nepal
REGISTRATION DATE	This is the date that the specific company registered
INDUSTRY NAME	This is the brand name that uniquely identify the business in operations also describe the sector
DISTRICT	This represents the geographic location of the registered enterprise operates with most reach
TOTAL CAPITAL	The full financial scale resources allocated to business, potential impact on employment and operational capacity help investor to gauge at it
FIXED CAPITAL	The portion of money that provide insight to long term assets; heavy like building, machinery, equipment and service oriented

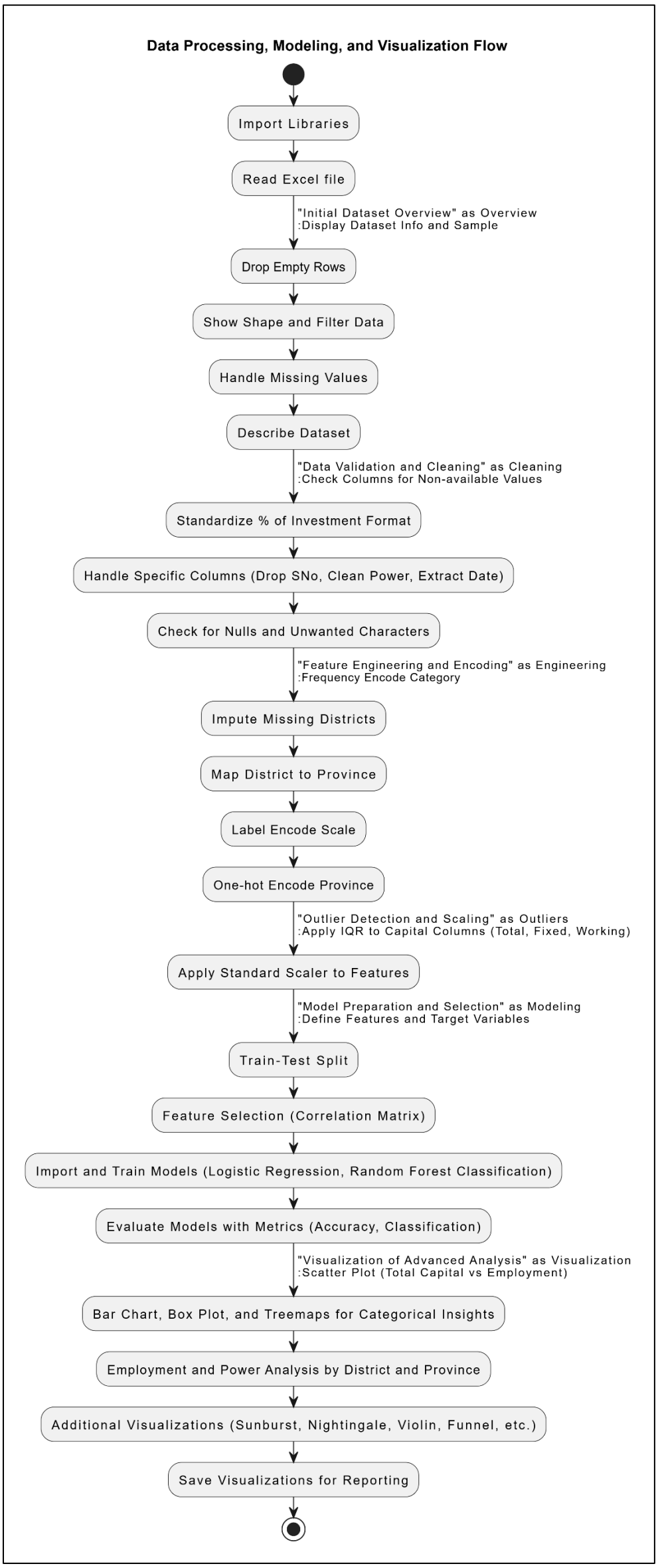
WORKING CAPITAL	This is the fund that is necessary as a short-term financial health and operational sustainable for smooth daily business liquidity and activities
EMPLOYMENT	Number of employees that are working in that particular or the work force size
CATEGORY	Type of industry that the enterprise fall under, helps in industry specific analysis 7 types
SCALE	This classifies the business size into 3 labels
POWER	Power requires or consumed for its operation, shows the resources on how much substantial kilowatt infrastructure use
% OF INVESTMENT	Shows the percentage of influence investment is it is majority foreign or local

2.1.2.Schema



Images 4. Data Schema

2.1.3.Data Flow



Images 5. Data Flow

2.2. Machine Learning Algorithms

2.2.1.Classification Algorithms

2.2.1.1. Logistic Regression

Logistic regression model is one of the most powerful statistical methods used to build relationship between one or more independent variables with a categorical dependent variable. It is primary use in classification task as its outcomes are binary meaning only one of two yes/no or 1/0. As it estimates the probability of event occurring. Its core is a logistic/sigmoid function, helping it mapping and predicting outcome, any real value that lies between the range of 1 and 0.

$$\boldsymbol{P(y = 1|X) = \frac{1}{1 + e^{-z}}}$$

Equation 1. sigmoid formula

where: $\boldsymbol{P(y = 1|X)}$ probability that event occur

$\boldsymbol{z = (B_0 + B_1X_1 + \cdots + B_kX_k)}$ is linear combination

$\boldsymbol{B_0, B_1, B_k}$ are coefficient (weights), need to be estimated

$\boldsymbol{X_1, X_k}$ are independent variables (predictors)

logit function is natural logarithm of the odds as it transforms the transforms the probability to log-odds of event occurring

$$\boldsymbol{logit(P) = \ln\left(\frac{P}{1 - P}\right) = (B_0 + B_1X_1 + \cdots + B_kX_k)}$$

Equation 2. linear combination of predictor and coefficient

maximum likelihood estimation (MLE) is used to find the values of coefficients(B) that maximize the likelihood of observing given data $\boldsymbol{L(B_0, B_1, \dots, B_k)}$

$$\boldsymbol{= \sum_{i=1}^{n} y_i \ln\big(P(y_i|X_i)\big) + (1 - y_i) \ln\big(1 - P(y_i|X_i)\big)}$$

Equation 3. maximum likelihood estimation

odd ratio is a way in interpretating the coefficients

$$\boldsymbol{OR = e^B}$$

Equation 4. odd ratio

types of logistic regression are for when depended variables. **binary:** has two possible outcomes, **ordinal:** has more than two ordered outcomes, **multinomial:** has more than two outcomes without inherent order

this makes logistic regression is an interpretable, flexible and powerful model for binary classification tasks as outcomes are transformed to probability and relationship between dependent and independent variables are modeled by using the function. the model use MLE and the performance is evaluated using metrics like accuracy, precision, recall, roc curve.

2.2.1.2. Random Forest Classifier

an ensemble method that builds on bagging (bootstrap aggregating) combines multiple decision tree predictions to produce more accurate, stable model trained on varying samples of the data (with replacement). here the decision tree model will split data into multiple subsets based on feature values in a tree structure as nodes are a feature, brands are decision rule and leaves are branded as outcome class. the model will split data using Gini impurity or entropy metrics to classify best outcome. nodes are split recursively until a minimum predefined depth is reached or when the leaf classify the data. to train each tree, $\boldsymbol{n}$ samples are randomly selected with replacement. a subset $\sqrt{\boldsymbol{m}}$ of feature is selected from total features $\boldsymbol{m}$. its tree grows until the maximum depth or minimum leaf sample.

Gini impurity is used for measuring probability of incorrect classification of random element chosen from labelled distribution of node. low score means better split.

$$Gini(D) = 1 - \sum_{i=1}^{c} P_i^2$$

Equation 5. gini impurity

D is the dataset at a node, c is the number of classes, P_i is probability of that instance being in class i

$$Gini(D) = 2p(1-p)$$

Equation 6. gini in binary classification

entropy is also measurement of impurity as it quantification of level of disorder or uncertainty in dataset

$$Entropy(D) = -\sum_{i=1}^{c} p_i \log_2(p_i)$$

Equation 7. entropy measurement

out of bag error is the bootstrap sample that were not included during training sample for validating the model as a built-in method so average across give unbiased estimate of accuracy.

$$= \frac{1}{N}\sum_{i=1}^{N} \| (\hat{y}OOB(X_i) \neq y_i)$$

Equation 8. out of bag error

where: N is the total number of data samples, $\hat{y}OOB(X_i)$ predicted label of (X_i) from OOB, $\hat{y}$ is true label of X_i, $\|$ is indicator function

prediction aggregation determined by taking majority vote across all trees $\hat{y} = mode(\{T_t(X)\}_{t=1}^{n})$

where: X is input feature vector, $T_t(X)$ is prediction o

feature importance is the ability to measure the importance of feature to reduce Gini impurity or entropy the most splits are deemed more important.

$$Feature\ Important(f) = \frac{1}{T}\sum_{t=1}^{T} I_t(f)$$

Equation 9. feature importance

where: T is total number of trees in forest

$I_t(f)$ importance of feature f in tree

information gain can be calculated with

$$= Entropy(parent) - \sum_{j} \frac{|D_j|}{|D|} \times Entropy(D_j)$$

Equation 10. information gained

where: D is the dataset before split, $D_j$is subset of data after split, $|D|$and $|D_j|$are size of datasets.

by aggregating trees trained on multiple random subsets of data and features it is less prone to overfitting, and suitable for higher dimensional complex and unseen datasets providing insights of features with importance for better predictions by estimating error

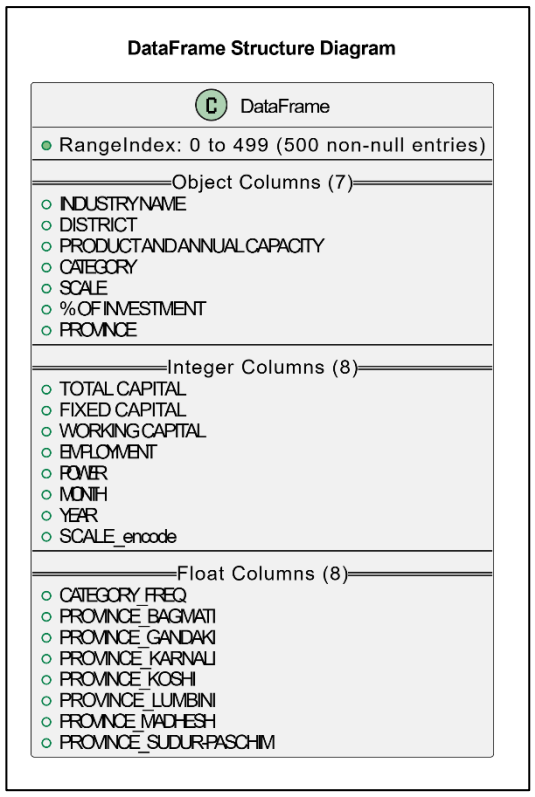
2.3. Requirements for Machine Learning Implementation

2.3.1.Data Requirements

machine learning algorithm typically require numerical data due to it mathematical computations to identify relationships and pattern form the data. in their raw form the text values like category cannot be directly processed.

so, machine learning works directly under numerical values so it be optimized and adjusted with weights during training. for this current project probabilistic model like logistic regression also require numerical representation inputs to calculate the probabilities.

here, the non-numeric inputs like categories in the data set must be transformed to numeric format before it is fed to the machine learning algorithm, we can use encoding techniques like label where unique integer is assigned, ordinal where the inherent order matters, one-hot when



Images 6. data frame structure diagram

2.3.2.Technical Requirements

Python is multipurpose language also huge in AI/DS/ML communities. praised for their intuitive, easy to understand and large and comprehensive open-source libraries backed by extensibility.

Google Colaboratory is the platform chosen to work on the assignment as it allows users to take advantage of using python as main executing programming language which is top in data analysis and machine learning, along with rich plethora of charts, images, html, latex in a single document. it is an **Interactive Python Notebook** for statistical simulations and mathematical computations required for advanced data analysis.

Pandas is a popular library used for manipulation, transformation, and other practical functions are in use to by data analytics and scientists for their work flow.

RE is a package for handling text related data, and the searching of pattern and sequence of strings were made easy to clean categorical datas.

Random is module to generate distributions of elements.

SciPy is extension of NumPy providing fundamental algorithms used in in-depth algebraic, statistical, optimizations, and more to mathematicians, scientists and engineering.

Plotly is a library for visualizations to create interactive figures, plots.

Sci-kit Learn is machine learning module. it is a multi-functional toolkit for with powerful and state-of-the-art models

2.3.3.Evaluation Metrics

first of all, True and False Positives and Negatives are used as useful metrics to calculate and evaluate several models. these are calculated at single fixed threshold

Accuracy is the proportion of all instances that were correctly classified. it is a coarse-gained measure of model quality for unspecified general model

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{correct}}{\boldsymbol{total}} = \frac{\boldsymbol{TP + TN}}{\boldsymbol{TO}}$$

Equation 11. accuracy

Precision is the proportion of all models true positive classifications among all positive prediction. precision increase as false positive decrease

$$\boldsymbol{Precision} = \frac{\boldsymbol{correctly\ classified\ actual\ positive}}{\boldsymbol{everthing\ classified\ as\ positive}} = \frac{\boldsymbol{TP}}{\boldsymbol{TP + FP}}$$

Equation 12. precision

Recall (Sensitivity) is the proportion of true positive predcitions among all actual positives classified correctly aka True Positive Rate. recall increase when false negative decrease

$$\boldsymbol{Recall} = \frac{\boldsymbol{correctly\ classified\ actual\ positive}}{\boldsymbol{all\ actual\ positives}} = \frac{\boldsymbol{TP}}{\boldsymbol{TP + FN}}$$

Equation 13. recall

False Positive Rate is the proportion of all actual negatives classified incorrectly as positives, aka false

alarm. these actual negatives are measures the fractions ligitmate and misclassified.

$$FPR = \frac{incorrectly\ classfied\ actual\ negatives}{all\ actual\ negatives}$$

$$= \frac{FP}{FP + TN}$$

Equation 14. false positive rate

F1-Score is identified as the harmonic mean of recall and precision

$$F1 = 2 \times \frac{Precison \times Recall}{Precision + Recall}$$

Equation 15. F1-Score

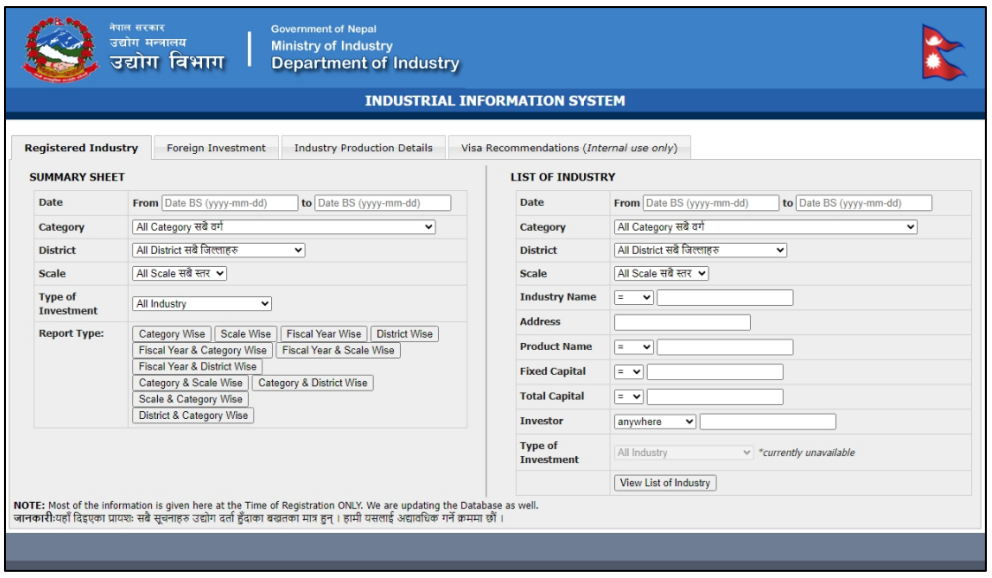
ROC AUC are curve plots the TPR (recall) against FPR and shows the measure of performance and discrimination ability, closer to 1 indicating a better model performance.

3. SECTION C

3.1. Collection of Data and Pre-Processing

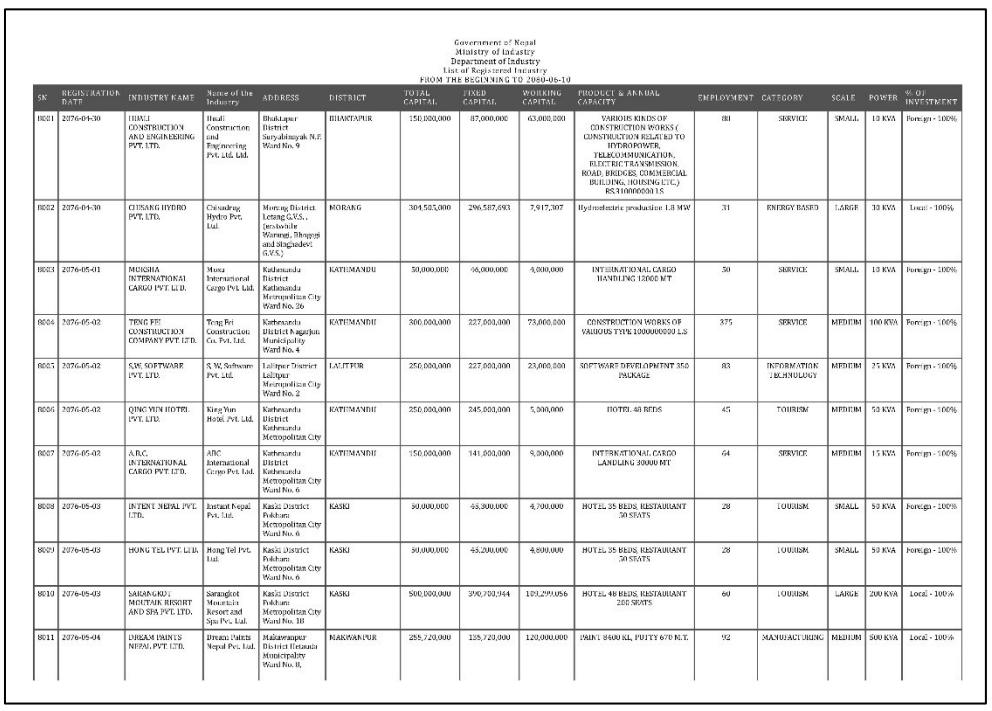
3.1.1.Sources

initially, we were informed about the requirements of the data collection process. later due to restrictions and problems we could scour online for datasets but those had to be unique and not readily available making sure it lined up with the requirement of the guidelines and objective provided to us. site like 'GitHub', 'Kaggle' were thoroughly searched and even considered options like 'web-scrapping'. later decided to dig into many open-source sites like 'open data nepal', 'nepal in data'. which lead to direct sources sites like 'nepal rastra bank', 'national data portal', 'central bank of nepal', 'national statistics office' which were official sites where dataset is provided form the government of Nepal and the respective departments and ministry are aggregated.



Images 7. industrial statistic system

while searching inside reports published by 'government of nepal, minister and department of industry' on the site final dataset was decided after checking a link '**113. 199. 192. 99 :8080/list_industry .php?page=**' which is the link to 'industrial information system' and has 'list of all registered industry' from the beginning year of 2018 BS till latest 2080 BS, spread across 19 pages. the total number of records found at the time of collection is 9147, which was collected in batch of 500 count. although it was in pdf format it wasn't a problem as many online sites provide free conversion to required format.


Government of Nepal
Ministry of Industry
Department of Industry
List of Registered Industry
FROM THE BEGINNING TO 2080-06-10

SN	REGISTRATION DATE	INDUSTRY NAME	Name of the Industry	ADDRESS	DISTRICT	TOTAL CAPITAL	FIXED CAPITAL	WORKING CAPITAL	PRODUCT & ANNUAL CAPACITY	EMPLOYMENT	CATEGORY	SCALE	POWER	% OF INVESTMENT
8021	2076-04-30	HALI CONSTRUCTION AND ENGINEERING PVT. LTD.	Hali Construction and Engineering Pvt. Ltd. Ltd.	Bhaktapur District Suryabinayak N.P. Ward No. 9	BHAKTAPUR	150,000,000	87,000,000	63,000,000	VARIOUS KINDS OF CONSTRUCTION WORKS (CONSTRUCTION RELATED TO HYDROPOWER, TELECOMMUNICATION, ELECTRICTRANSMISSION ROAD, BRIDGES, COMMERCIAL BUILDING, HOUSING ETC.) RS3100000015	80	SERVICE	SMALL	10 KVA	Foreign - 100%
8022	2076-04-30	CHISANG HYDRO PVT. LTD.	Chisanbng Hydro Pvt. Ltd.	Morang District Letang G.V.S., (mahendra Waimai, Bhogati and Singhadevi G.V.S.)	MORANG	304,585,000	296,587,693	7,917,307	Hydroelectric production 1.8 MW	31	ENERGY BASED	LARGE	10 KVA	Local - 100%
8023	2076-05-01	MOKSHA INTERNATIONAL CARGO PVT. LTD.	Moksa International Cargo Pvt. Ltd.	Kathmandu District Kathmandu Metropolitan City Ward No. 26	KATHMANDU	50,000,000	46,000,000	4,000,000	INTERNATIONAL CARGO HANDLING 15000 MT	50	SERVICE	SMALL	10 KVA	Foreign - 100%
8024	2076-05-02	TENG FEI CONSTRUCTION COMPANY PVT. LTD.	Teng Fei Construction Co. Pvt. Ltd.	Kathmandu Kathmandu District Nagarjun Municipality Ward No. 4	KATHMANDU	300,000,000	227,000,000	73,000,000	CONSTRUCTION WORKS OF VARIOUS TYPE 1000000000 LS	375	SERVICE	MEDIUM	100 KVA	Foreign - 100%
8025	2076-05-02	SJIC SOFTWARE PVT. LTD.	S. W. Software Pvt. Ltd.	Lalitpur District Lalitpur Metropolitan City Ward No. 2	LALITPUR	250,000,000	227,000,000	23,000,000	SOFTWART DEVELOPMENT 350 PACKAGE	83	INFORMATION TECHNOLOGY	MEDIUM	25 KVA	Foreign - 100%
8026	2076-05-02	QING YUN HOTEL PVT. LTD.	King Yun Hotel Pvt. Ltd.	Kathmandu District Kathmandu Metropolitan City	KATHMANDU	250,000,000	245,000,000	5,000,000	HOTEL 48 BEDS	45	TOURISM	MEDIUM	50 KVA	Foreign - 100%
8027	2076-05-02	A.R.C. INTERNATIONAL CARGO PVT. LTD.	ARC International Cargo Pvt. Ltd.	Kathmandu District Kathmandu Metropolitan City Ward No. 6	KATHMANDU	150,000,000	141,000,000	9,000,000	INTERNATIONAL CARGO LANDLING 30000 MT	64	SERVICE	MEDIUM	15 KVA	Foreign - 100%
8028	2076-05-03	INTENT NEPAL PVT. LTD.	Intent Nepal Pvt. Ltd.	Kaski District Pokhara Metropolitan City Ward No. 6	KASKI	50,000,000	45,300,000	4,700,000	HOTEL 25 BEDS, RESTAURANT 50 SEATS	28	TOURISM	SMALL	50 KVA	Foreign - 100%
8029	2076-05-03	HONG YEL PVT. LTD.	Hong Yel Pvt. Ltd.	Kaski District Pokhara Metropolitan City Ward No. 6	KASKI	50,000,000	45,200,000	4,800,000	HOTEL 25 BEDS, RESTAURANT 50 SEATS	28	TOURISM	SMALL	50 KVA	Foreign - 100%
8030	2076-05-03	SARANGKOT MOUTAIN RESORT AND SPA PVT. LTD.	Sarangkot Mountain Resort and Spa Pvt. Ltd.	Kaski District Pokhara Metropolitan City Ward No. 18	KASKI	500,000,000	396,700,944	103,299,056	HOTEL 48 BEDS, RESTAURANT 200 SEATS	60	TOURISM	LARGE	200 KVA	Local - 100%
8031	2076-05-04	DREAM PAINTS NEPAL PVT. LTD.	Dream Paints Nepal Pvt. Ltd.	Makawanpur District Hetauda Municipality Ward No. 8,	MAKWANPUR	255,720,000	135,720,000	120,000,000	PAINT 9600 KL, PUTTY 670 M.T.	92	MANUFACTURING	MEDIUM	500 KVA	Local - 100%

Images 8. dataset in pdf format

the converted excel sheet was later cleaned within excel to make it easy to us for further and advanced analysis. the un recognizable word which were converted to image were removed and the respective columns were removed. the completely empty rows were removed.

3.1.2.Loading

```
#@title pandas and read specific excel
import pandas as pd
df = pd.read_excel('0_8001-8500.xlsx', sheet_name='Sheet1')
```

Code 1. import pandas and read excel sheet

pandas library was imported and specific sheet from excel file was read

3.1.3.Structure

```
#@title set option to display max
pd.set_option('display.max_columns', None)
# pd.set_option('display.max_rows', None)
```

Code 2. max column set option

option to display max columns was selected, none means all columns would be shown

```
# @title shape
df.shape
```

```
(500, 13)
```

Code 3. shape row and columns

shape of the dataset was seen as part of initial look

```
#@title info
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 13 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   S. NO.                        500 non-null    float64
 1   REGISTRATION DATE             500 non-null    datetime64[ns]
 2   INDUSTRY NAME                 500 non-null    object
 3   DISTRICT                      486 non-null    object
 4   TOTAL CAPITAL                 500 non-null    int64
 5   FIXED CAPITAL                 500 non-null    int64
 6   WORKING CAPITAL               500 non-null    int64
 7   PRODUCT AND ANNUAL CAPACITY   500 non-null    object
 8   EMPLOYMENT                    500 non-null    int64
 9   CATEGORY                      500 non-null    object
 10  SCALE                         500 non-null    object
 11  POWER                         500 non-null    object
 12  % OF INVESTMENT               500 non-null    object
dtypes: datetime64[ns](1), float64(1), int64(4), object(7)
memory usage: 50.9+ KB
```

Code 4. information

info function is used to check range of entries, total columns, their names, non-null count, data types, and memory used

3.1.4.Preprocessing

3.1.4.1. Validation

```
#@title sample original dataset
display(df.sample(4))
```

	S. NO.	REGISTRATION DATE	INDUSTRY NAME	DISTRICT	TOTAL CAPITAL	FIXED CAPITAL	WORKING CAPITAL
452	8453.0	2078-03-22	RADHAKRISHNAHOMEAPPLIANCESPVT.LTD.	KATHMANDU	50000000	38000000	12000000
38	8039.0	2076-06-08	XIANGLUYUANHOTELPVT.LTD.	LALITPUR	250000000	242000000	8000000
141	8142.0	2076-08-26	SHYAMILAIRONANDSTEELINDUSTRIESPVT.LTD.	JHAPA	200000000	150300000	49700000
117	8118.0	2076-08-15	SPRINGHILLHOTELPVT.LTD.	NaN	50000000	47000000	3000000

Code 5. sample

sample function is used to choose and show the variability in the dataset

```
#@title display non available
display(df1.isna().sum())
```

```
S. NO.                         0
REGISTRATION DATE              0
INDUSTRY NAME                  0
DISTRICT                      14
TOTAL CAPITAL                  0
FIXED CAPITAL                  0
WORKING CAPITAL                0
PRODUCT AND ANNUAL CAPACITY    0
EMPLOYMENT                     0
CATEGORY                       0
SCALE                          0
POWER                          0
% OF INVESTMENT                0
dtype: int64
```

Code 6. not available

isna is used to show sum count of rows that value are not available. here 14 districts seem to be missing which will be handled later

```
#@title describe df1
display(df1.describe())
```

	S. NO.	REGISTRATION DATE	TOTAL CAPITAL	FIXED CAPITAL	WORKING CAPITAL	EMPLOYMENT
count	500.000000	500	5.000000e+02	5.000000e+02	5.000000e+02	500.000000
mean	8250.500000	2076-11-05 13:14:52.800000	6.454023e+08	5.805375e+08	6.486473e+07	63.918000
min	8001.000000	1978-03-01 00:00:00	2.135500e+06	8.855000e+05	9.000000e+05	0.000000
25%	8125.750000	2076-08-18 00:00:00	1.100000e+08	9.100000e+07	9.419560e+06	30.000000
50%	8250.500000	2077-03-28 12:00:00	2.000000e+08	1.513500e+08	2.585000e+07	45.000000
75%	8375.250000	2077-11-03 18:00:00	3.676000e+08	2.565223e+08	6.212541e+07	70.000000
max	8500.000000	2078-05-18 00:00:00	1.762406e+10	1.757100e+10	2.471689e+09	550.000000
std	144.481833	NaN	1.526715e+09	1.494585e+09	1.659136e+08	65.985236

Code 7. describe

describe function is used to check the central tendency, min-max, mean, count and standard deviation of numerical values present

```
#@title scale info and value count
display(df1['SCALE'].info())
display(df1['SCALE'].value_counts())
```

```
<class 'pandas.core.series.Series'>
RangeIndex: 500 entries, 0 to 499
Series name: SCALE
Non-Null Count  Dtype
--------------  -----
500 non-null    object
dtypes: object(1)
memory usage: 4.0+ KB
None
SCALE
SMALL     216
MEDIUM    183
LARGE     101
Name: count, dtype: int64
```

Code 8. specific coulumn infro

value count is used to check for count of categories.

```
#@title category info and value count
display(df1['CATEGORY'].info())
display(df1['CATEGORY'].value_counts())
```

```
<class 'pandas.core.series.Series'>
RangeIndex: 500 entries, 0 to 499
Series name: CATEGORY
Non-Null Count  Dtype
--------------  -----
500 non-null    object
dtypes: object(1)
memory usage: 4.0+ KB
None
CATEGORY
MANUFACTURING             155
TOURISM                   129
SERVICE                    83
ENERGY                     72
INFORMATION TECHNOLOGY     25
AGR0AND\nFORESTRY          20
AGRO AND FORESTRY          13
INFRASTRUCTURE              2
MINERAL                     1
Name: count, dtype: int64
```

Code 9. specific column value count

```
#@title check data type
df2.dtypes
```

```
Out[21]:
S. NO.                          float64
REGISTRATION DATE        datetime64[ns]
INDUSTRY NAME                   object
DISTRICT                        object
TOTAL CAPITAL                    int64
FIXED CAPITAL                    int64
WORKING CAPITAL                  int64
PRODUCT AND ANNUAL CAPACITY     object
EMPLOYMENT                       int64
CATEGORY                        object
SCALE                           object
POWER                           object
% OF INVESTMENT                 object
dtype: object
```

Code 10. dataset data types

dtypes is used to check datatype of the columns

3.1.4.2. Cleaning

```
#@title regular expression
import re
```

Code 11. regular expression

regular expression library is imported to

```
#@title remove repeating liability of stakeholder and cleaninf

df2['INDUSTRY NAME'] = df2['INDUSTRY NAME'].replace(r'PVT\.|LTD\.|\n', '', regex=True) \
                                           .replace(r'\s+', ' ', regex=True) \
                                           .str.strip()
```

Code 12. clean industry name

remove the repeating words describing if the industry liability of stakeholders

```
#@title turn % of investment to standard format
def clean_investment_percentage(value):
    value = str(value).replace('\n', ' ').replace('0Zo', ' ').replace('0/o', ' ').replace('/', ' ')
    value = re.sub(r'\s+', ' ', value).strip()  # Remove extra spaces
    local_match = re.search(r'Local\s*[---:\s]?\s*([\d.,]+)%', value)
    foreign_match = re.search(r'Foreign\s*[---:\s]?\s*([\d.,]+)%', value)

    local_pct = local_match.group(1).replace(',', '') if local_match else '0'
    foreign_pct = foreign_match.group(1).replace(',', '') if foreign_match else '0'

    return f"Local - {local_pct}%, Foreign - {foreign_pct}%"

df2['% OF INVESTMENT'] = df2['% OF INVESTMENT'].apply(clean_investment_percentage)
```

Code 13. clean % of investment

this function cleans the distribution of investment by the local or foreign in percent shares and the output is shown below kept in a standard format.

```
% OF INVESTMENT
Local - 100%, Foreign - 0%            269
Local - 0%, Foreign - 100%            196
Local - 0%, Foreign - 0%                5
Local - 40%, Foreign - 60%              4
Local - 15%, Foreign - 85%              3
Local - 51%, Foreign - 49%              3
Local - 20%, Foreign - 80%              3
Local - 50%, Foreign - 50%              2
Local - 10%, Foreign - 90%              2
Local - 6.25%, Foreign - 93.75%         2
Local - 66.42%, Foreign - 33.58%        1
Local - 36%, Foreign - 64%              1
Local - 34.221%, Foreign - 65.779%      1
Local - 5.67%, Foreign - 94.33%         1
Local - 9.91%, Foreign - 0%             1
Local - 15.24%, Foreign - 84.76%        1
Local - 13.04%, Foreign - 86.96%        1
Local - 49%, Foreign - 51%              1
Local - 16.667%, Foreign - 83.333%      1
Local - 20.29%, Foreign - 79.71%        1
Local - 6%, Foreign - 94%               1
Name: count, dtype: int64
```

Code 14.% of investment unique

```
#@title only keep numbers from power
df2['POWER'] = df2['POWER'].apply(lambda x: re.sub(r'\D', '', str(x)))
```

Code 15. clean power consumption

the unit was removed from the perceived power consumed by a single company or brand measured in KVA, so only numeric value is extracted

```
#@title clean category
df2['CATEGORY'] = df2['CATEGORY'].str.replace('AGR0AND\nFORESTRY', 'AGRO AND FORESTRY')
df2['CATEGORY'].unique()
```

```
array(['SERVICE', 'ENERGY', 'INFORMATION TECHNOLOGY', 'TOURISM',
       'MANUFACTURING', 'AGRO AND FORESTRY', 'MINERAL', 'INFRASTRUCTURE'],
      dtype=object)
```

Code 16. category replacement and unique

category was cleaned to be unique and defining

3.1.5.Feature Engineering

```
#@title fix date column extract year and month

# convert  to a string
df2['REGISTRATION DATE'] = df2['REGISTRATION DATE'].astype(str)

# date part
df2['MONTH'] = df2['REGISTRATION DATE'].str[5:7].astype(int)

# year part
df2['YEAR'] = df2['REGISTRATION DATE'].str[:4]
df2['YEAR'] = pd.to_numeric(df2['YEAR'], errors='coerce')
df2['YEAR'] = df2['YEAR'].astype('Int64')

# drop original
df2 = df2.drop('REGISTRATION DATE', axis=1)
```

Code 17. year month extracted from date

month and year is extracted from registration date

```
#@title split investment
def split_investment(investment_str):
    local_pct = 0
    foreign_pct = 0
    try:
      parts = investment_str.split(',')
      for part in parts:
          if 'Local' in part:
              local_pct = float(part.split('-')[1].replace('%', '').strip())
          elif 'Foreign' in part:
              foreign_pct = float(part.split('-')[1].replace('%', '').strip())
    except:
        print(f"Error processing: {investment_str}")
        return local_pct, foreign_pct
    return local_pct, foreign_pct

df4[['Local_Investment', 'Foreign_Investment']] = df2['% OF INVESTMENT'].apply(lambda x: pd.Series(split_investment(x)))
```

Code 18. split investment

investment percent was split into local and foreign

	% OF INVESTMENT	Local_Investment	Foreign_Investment
362	Local - 100%, Foreign - 0%	100.000	0.000
190	Local - 100%, Foreign - 0%	100.000	0.000
358	Local - 0%, Foreign - 100%	0.000	100.000
181	Local - 100%, Foreign - 0%	100.000	0.000
92	Local - 16.667%, Foreign - 83.333%	16.667	83.333

Code 19. check split in float

3.1.5.1. Imputation

```
#@title random
import random
```

Code 20. import random

random function is imported

```
# @title list of districts to randomly impute
districts_to_impute = ['DOLPA', 'MUGU', 'HUMLA', 'JUMLA', 'SALYAN', 'JAJARKOT', 'DAILEKH', 'SURKHET', 'KALIKOT']
```

Code 21. impute districts of karnali

these districts were missing from the data which is from Karnali province

```
# @title randomly impute districts to the identified rows
for index in rows_to_impute.index:
    df2.loc[index, 'DISTRICT'] = random.choice(districts_to_impute)
```

Code 22. randomly impute to rows

then impute into the rows at random

```
#@title district and region dictionary
province_district = {
    'KOSHI': ['TAPLEJUNG', 'SANKHUWASABHA', 'SOLUKHUMBU', 'UDAYAPUR', 'PANCHTHAR', 'ILAM', 'TERHATHUM',
              'DHANKUTA', 'BHOJPUR', 'KHOTANG', 'OKHALDHUNGA', 'JHAPA', 'MORANG', 'SUNSARI'],
    'MADHESH': ['MAHOTTARI', 'RAUTAHAT', 'DHANUSHA', 'SIRAHA', 'BARA', 'SARLAHI', 'PARSA', 'SAPTARI'],
    'BAGMATI': ['DOLKHA', 'SINDHUPALCHOWK', 'RASUWA', 'MAKWANPUR', 'BHAKTAPUR', 'LALITPUR',
                'KATHMANDU', 'NUWAKOT','RAMECHHAP', 'KAVRE', 'DHADING', 'SINDHULI', 'CHITWAN'],
    'GANDAKI': ['MANANG', 'MUSTANG', 'PARBAT', 'SYANGJA','TANAHU', 'LAMJUNG', 'BAGLUNG', 'KASKI','MYAGDI', 'GORKHA', 'NAWALPARASI'],
    'LUMBINI': ['PALPA', 'ARGHAKHACHI', 'RUKUM', 'GULMI', 'PYUTHAN', 'ROLPA', 'RUPANDEHI', 'KAPILBASTU','DANG', 'BANKE', 'BARDIYA'],
    'KARNALI': ['DOLPA', 'MUGU', 'HUMLA', 'JUMLA', 'SALYAN', 'JAJARKOT', 'DAILEKH', 'SURKHET', 'KALIKOT'],
    'SUDUR-PASCHIM': ['BAJURA', 'BAJHANG', 'DARCHULA', 'ACHHAM', 'DOTI', 'BAITADI', 'KAILALI', 'KANCHANPUR']
}
```

Code 23. dictionary of province and districts

```
#@title apply function to map district to province
def map_district_to_province(district):
    for province, districts in province_district.items():
        if district in districts:
            return province
    return None

df2['PROVINCE'] = df2['DISTRICT'].apply(map_district_to_province)
```

Code 24. function to map

dictionary mapping is used to map districts to their respective province in a function

```
#@title list null
print(df2[df2['PROVINCE'].isnull()]['DISTRICT'].unique().tolist())
```

```
[]
```

Code 25. check for null and unique in district then print list

checking for the list of empty province and unique districts

3.1.6.Preparation

3.1.6.1. Encoding

```
#@title apply frequency encoding to category
district_counts = df3['CATEGORY'].value_counts(normalize=True)
df3['CATEGORY_FREQ'] = df3['CATEGORY'].map(district_counts)
display(df3[['CATEGORY','CATEGORY_FREQ']].value_counts())
```

```
CATEGORY                CATEGORY_FREQ
MANUFACTURING           0.310           155
TOURISM                 0.258           129
SERVICE                 0.166            83
ENERGY                  0.144            72
AGRO AND FORESTRY       0.066            33
INFORMATION TECHNOLOGY  0.050            25
INFRASTRUCTURE          0.004             2
MINERAL                 0.002             1
Name: count, dtype: int64
```

Code 26. frequency encode category

frequency encoding is used in category columns as it had to count total distribution of 500 entries across 8 categories

```
#@title import label encoding for scale

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
df3['SCALE_encode'] = le.fit_transform(df3['SCALE'])
df3['SCALE_encode'] = df3['SCALE_encode'] + 1

display(df3[['SCALE','SCALE_encode']].value_counts())
```

```
SCALE   SCALE_encode
SMALL   3               216
MEDIUM  2               183
LARGE   1               101
Name: count, dtype: int64
```

Code 27. label encode scale

label encoding is used in scale columns and they have importance in priority and hierarchy

```
#@title reduceing district to province
display(df3['PROVINCE'].value_counts())
```

```
PROVINCE
BAGMATI          267
GANDAKI           67
KOSHI             63
MADHESH           41
LUMBINI           38
KARNALI           14
SUDUR-PASCHIM     10
Name: count, dtype: int64
```

Code 28. check value count of province

```
#@title import one hot encoding for province
from sklearn.preprocessing import OneHotEncoder

ohe = OneHotEncoder(sparse_output=False, handle_unknown='ignore')

ohe.fit(df3[['PROVINCE']])

encoded_data = ohe.transform(df3[['PROVINCE']])

feature_names = ohe.get_feature_names_out(['PROVINCE'])
for i, feature_name in enumerate(feature_names):
    df3[feature_name] = encoded_data[:, i]
```

Code 29. apply one hot encoding to province

one hot encoding is used in province column as location has no ranking

3.1.7.Reduction

3.1.7.1. Outliers

```
#@title import scipy
from scipy.stats import iqr
```

Code 30. import interquartile range from scipy stats

interquartile range from scipy stats is imported

```
#@title apply interquartile range with scipy in total capital. fixed cap

# calculate IQR for specified columns
for col in ['TOTAL CAPITAL', 'FIXED CAPITAL', 'WORKING CAPITAL']:
    q1 = df4[col].quantile(0.25)
    q3 = df4[col].quantile(0.75)
    iqr_val = iqr(df4[col])
    print(f"IQR for {col}: {iqr_val}\n")

    # filtering out outliers
    upper_bound = q3 + 1.5 * iqr_val
    lower_bound = q1 - 1.5 * iqr_val
    df4 = df4[(df4[col] >= lower_bound) & (df4[col] <= upper_bound)]
```

Code 31. formula to calculate iqr and fucntion

IQR is used on the three capitals total, fixed, and working capital

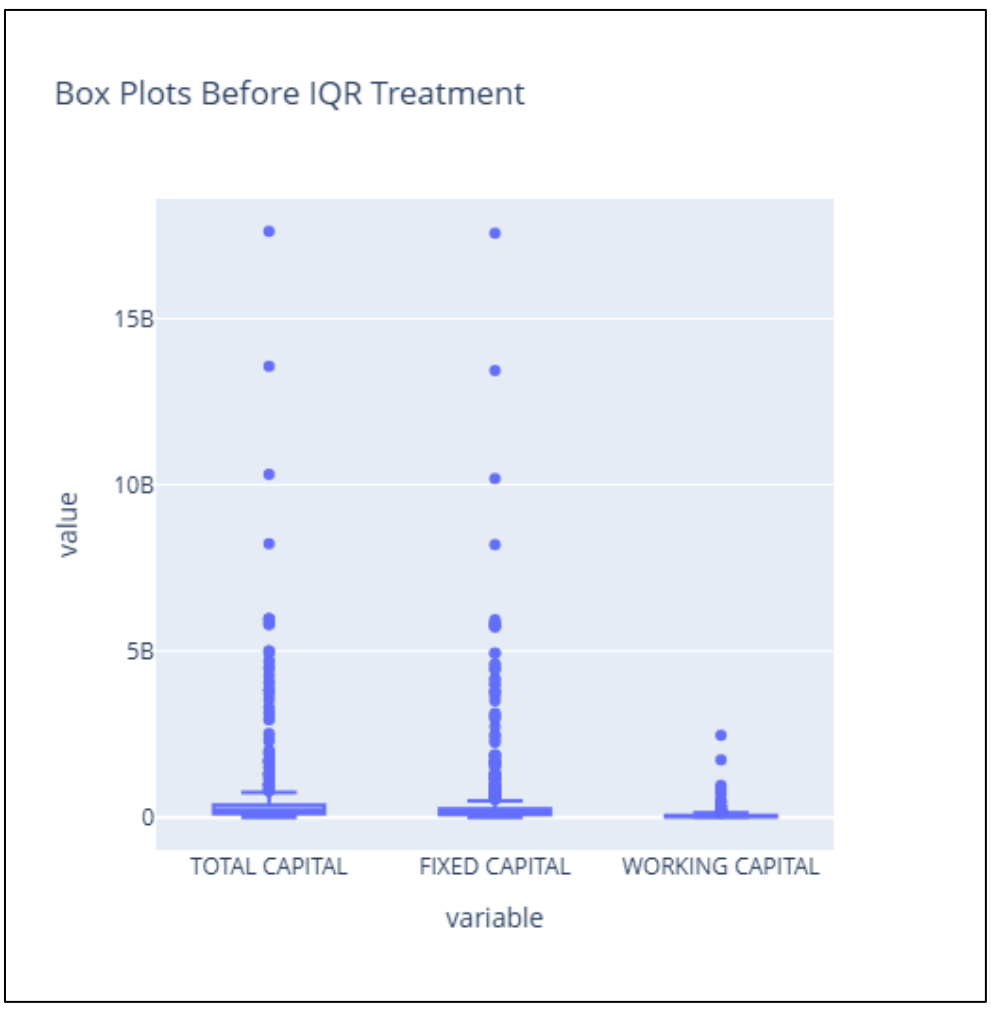


Figure 1. before iqr

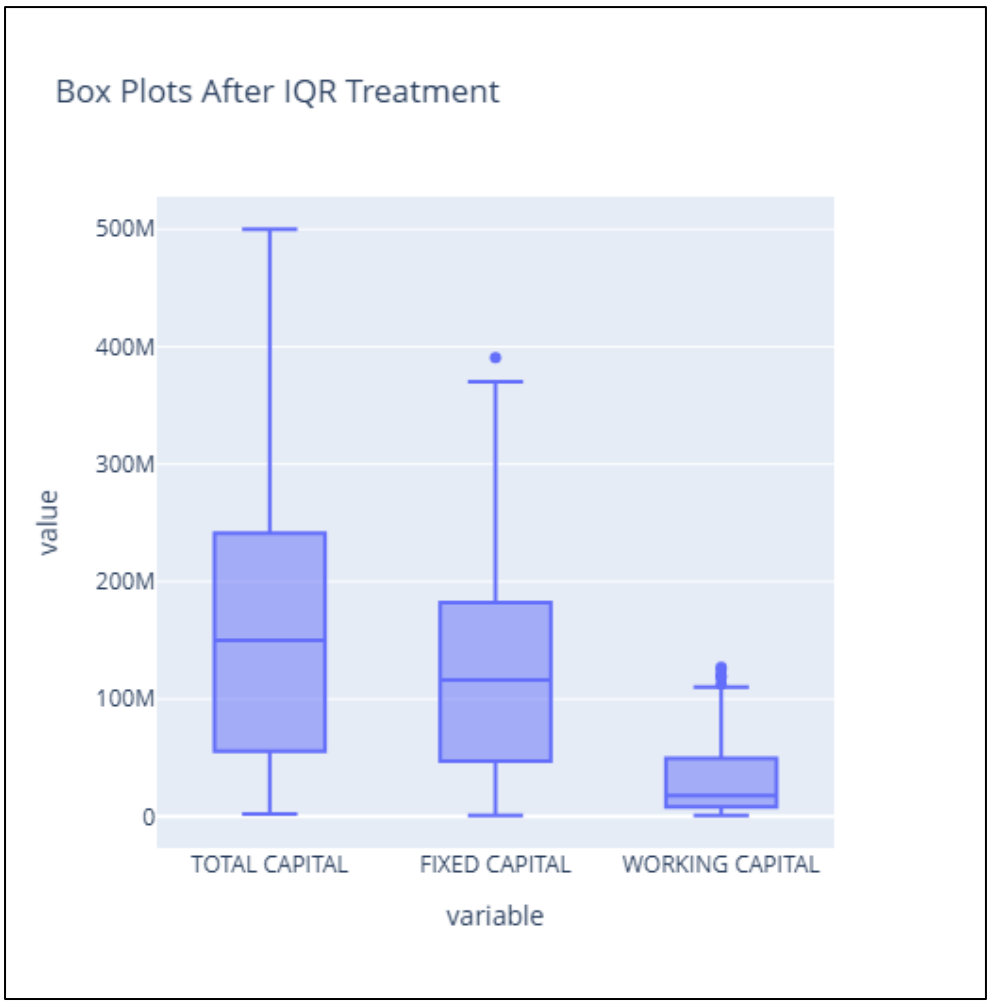


Figure 2. after iqr

before and after is shown the rows with outliers were removed for from dataset and used to train model more accurately

3.2. Model

3.2.1.Target and Features

```
# @title features and target
features = ['TOTAL CAPITAL', 'FIXED CAPITAL', 'WORKING CAPITAL', 'POWER', 'SCALE_encode',
            'PROVINCE_BAGMATI', 'PROVINCE_GANDAKI', 'PROVINCE_KOSHI', 'PROVINCE_LUMBINI','PROVINCE_MADHESH', 'PROVINCE_SUDUR-PASCHIM',
            'CATEGORY_FREQ','Local_Investment', 'Foreign_Investment']
target = 'CATEGORY'
```

Code 32. target and feature

columns were chosen for features and target

```
#@title define feature and target
X = df4[features]
y = df4[target]
```

Code 33. define X and y

feature is labeled as X and target is defined as y

3.2.2.Train Test Split

```
#@title import and train test splot
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Code 34. import and split dataset

train test split is used on the X and y variables as 20% of X and y is for testing, 80% of X and y is for training the model

3.2.3.Algorithms

```
# @title import models
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
```

Code 35. import models

logistic regression as linear model and random forest classifier as ensemble from scikit-learn were imported for classification task

```
#@title list of models to evaluate
models = [
    LogisticRegression(max_iter=1000, random_state=42),
    RandomForestClassifier(random_state=42),
]
```

Code 36. list of models

model is listed with parameters

3.2.4.Evaluation

```
#@title metrics function accuracy and classification
from sklearn.metrics import accuracy_score, classification_report
```

Code 37. import metrics of evaluations

accuracy score and classification report is imported for metrics evaluations

```
# @title function to train and evaluate a model
def evaluate_model(model, X_train, X_test, y_train, y_test):
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    report = classification_report(y_test, y_pred)
    print(f"Model: {model.__class__.__name__}")
    print(f"Accuracy: {accuracy:.4f}")
    print(f"Classification Report:\n{report}\n")
    return accuracy, report
```

Code 38. function to evaluate train print model

function to train and evaluate is written.

```
# @title evaluate each model
results = {}  # Store results for comparison
for model in models:
    accuracy, report = evaluate_model(model, X_train, X_test, y_train, y_test)
    results[model.__class__.__name__] = {'accuracy': accuracy, 'report': report}
```

Code 39. to display the results

results are stored and shown together for better comparison

```
Model: LogisticRegression
Accuracy: 0.6216
Classification Report:
                        precision    recall  f1-score   support

     AGRO AND FORESTRY       0.00      0.00      0.00         7
                ENERGY       0.00      0.00      0.00         1
INFORMATION TECHNOLOGY       0.00      0.00      0.00         2
        INFRASTRUCTURE       0.00      0.00      0.00         1
         MANUFACTURING       0.62      0.89      0.73        18
               SERVICE       0.70      0.41      0.52        17
               TOURISM       0.61      0.82      0.70        28

              accuracy                           0.62        74
             macro avg       0.27      0.30      0.28        74
          weighted avg       0.54      0.62      0.56        74
```

Code 40. evaluation of logistic regression

the accuracy calculated is 0.6216

```
Model: RandomForestClassifier
Accuracy: 0.9595
Classification Report:
                        precision    recall  f1-score   support

     AGRO AND FORESTRY       0.83      0.71      0.77         7
                ENERGY       0.50      1.00      0.67         1
INFORMATION TECHNOLOGY       1.00      1.00      1.00         2
        INFRASTRUCTURE       0.00      0.00      0.00         1
         MANUFACTURING       1.00      1.00      1.00        18
               SERVICE       1.00      1.00      1.00        17
               TOURISM       0.97      1.00      0.98        28

              accuracy                           0.96        74
             macro avg       0.76      0.82      0.77        74
          weighted avg       0.95      0.96      0.95        74
```

Code 41. evaluation of random forest classifier

the accuracy calculated is 0.9595

4. SECTION D

4.1. Reporting Model

4.1.1.Normal

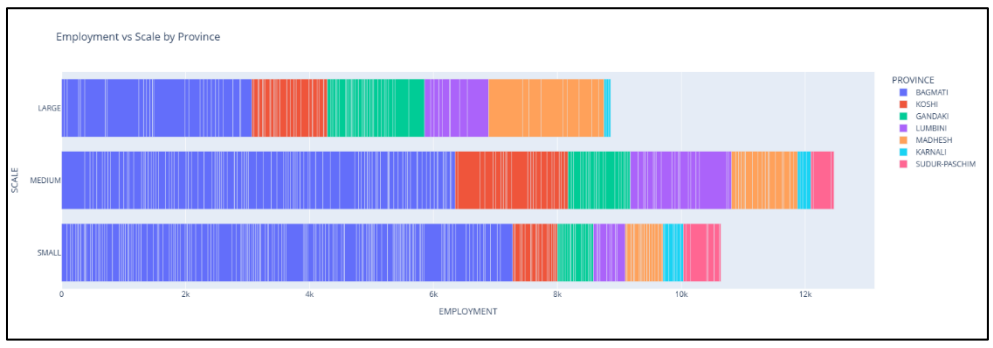


Figure 3. Employment in each Scale by Province.

this shows the stacked bar graph of distribution of employment across in all 3 level of scale and color coded based on province. the stakeholders can view important province and maybe equally distribute working opportunity across the nation.

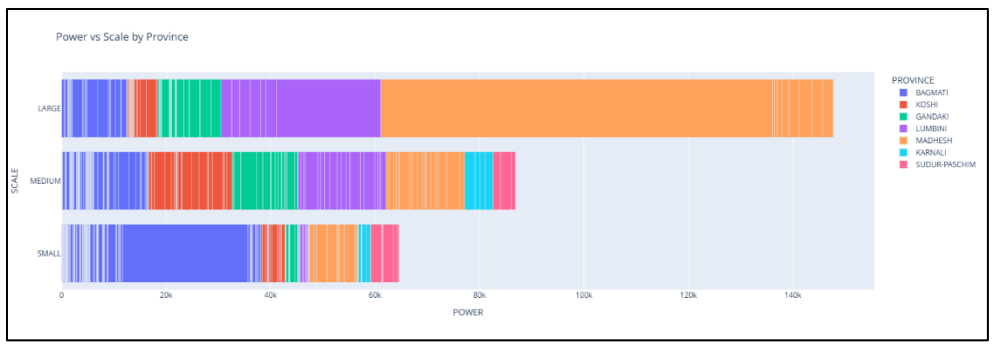


Figure 4. power vs scale by province

this stacked bar graph shows the power consumption in KVA unit across 3 scale of industry colored by province. the department of industry can recommend government to provide back up to huge consumption, maybe persuade to step up the production or introduced sustainable and efficient energy maybe nuclear energy.

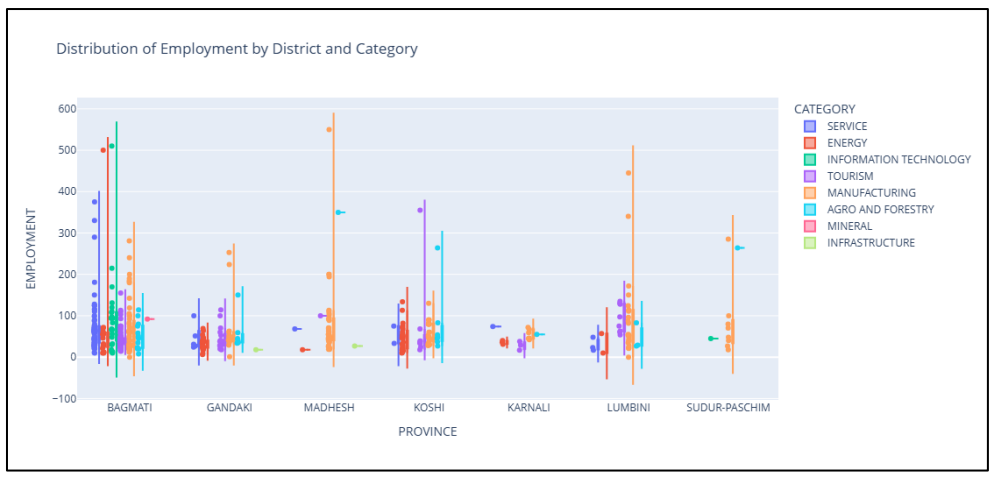


Figure 5. distribution of employment by district and category

the violin plot will show the distribution and concentrations like in box and density also the range of data. the concentration and width indicate frequency narrow is low, wide is high. it is shaped like violin body. the stakeholder can see the central tendency, quartiles, overlays like strip and swarm plots.

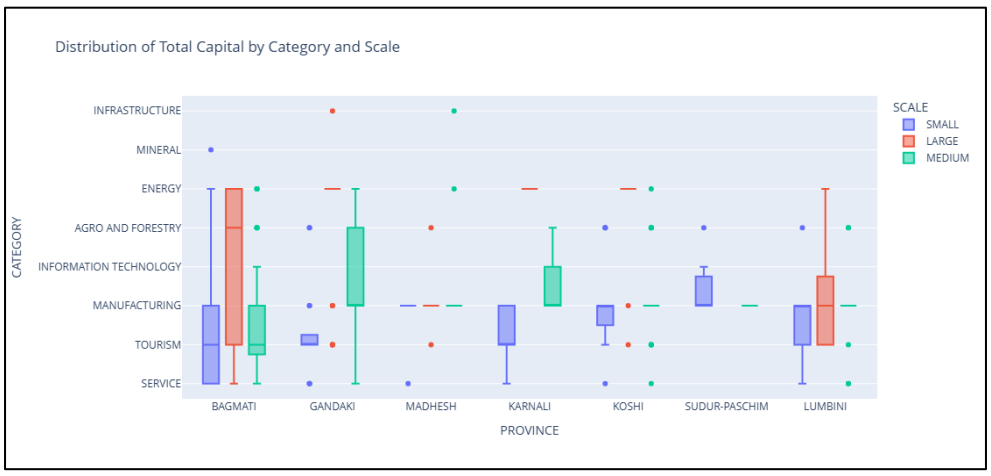


Figure 6. distribution of total capital by category and scale

multiple box plots will show the distribution of total capital by categories and colored scale. allowing department of industry to see outliers and concentrations. maybe it will help company to help grow and find a sweet spot.

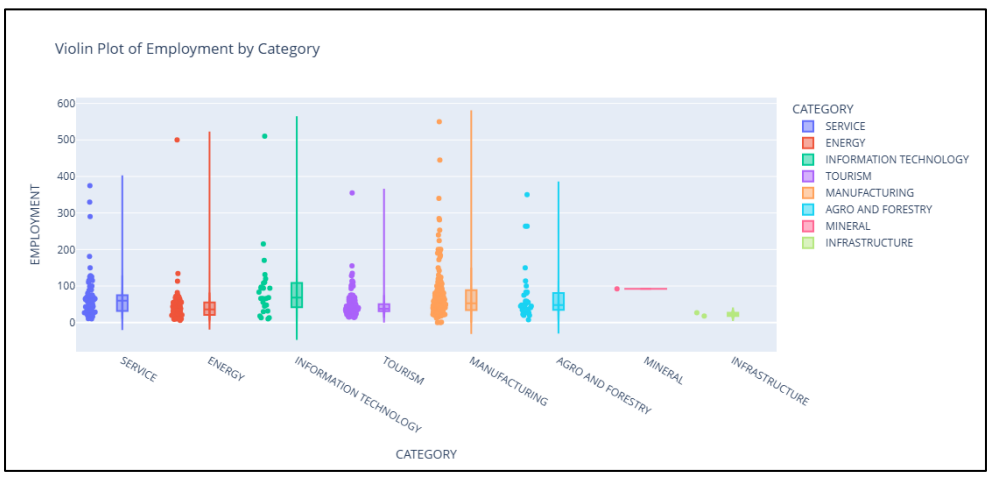


Figure 7. violin plot of employment by category

this like before will show the spread and central tendency of employment by category

4.1.2.grouping

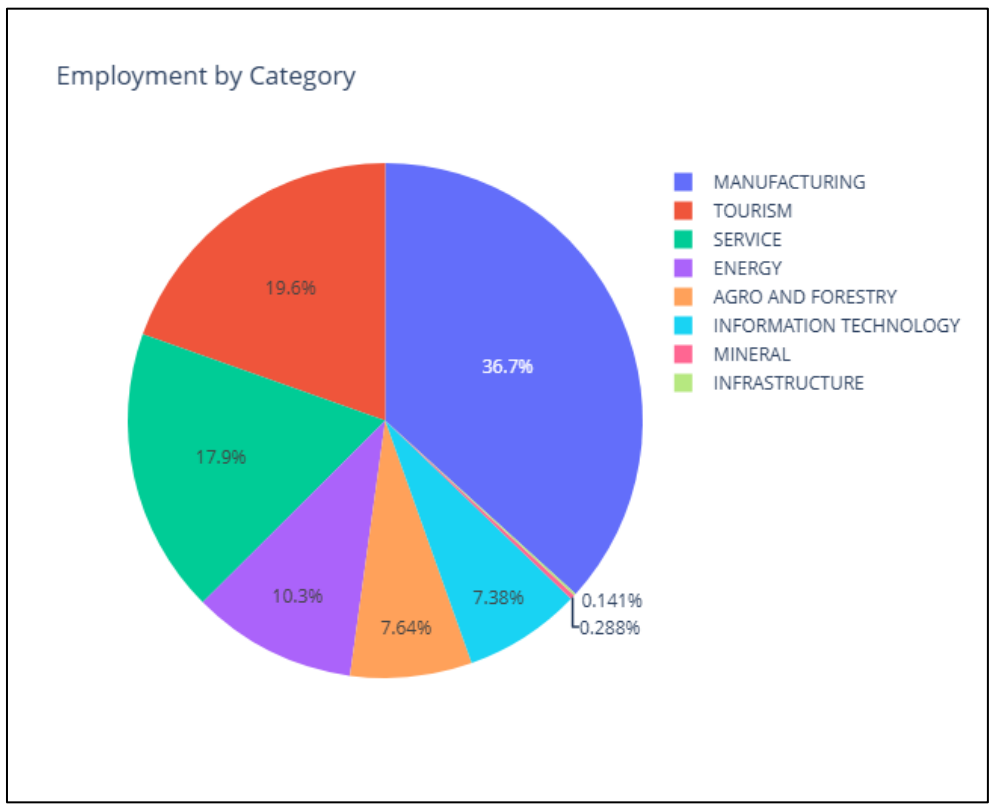


Figure 8. employment by category

this pizza chart shows employment by category

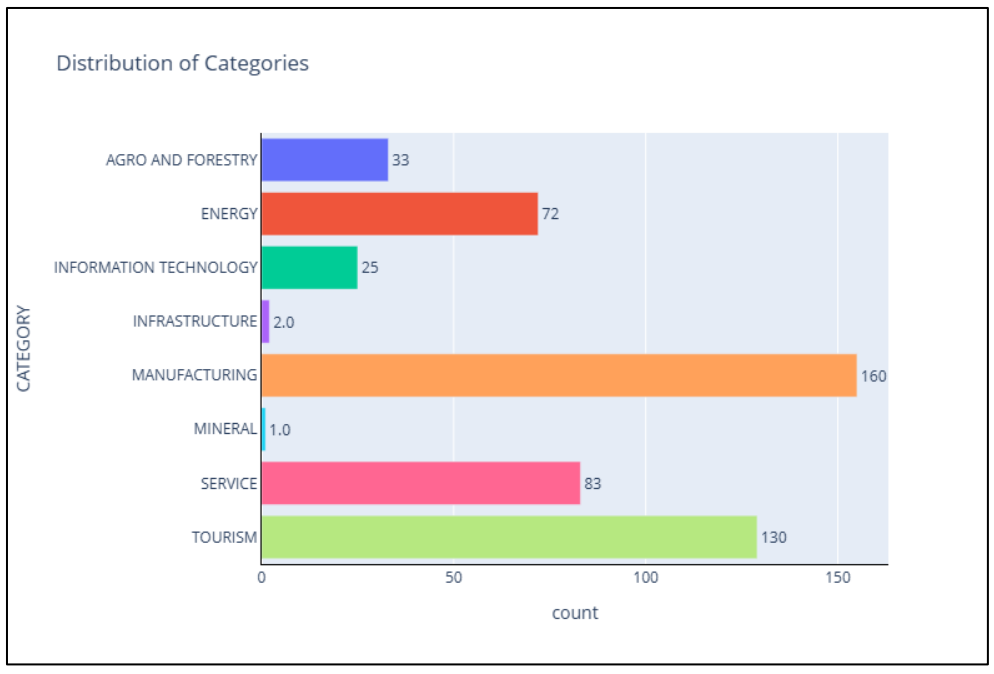


Figure 9. distribution of categories by count

this horizontal bar graph shows the distribution of categories by its count, individually color coded. the industry minister can maybe equally distribute priorities based on natural resources found in Nepal.

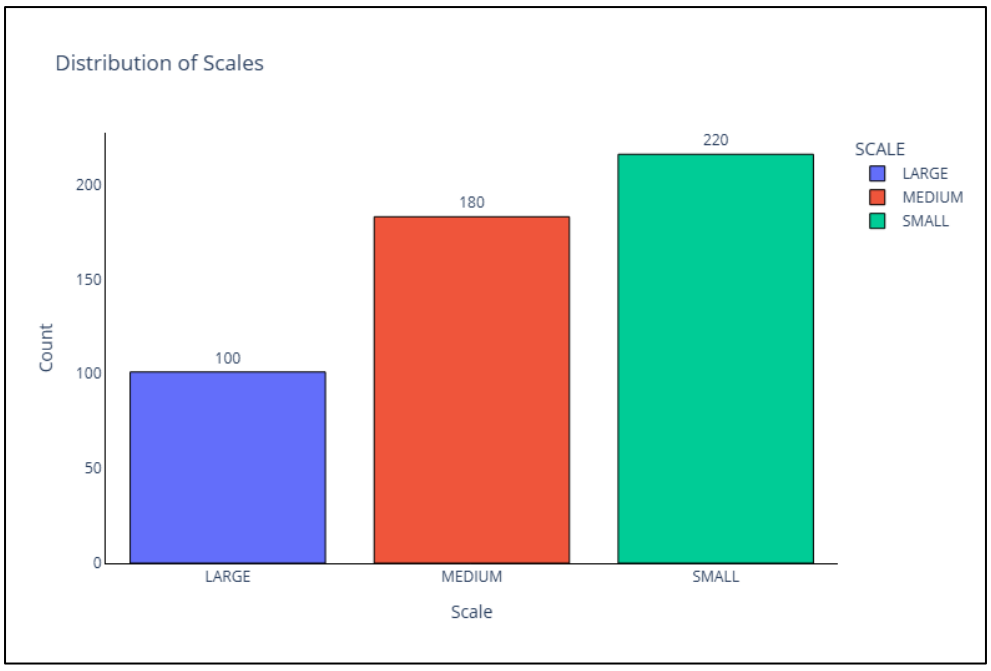


Figure 10. distribution of scale by count

this bar chart shows the distribution of scale by count. it is seen that small scale industry is still dominant in the country. the industry should help each other so all grow together.

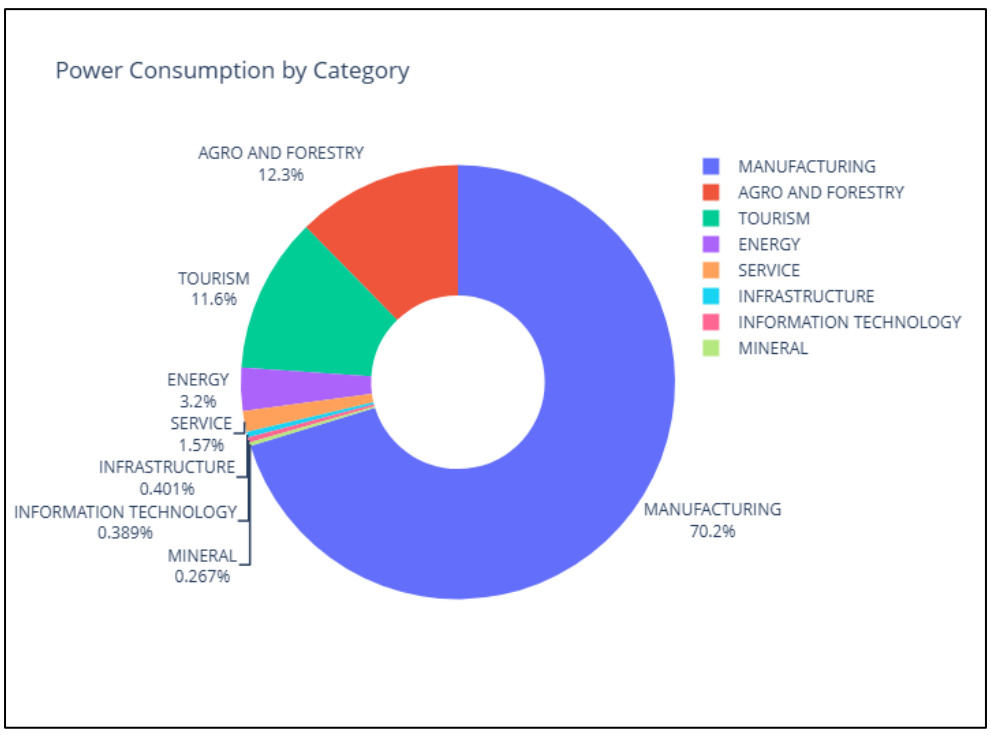


Figure 11. donut chart power consumptions

this donut chart shows the distributions of power consumption in percentage. this may help the stakeholder to better grasp the consumption by category

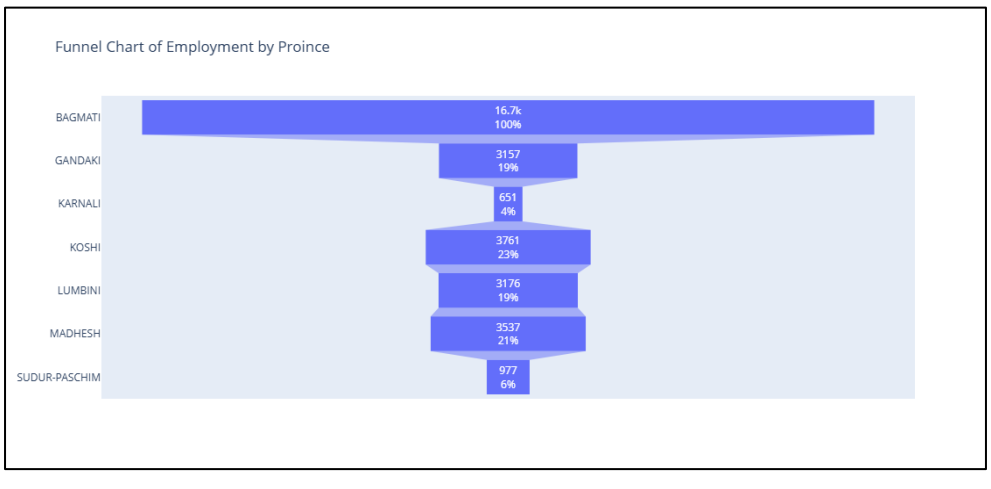


Figure 12. funnel chart of employment by province

to summarize large amount of data funnel chart can be used. it is important business metrics and can find bottle necks. the stakeholders can peek at the chart for comparative of size in employment and drop off in job retention.

4.1.3.Slightly Advanced

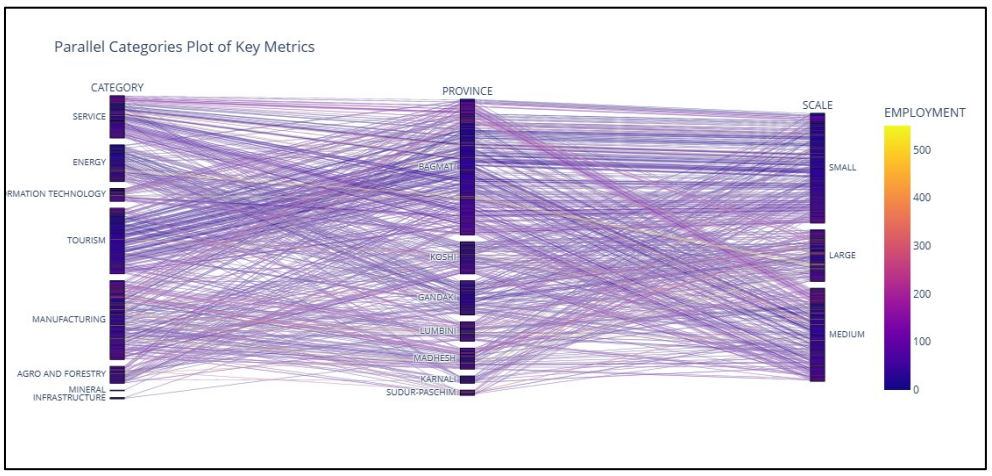


Figure 13. parallel categories of key metrics

the parallel categories plot will help to visualize the categories on the axis and showing pathways to indicate possible relationships and combinations,

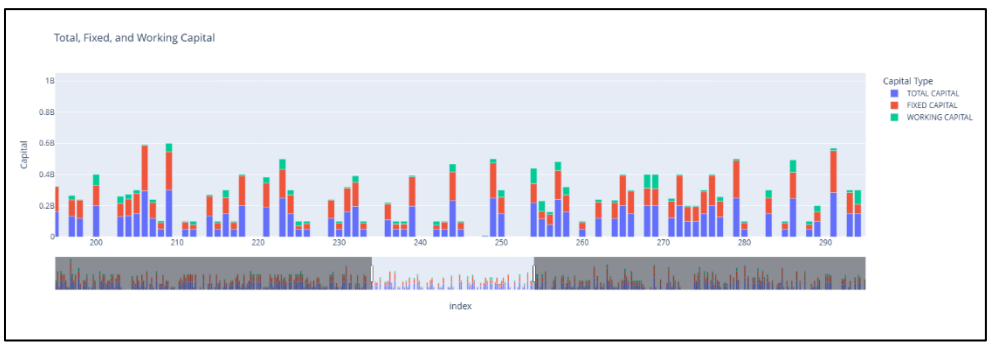


Figure 14. total, fixed, working capital

this stacked bar graph with sliding features can help visualize the how different brands and companies' capital is distributed in range.

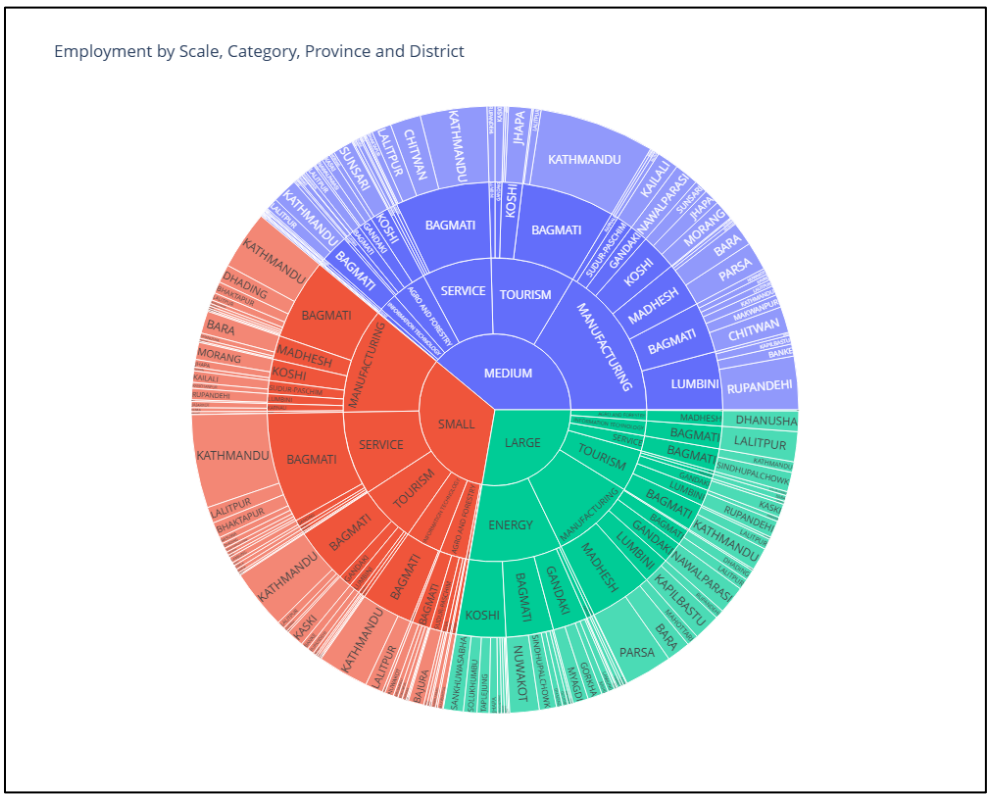


Figure 15. sun burst chart

employment distribution by 3 scales, 8 categories, 7 province and 45 districts.

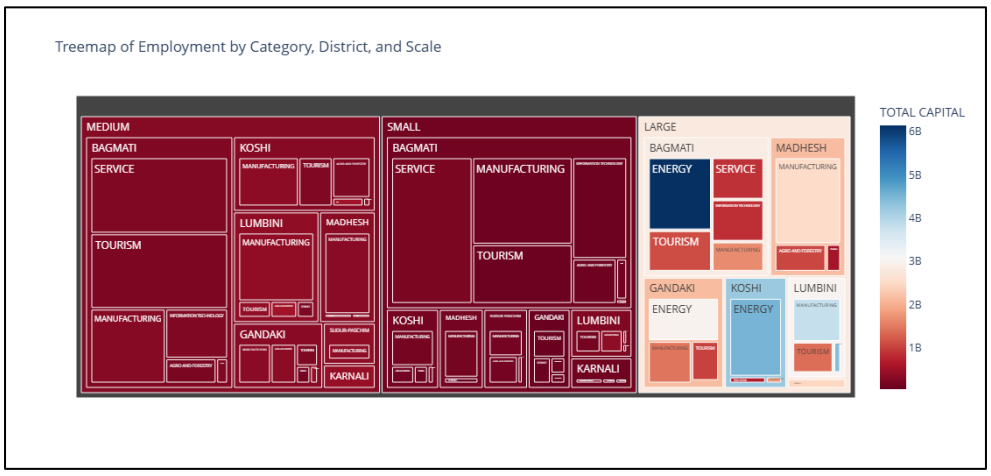


Figure 16. tree map of employment by category, district and scale

this tree map visual will help to display hierarchical data, that cannot be shown with bar graph and proportions taken by each category is shown that can be color coded

4.1.4.Aggregation for Report

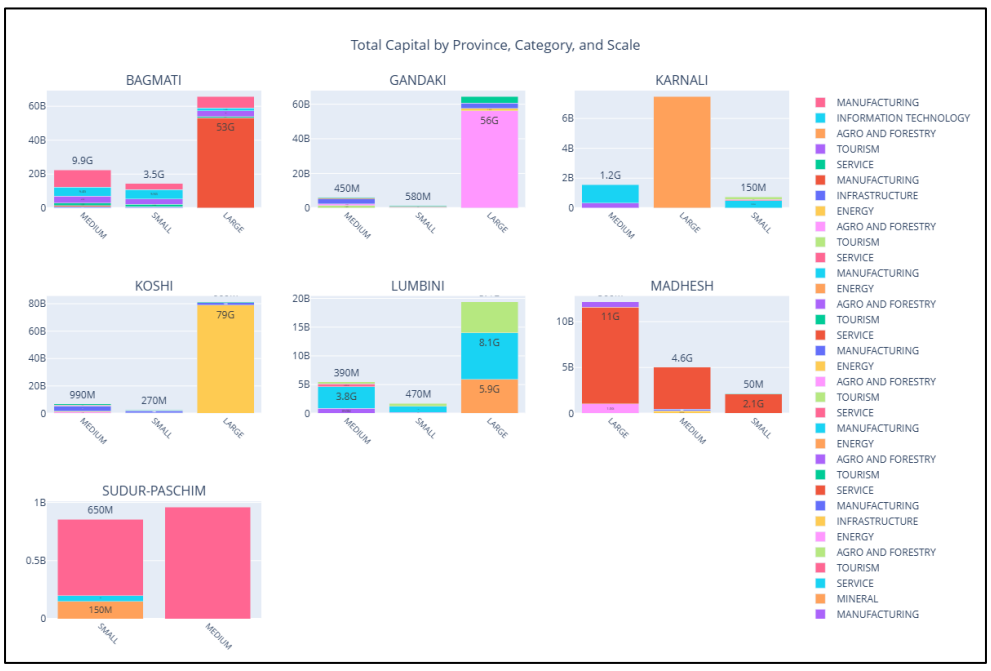


Figure 17. total capital by province category and scale

showing the distribution of scales total capacity based on provinces and colored by the categories

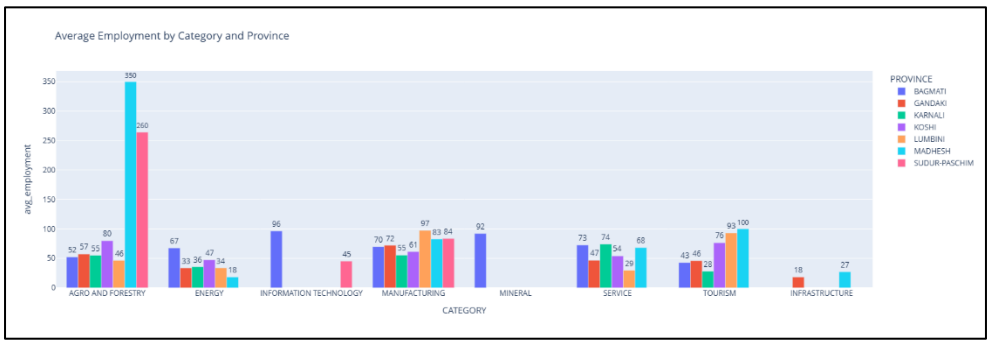


Figure 18. average employment by category and province

this bar charts stacked together shows the mean of distribution of employment of each category and colored by province

```
# @title sepearte dataframes based on different provi

provinces = df0['PROVINCE'].unique()  # Assuming you
categories = df0['CATEGORY'].unique()
scales = df0['SCALE'].unique()

separated_dfs = {}

for province in provinces:
  for category in categories:
    for scale in scales:
      key = f"{province}_{category}_{scale}"
      separated_dfs[key] = df0[
          (df0['PROVINCE'] == province) &
          (df0['CATEGORY'] == category) &
          (df0['SCALE'] == scale)
      ].copy()

for key, df in separated_dfs.items():
    if not df.empty:
        print(f"\nDataFrame for {key}:")
        display(df.head())
    else:
        print(f"\nDataFrame for {key}: (Empty)")
```

Code 42. function to combine unique values of data frame and aggregate them

creating separate data frames based on different province, categories and scale then display unique

DataFrame for KOSHI_TOURISM_SMALL:

	INDUSTRY NAME	DISTRICT	TOTAL CAPITAL	FIXED CAPITAL	WORKING CAPITAL	PRODUCT AND ANNUAL CAPACITY	EMPLOYMENT	CATEGORY	SCALE	POWER	% OF INVESTMENT	MONTH	YEAR
156	CELESTIALRESORTS	ILAM	100000000	95000000	5000000	HOTEL30BEDSRESTAURANT50SEATS	35	TOURISM	SMALL	70	Local - 0%, Foreign - 100%	9	2076
238	FATTSERNVEGRESTAURANT	SUNSARI	50000000	33000000	17000000	RESTAURANT40SEATS	18	TOURISM	SMALL	50	Local - 0%, Foreign - 100%	3	2077
271	PUMORIJOURNEYS	SOLUKHUMBU	123200000	105200000	18000000	HOTEL20BEDSRESTAURANT100SEATS	24	TOURISM	SMALL	100	Local - 100%, Foreign - 0%	4	2077

DataFrame for KOSHI_TOURISM_LARGE:

	INDUSTRY NAME	DISTRICT	TOTAL CAPITAL	FIXED CAPITAL	WORKING CAPITAL	PRODUCT AND ANNUAL CAPACITY	EMPLOYMENT	CATEGORY	SCALE	POWER	% OF INVESTMENT	MONTH	YEAR	PROVIN
453	MOUNTEVERESTCABLECAR	SOLUKHUMBU	660000000	625000000	35000000	CABLECAR1120000PERSONS	68	TOURISM	LARGE	2000	Local - 100%, Foreign - 0%	3	2078	KO

DataFrame for KOSHI_TOURISM_MEDIUM:

	INDUSTRY NAME	DISTRICT	TOTAL CAPITAL	FIXED CAPITAL	WORKING CAPITAL	PRODUCT AND ANNUAL CAPACITY	EMPLOYMENT	CATEGORY	SCALE	POWER	% OF INVESTMENT	MONTH	YEAR	PR
126	VEGASRECREATIONNEPAL	JHAPA	250000000	192000000	58000000	CASINOPLAYERS200000PERSONS	355	TOURISM	MEDIUM	150	Local - 100%, Foreign - 0%	8	2076	
191	NEPALIRIKAHOTEL	MORANG	222925000	207900000	15025000	HOTEL48BEDSRESTAURANT100SEATS	39	TOURISM	MEDIUM	100	Local - 100%, Foreign - 0%	11	2076	
373	NEPALIRIKAHOTEL(1)	JHAPA	171200000	159200000	12000000	HOTEL48BEDRESTAURANT80SEAT	38	TOURISM	MEDIUM	100	Local - 100%, Foreign - 0%	11	2077	
448	HOTELYARSHA	SUNSARI	350766500	337000000	13766500	HOTEL42BEDRESTAURANT60SEAT	34	TOURISM	MEDIUM	250	Local - 100%, Foreign - 0%	3	2078	

Images 9. data frame of aggregates

```
# @title Employment Generation per District:
district_summary = df0.groupby(['DISTRICT']).agg({'EMPLOYMENT': 'sum'})
district_summary = district_summary.sort_values(by='EMPLOYMENT', ascending=False)


display(district_summary)

fig = px.bar(district_summary, x=district_summary.index, y='EMPLOYMENT',
             title='Employment by District')
fig.show()
```

Code 43. district summary code to see mean of employment sum

DISTRICT	EMPLOYMENT
KATHMANDU	8800
LALITPUR	2686
RUPANDEHI	1827
PARSA	1296
BARA	1287
CHITWAN	1246
NAWALPARASI	1157
KASKI	1080
JHAPA	1058

Images 10. district and employment data frame

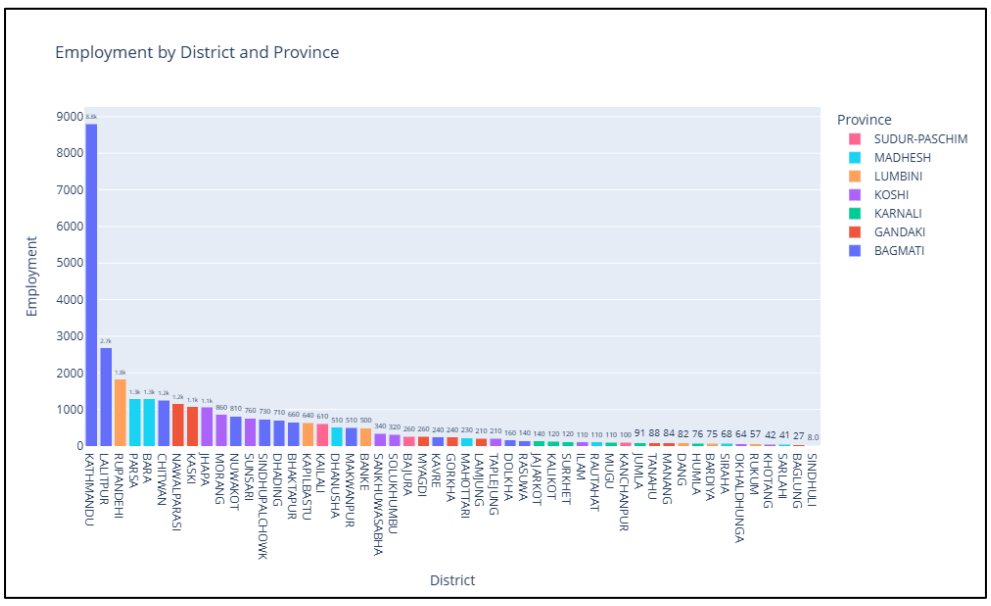


Figure 19. employment by district and province

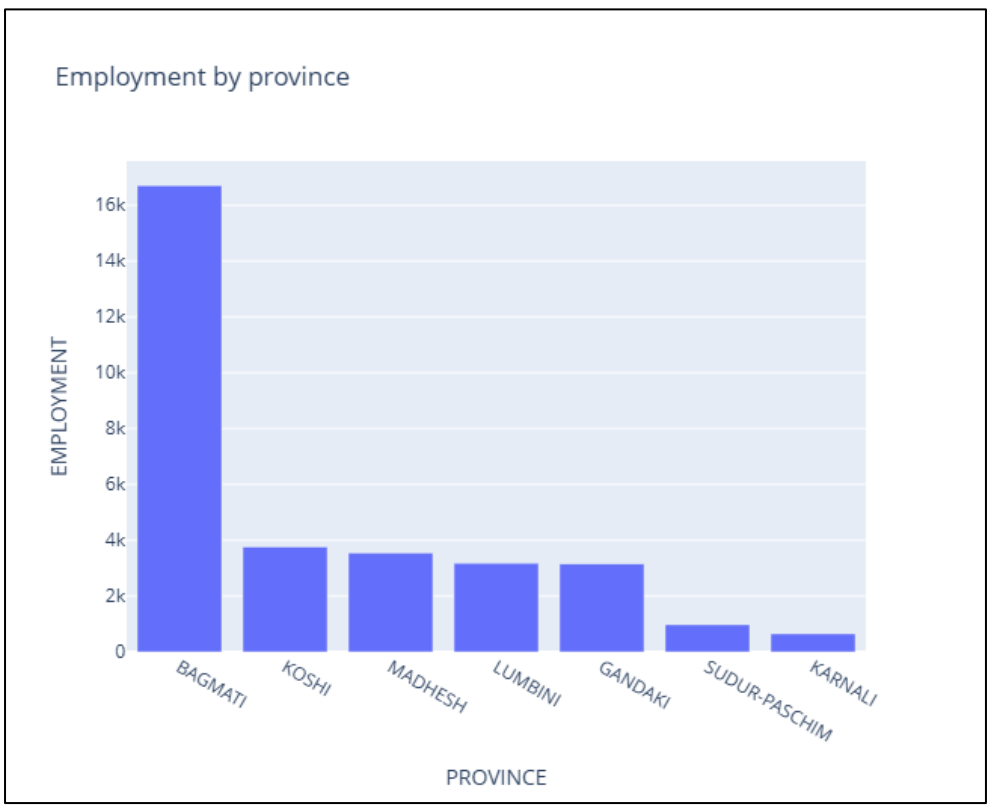


Figure 20. employment and province.

these graphs show the distribution that the stakeholder may look and really ponder and make sound decisions.

4.2. Future Enhancements

- Dash from Plotly can be used as interactive GUI and highly customizable with python being most popular programming language,
- maybe a live display of data on performance indicator investment growth and trend can be shown
- Interactive buttons with Key metrics and KPI quick assessment for stakeholders
- metrics like industry growth, capital distribution, employment trends real time update
- monthly and annual reports can be documented with automation technologies, where summarization is possible with performance and accuracy
- for non-technical users' in-depth visual analytics can help in exploration
- this dataset only has 500 rows and 13 columns in future the system must be handle large data, easy integration, scalable, effective report,
- user growth indicators like directly in excel sheet from instead of pdf
- implementation of mobile applications

5. Appendix

5.1. References

Home. (n.d.). Nepaleconomicforum.org. https://nepaleconomicforum.org/

IBISWorld, I. (2019). *IBISWorld - Industry Market Research, Reports, and Statistics*. Ibisworld.com. https://www.ibisworld.com/industry-trends/

IFC. (2023). *International Finance Corporation (IFC)*. IFC. https://www.ifc.org/en/home

Kalpana Khanal. (2023, December 4). *Start-Up Businesses and Micro, Small and Medium Enterprises in Nepal: A Policy Perspective Kathmandu, Nepal*. https://www.researchgate.net/publication/376190585_Start-Up_Businesses_and_Micro_Small_and_Medium_Enterprises_in_Nepal_A_Policy_Perspective_Kathmandu_Nepal

kodiary.com, & kodiary.com. (2023). *CNI NEPAL - Confederation of Nepalese Industries*. Cni.org.np. https://cni.org.np/

Madgavkar, A., Piccitto, M., White, O., Ramírez, M. J., Mischke, J., & Chockalingam, K. (2024, May 2). *Opportunities for Small Businesses to Boost Productivity | McKinsey*. Www.mckinsey.com. https://www.mckinsey.com/mgi/our-research/a-microscope-on-small-businesses-spotting-opportunities-to-boost-productivity

Marketresearch. (2019). *MarketResearch.com: Market Research Reports and Industry Analysis*. Marketresearch.com. https://www.marketresearch.com/

Nepal Rastra Bank - Central Bank of Nepal. (2012). Nrb.org.np. https://www.nrb.org.np/

Open Knowledge World Bank. (n.d.). Openknowledge.worldbank.org. https://openknowledge.worldbank.org/home

Pandas. (2024). *pandas documentation — pandas 1.0.1 documentation*. Pandas.pydata.org. https://pandas.pydata.org/docs/

Plotly. (2023). *Plotly Python Graphing Library*. Plotly.com. https://plotly.com/python/

scikit-learn: machine learning in Python — scikit-learn 1.6.dev0 documentation. (2024). Scikit-Learn.org. https://scikit-learn.org/dev/index.html

Statista. (2024). *The Statistics Portal for Market data, Market Research and Market Studies*. Statista.com; Statista. https://www.statista.com/

Supporting small and medium industry clusters | UNIDO. (2024). UNIDO. https://www.unido.org/our-focus-advancing-economic-competitiveness/supporting-small-and-medium-industry-clusters

The World Bank. (2023). *Nepal Development Update*. World Bank. https://www.worldbank.org/en/country/nepal/publication/nepaldevelopmentupdate

उधोग विभाग, त्रिपुरेश्वर, काठमाडौँ. (n.d.). उधोग विभाग, त्रिपुरेश्वर, काठमाडौँ. https://www.doind.gov.np/

United Nations. (n.d.). *Micro-, Small and Medium-sized Enterprises (MSMEs) | Department of Economic and Social Affairs*. Sdgs.un.org. https://sdgs.un.org/topics/capacity-development/msmes

World Bank. (2019, October 16). *Small and Medium Enterprises (SMEs) Finance*. World Bank; www.worldbank.org. https://www.worldbank.org/en/topic/smefinance

(2024). Moics.gov.np. https://moics.gov.np/

5.2. List of Abbverations

AI	Artificial Intelligences	**IMIS**	Industry Management Information System
ML	Machine Learnings	**DM**	Data Modelling
DB	Data Base	**PVT LTD**	Private Limited
NP	NumPy	**KVA**	Kilo Volt Ampere
OSS	Open-Source Software	**LE**	Label Encoding
PD	Pandas	**OHE**	One Hot Encoding
PY	Python	**freq**	Frequency
SK	Scikit-learn	**X, y**	Feature, Target
RF	Random Forest	**S No**	Serial Number
KPI	Key Performance Index	**P**	Probability
IT	Information Technology	**B**	Coefficient (weight)
CSV	Comma Separated Value	**logit**	Natural Logarithm
IQR	Inter Quartile Range	**MLE**	Maximum Likelihood Estimation
RE	Regular Expression	**OR**	Odd Ratio
NSO	National Statistic Office	**D**	Entropy
NRB	Nepal Rastra Bank	**OOB**	Out of Bag
CBS	Central Bureau of Statistics	**F**	Feature
DS	Data Science	**T**	Tree
NDP	National Data Portal	**DF**	Data Frame
MSE	Micro Small Enterprise	**IPYNB**	Interactive Python Notebook
T/F	True False	**AD**	Anno Domini
P/N	Positive Negative	**MLE**	Medium-Large Enterprise
CBN	Central Bank Nepal	**TO**	Total Observation
FPR	False Positive Rate	**BS**	Bikram Sambat
ROC	Receiver Operating Characteristics	**AUC**	Area Under Curve