Utkarsh Tripathi

[ ]:

[ ]:

Credit Risk Modelling

- *Credit risk is an important topic in the field of finance because banks and other financial institutions heavily invest in reducing their credit risk.*
- *There is an element of risk when these financial institutions lend money to clients. So ideally,to reduce their risk and loan money only to clients they are sure are going to pay them back.*

Project goal: The objective of this project is to build a model to predict probability of a client defaulting a loan.

- Defaulting on a loan is the failure of a borrower to pay the principal or interest on a security or loan.
- *The dataset has been collected from the UCI machine learning repository*
- **The following steps will be followed in building the model.**
- Data preparation and Pre-processing
- Exploratory Data Analysis
- Feature Engineering and Selection
- Model Development and Model Evaluation
- Hyperparameter Tuning

[2]:
```
#importing necessary packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from matplotlib import rcParams
rcParams['figure.figsize']=(12,5)
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
sns.set_style('whitegrid')
pd.set_option('display.max_columns',None)

from collections import Counter
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import f1_score , confusion_matrix , classification_report,roc_auc_score , roc_curve
from sklearn.model_selection import StratifiedKFold , cross_val_score
from sklearn.model_selection import GridSearchCV
```

[3]:
```
#importing the data
data = pd.read_excel('Credit_Risk_PD_Model_Dataset.xlsx')
data.head()
```

[3]:

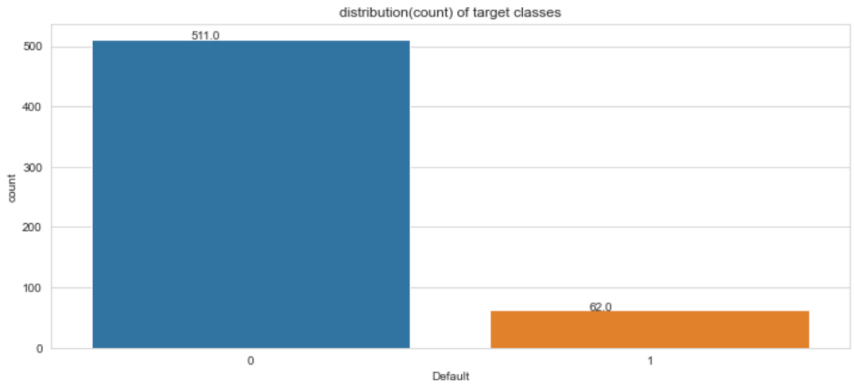
	Industry	CASH_AND_CASH_EQUIVALENTS_floored1	COST_OF_GOODS_SOLD_1	CURRENT_ASSETS_1	CURRENT_LIABILITIES_1	DEPRECIATION_1	EBITDA_1	FIXED
0	Wholesale	323773.0	NaN	476381.0	516192.0	78247.0	357842.0	
1	EduArtsHealthSocial	18593.0	11753.0	21913.0	257405.0	6988.0	42274.0	
2	Retail	202720.0	3420276.0	665190.0	373352.0	76899.0	406223.0	
3	Agriculture - Other	87515.0	7905311.0	1820195.0	2153519.0	702651.0	675750.0	
4	EduArtsHealthSocial	272938.0	237933.0	381958.0	681622.0	NaN	-515748.0	

[4]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 573 entries, 0 to 572
Data columns (total 39 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   Industry                            573 non-null    object
 1   CASH_AND_CASH_EQUIVALENTS_floored1  573 non-null    float64
 2   COST_OF_GOODS_SOLD_1                518 non-null    float64
 3   CURRENT_ASSETS_1                    573 non-null    float64
 4   CURRENT_LIABILITIES_1               573 non-null    float64
 5   DEPRECIATION_1                      534 non-null    float64
```

```
 6   EBITDA_1                            573 non-null    float64
 7   FIXED_AND_NON_CURRENT_ASSETS_1      561 non-null    float64
 8   GROSS_PROFIT_1                      573 non-null    float64
 9   INTEREST_EXPENSE_1                  480 non-null    float64
 10  LONG_TERM_DEBT_1                    359 non-null    float64
 11  NET_FIXED_ASSETS_1                  555 non-null    float64
 12  NET_INCOME_1                        573 non-null    float64
 13  NET_INTANGIBLES_1                   573 non-null    float64
 14  NET_TRADE_RECV_1                    475 non-null    float64
 15  NON_CURRENT_LIABILITIES_1           449 non-null    float64
 16  RETAINED_EARNINGS_1                 573 non-null    float64
 17  TOTAL_ASSETS_1                      573 non-null    float64
 18  TOTAL_DEBT_1                        527 non-null    float64
 19  TOTAL_INVENTORY_1                   481 non-null    float64
 20  TURNOVER_1                          573 non-null    float64
 21  WORKING_CAPITAL_1                   573 non-null    float64
 22  Debt_to_TNW_1                       527 non-null    float64
 23  Equity_to_Liability_1               573 non-null    float64
 24  Equity_to_Assets_1                  573 non-null    float64
 25  Long_Term_Debt_to_Equity_1          359 non-null    float64
 26  NetFixedAssets_to_TotalDebt_1       515 non-null    float64
 27  Total_Liab_to_Total_Assets_1        573 non-null    float64
 28  EBITDA_to_total_Debt_1              527 non-null    float64
 29  EBITDA_to_Interest_1                480 non-null    float64
 30  EBITDA_to_curr_liab_1               573 non-null    float64
 31  Cash_to_Total_assets_1              573 non-null    float64
 32  Current_Ratio_1                     573 non-null    float64
 33  Liquidity_Ratio_1                   475 non-null    float64
 34  Curr_to_Total_Assets_1              573 non-null    float64
 35  Gross_Margin_perc_1                 518 non-null    float64
 36  Net_Income_Margin_1                 573 non-null    float64
 37  Total_Liab_to_Total_Assets_2_1      562 non-null    float64
 38  Default                             573 non-null    float64
dtypes: float64(38), object(1)
memory usage: 174.7+ KB
```

[5]:
```
#checking for the distribution(count) of target classes
data['Default'] = data['Default'].astype('int')
ax = sns.countplot(data['Default'])
plt.title('distribution(count) of target classes')
for p in ax.patches:
     ax.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.25, p.get_height()+0.01))
```



[6]:
```
#checking for missing values in the data
round(data.isna().sum()/len(data)*100,2).sort_values(ascending=False)
```

[6]:
```
Long_Term_Debt_to_Equity_1          37.35
LONG_TERM_DEBT_1                    37.35
NON_CURRENT_LIABILITIES_1           21.64
Liquidity_Ratio_1                   17.10
NET_TRADE_RECV_1                    17.10
INTEREST_EXPENSE_1                  16.23
EBITDA_to_Interest_1                16.23
TOTAL_INVENTORY_1                   16.06
NetFixedAssets_to_TotalDebt_1       10.12
Gross_Margin_perc_1                  9.60
COST_OF_GOODS_SOLD_1                 9.60
TOTAL_DEBT_1                         8.03
EBITDA_to_total_Debt_1               8.03
Debt_to_TNW_1                        8.03
DEPRECIATION_1                       6.81
NET_FIXED_ASSETS_1                   3.14
FIXED_AND_NON_CURRENT_ASSETS_1       2.09
Total_Liab_to_Total_Assets_2_1       1.92
Curr_to_Total_Assets_1               0.00
Current_Ratio_1                      0.00
Net_Income_Margin_1                  0.00
Cash_to_Total_assets_1               0.00
EBITDA_to_curr_liab_1                0.00
Total_Liab_to_Total_Assets_1         0.00
Industry                             0.00
Equity_to_Assets_1                   0.00
Equity_to_Liability_1                0.00
WORKING_CAPITAL_1                    0.00
TURNOVER_1                           0.00
CASH_AND_CASH_EQUIVALENTS_floored1   0.00
TOTAL_ASSETS_1                       0.00
RETAINED_EARNINGS_1                  0.00
```

```
NET_INTANGIBLES_1                0.00
NET_INCOME_1                     0.00
GROSS_PROFIT_1                   0.00
EBITDA_1                         0.00
CURRENT_LIABILITIES_1            0.00
CURRENT_ASSETS_1                 0.00
Default                          0.00
dtype: float64
```

- considering a threshold of 25% for the missing values and,
- dropping the features which are having missing values more than the threshold

Removing features with more than 25% missing values

[7]:
```
#dropping missing values
data.dropna(axis=1,thresh=round(len(data)-len(data)/100*25),inplace=True)
```

[8]:
```
#checking for the distribution of independent variables
data.describe().apply(lambda s : s.apply('{0:.5f}'.format))
```

[8]:

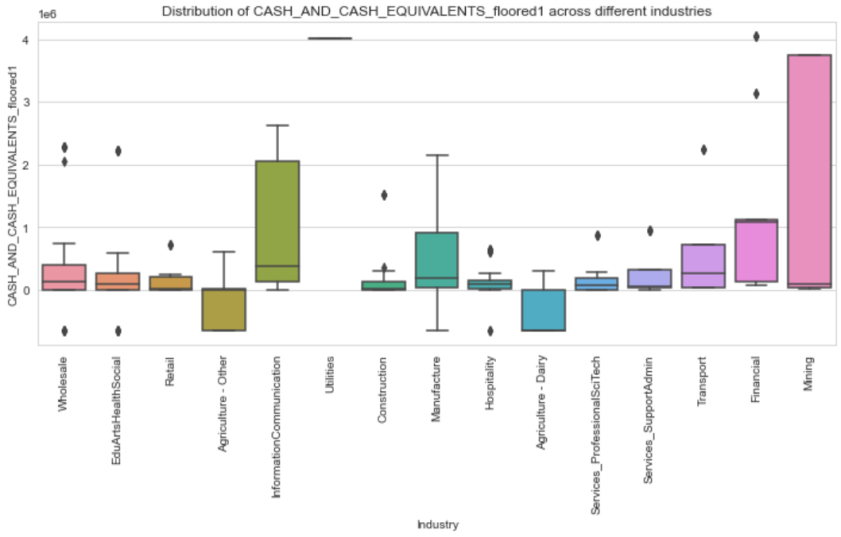
	CASH_AND_CASH_EQUIVALENTS_floored1	COST_OF_GOODS_SOLD_1	CURRENT_ASSETS_1	CURRENT_LIABILITIES_1	DEPRECIATION_1	EBITDA_1	FIXED_AND_NON
count	573.00000	518.00000	573.00000	573.00000	534.00000	573.00000	
mean	260737.46422	2418543.23359	1882030.51832	1519872.27051	95440.10300	354311.38220	
std	821575.22736	2862968.43545	3882481.64782	3559609.21764	155056.63976	852420.84859	
min	-644364.00000	5116.00000	1936.00000	228.00000	271.00000	-7324453.00000	
25%	790.00000	315268.00000	236199.00000	253171.00000	14941.00000	73034.00000	
50%	49605.00000	1024016.00000	770200.00000	706407.00000	45219.00000	189612.00000	
75%	284872.00000	4245791.75000	2180921.00000	1656906.00000	111614.25000	489874.00000	
max	4052236.00000	12212492.00000	34522271.00000	32716574.00000	1334359.00000	7660463.00000	

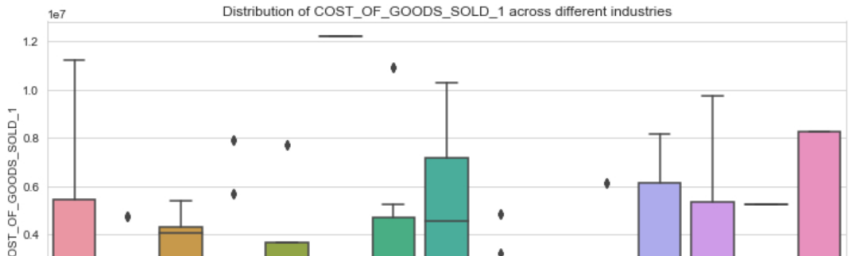
Checking for the distribution of the features with respect to the different industries

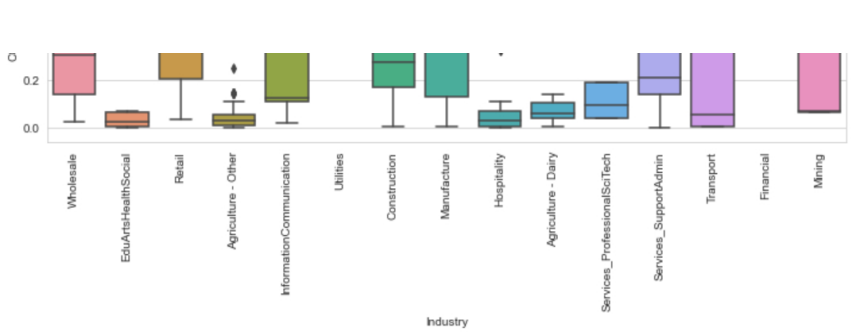
[9]:
```
data['Industry'].unique()
```

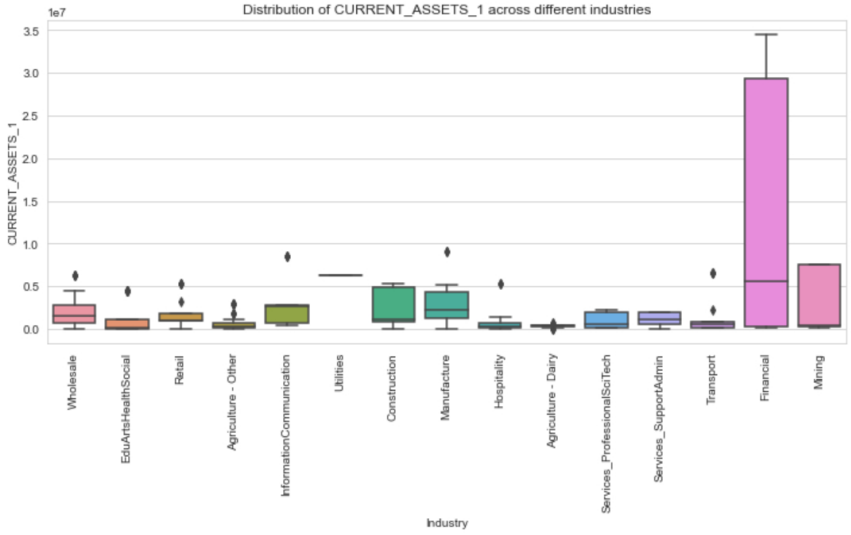
[9]:
```
array(['Wholesale', 'EduArtsHealthSocial', 'Retail',
       'Agriculture - Other', 'InformationCommunication', 'Utilities',
       'Construction', 'Manufacture', 'Hospitality',
       'Agriculture - Dairy', 'Services_ProfessionalSciTech',
       'Services_SupportAdmin', 'Transport', 'Financial', 'Mining'],
      dtype=object)
```

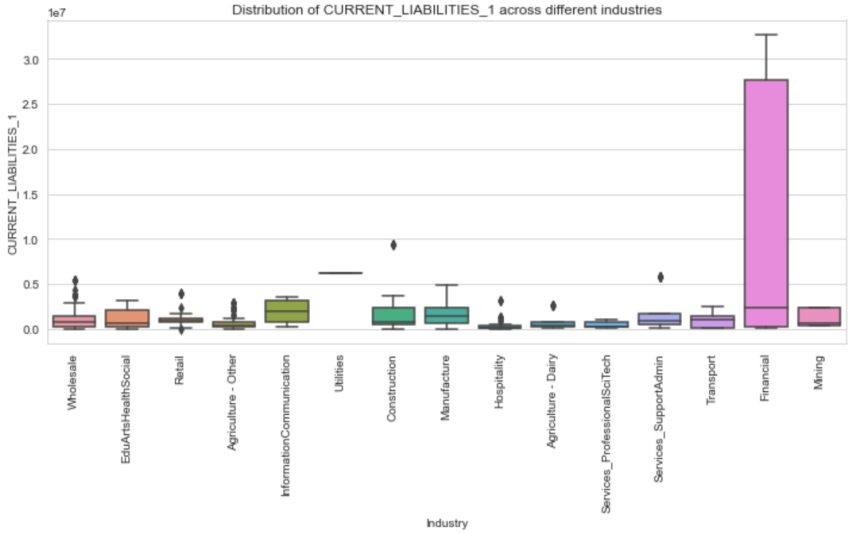
[10]:
```
numerical_features = data.select_dtypes(['float64']).columns
for feature in numerical_features:
    plt.figure(figsize=(12,5))
    sns.boxplot(y=data[feature],x=data['Industry'])
    plt.xticks(rotation=90)
    plt.title(f'Distribution of {feature} across different industries')
```

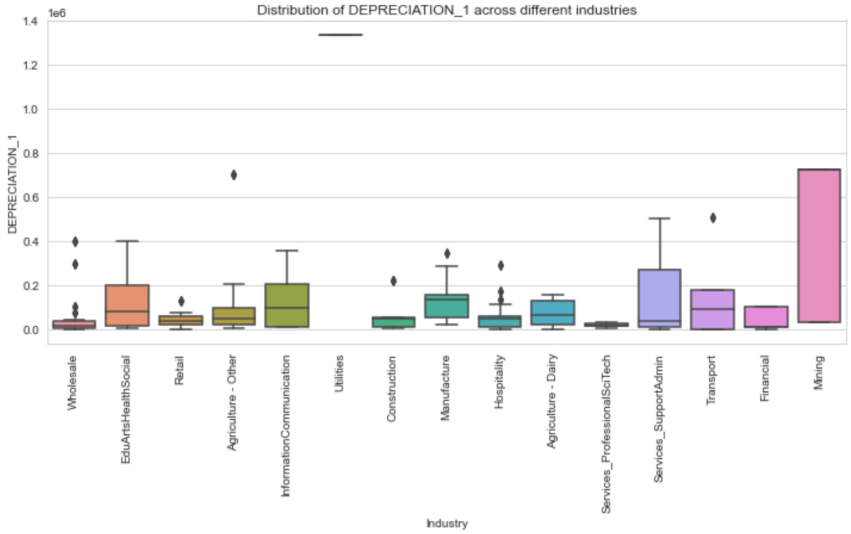




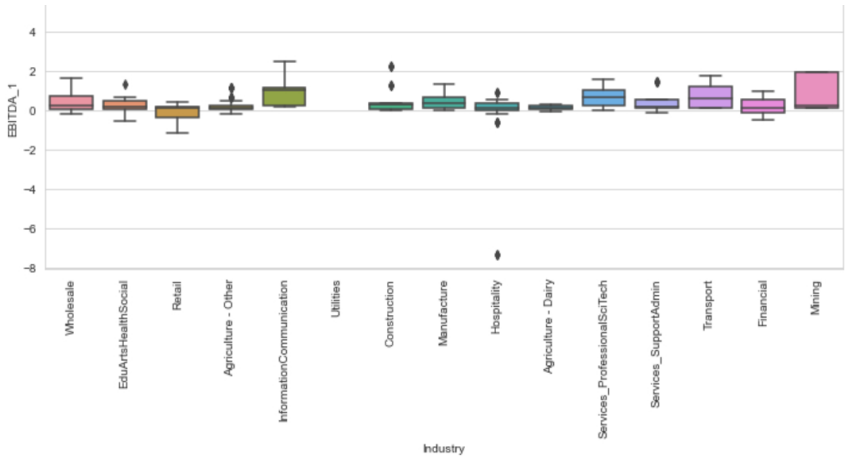

0.2
0.0
Wholesale
EduArtsHealthSocial
Retail
Agriculture - Other
InformationCommunication
Utilities
Construction
Manufacture
Hospitality
Agriculture - Dairy
Services_ProfessionalISciTech
Services_SupportAdmin
Transport
Financial
Mining
Industry

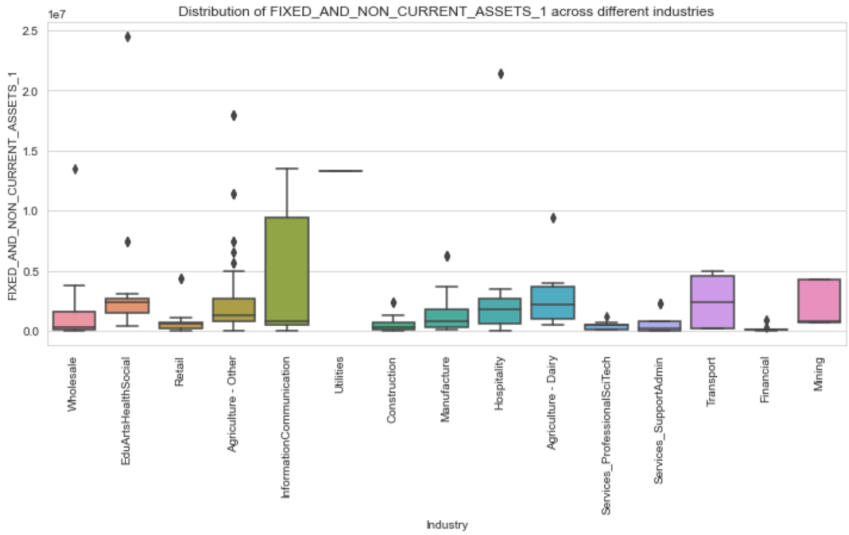

Distribution of CURRENT_ASSETS_1 across different industries
1e7
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
CURRENT_ASSETS_1
Wholesale
EduArtsHealthSocial
Retail
Agriculture - Other
InformationCommunication
Utilities
Construction
Manufacture
Hospitality
Agriculture - Dairy
Services_ProfessionalISciTech
Services_SupportAdmin
Transport
Financial
Mining
Industry

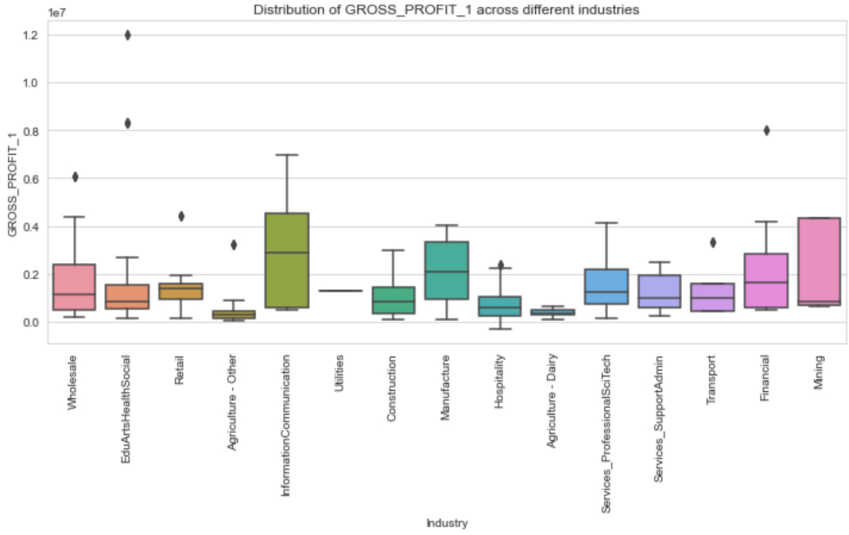

Distribution of CURRENT_LIABILITIES_1 across different industries
1e7
3.0
2.5
2.0
1.5
1.0
0.5
0.0
CURRENT_LIABILITIES_1
Wholesale
EduArtsHealthSocial
Retail
Agriculture - Other
InformationCommunication
Utilities
Construction
Manufacture
Hospitality
Agriculture - Dairy
Services_ProfessionalISciTech
Services_SupportAdmin
Transport
Financial
Mining
Industry

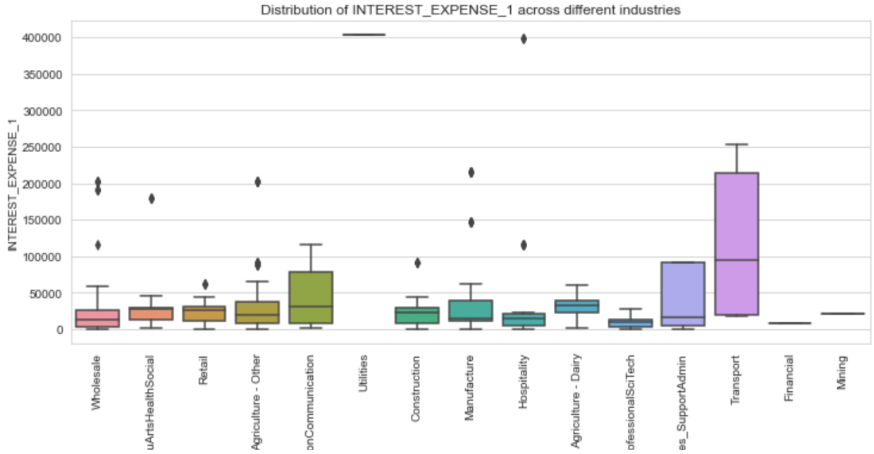

Distribution of DEPRECIATION_1 across different industries
1e6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
DEPRECIATION_1
Wholesale
EduArtsHealthSocial
Retail
Agriculture - Other
InformationCommunication
Utilities
Construction
Manufacture
Hospitality
Agriculture - Dairy
Services_ProfessionalISciTech
Services_SupportAdmin
Transport
Financial
Mining
Industry


Distribution of EBITDA_1 across different industries
1e6
8
6

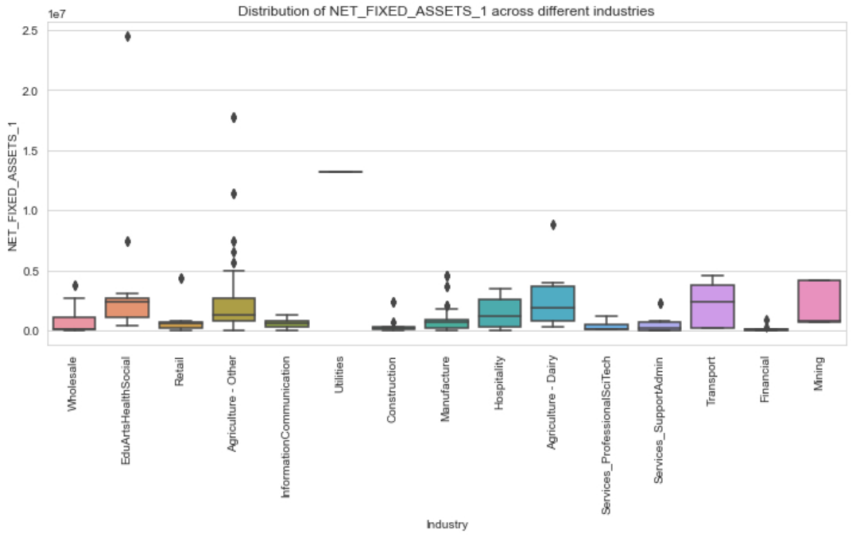

EBITDA_1
4
2
0
−2
−4
−6
−8
Wholesale
EduArtsHealthSocial
Retail
Agriculture - Other
InformationCommunication
Utilities
Construction
Manufacture
Hospitality
Agriculture - Dairy
Services_ProfessionalSciTech
Services_SupportAdmin
Transport
Financial
Mining
Industry

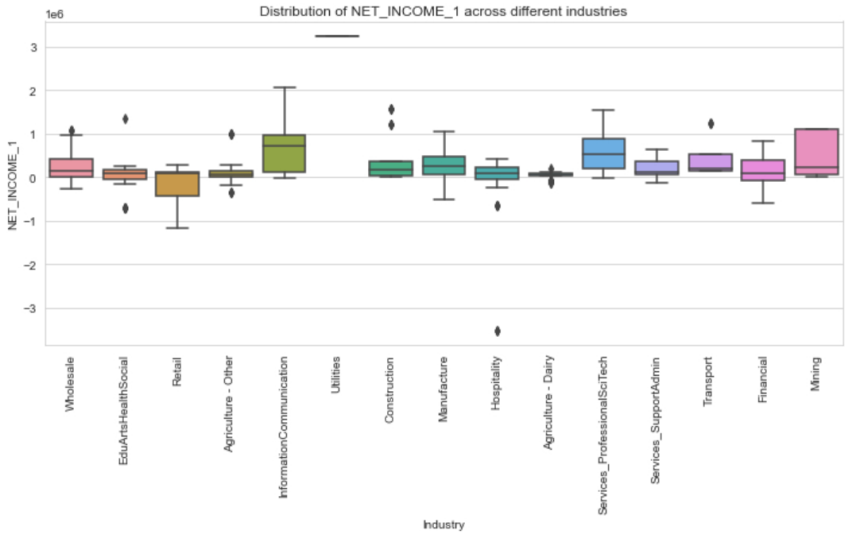

Distribution of FIXED_AND_NON_CURRENT_ASSETS_1 across different industries
1e7
FIXED_AND_NON_CURRENT_ASSETS_1
2.5
2.0
1.5
1.0
0.5
0.0
Wholesale
EduArtsHealthSocial
Retail
Agriculture - Other
InformationCommunication
Utilities
Construction
Manufacture
Hospitality
Agriculture - Dairy
Services_ProfessionalSciTech
Services_SupportAdmin
Transport
Financial
Mining
Industry

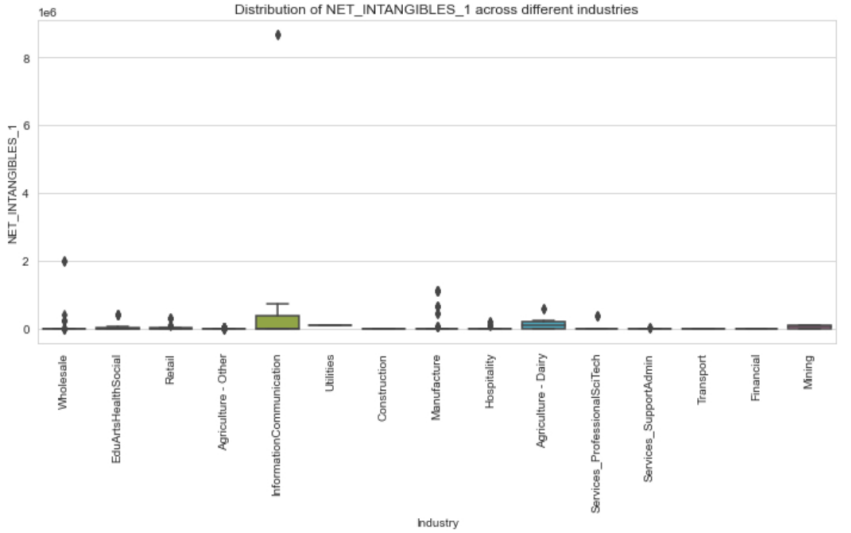

Distribution of GROSS_PROFIT_1 across different industries
1e7
GROSS_PROFIT_1
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Wholesale
EduArtsHealthSocial
Retail
Agriculture - Other
InformationCommunication
Utilities
Construction
Manufacture
Hospitality
Agriculture - Dairy
Services_ProfessionalSciTech
Services_SupportAdmin
Transport
Financial
Mining
Industry


Distribution of INTEREST_EXPENSE_1 across different industries
INTEREST_EXPENSE_1
400000
350000
300000
250000
200000
150000
100000
50000
0
Wholesale
Retail
Utilities
Construction
Manufacture
Hospitality
Transport
Financial
Mining

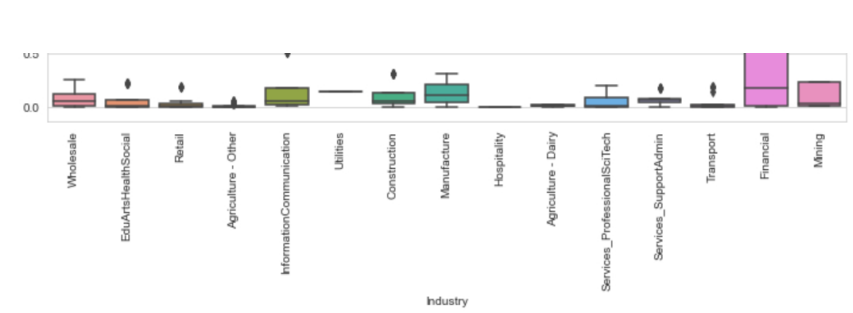

Industry

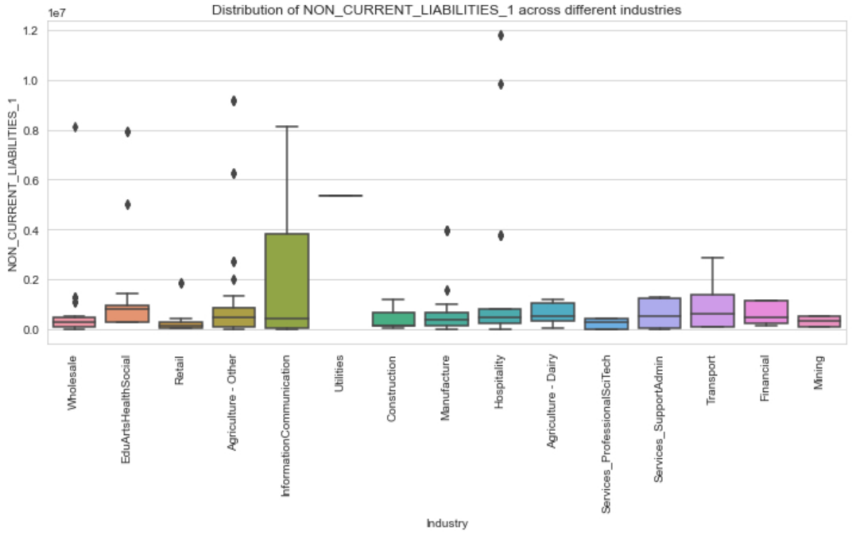

Distribution of NET_FIXED_ASSETS_1 across different industries
1e7
NET_FIXED_ASSETS_1
Wholesale
EduArtsHealthSocial
Retail
Agriculture - Other
InformationCommunication
Utilities
Construction
Manufacture
Hospitality
Agriculture - Dairy
Services_ProfessionalSciTech
Services_SupportAdmin
Transport
Financial
Mining
Industry

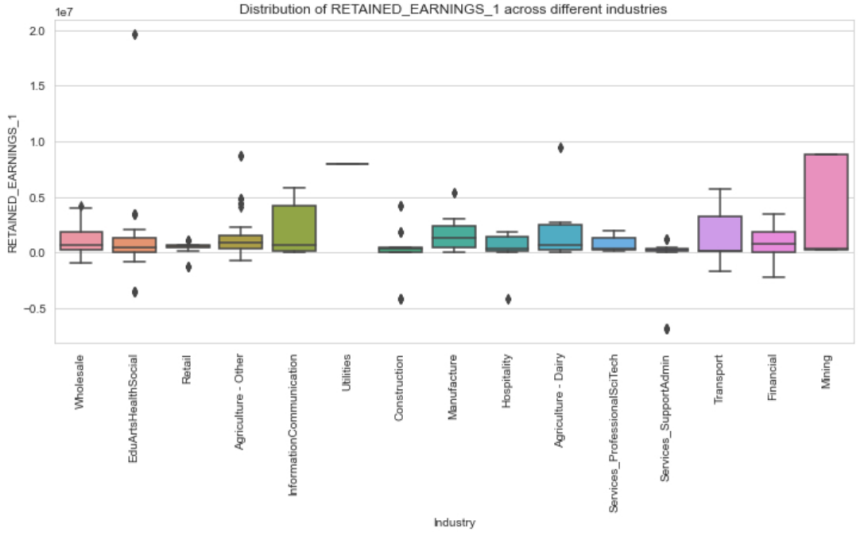

Distribution of NET_INCOME_1 across different industries
1e6
NET_INCOME_1
Wholesale
EduArtsHealthSocial
Retail
Agriculture - Other
InformationCommunication
Utilities
Construction
Manufacture
Hospitality
Agriculture - Dairy
Services_ProfessionalSciTech
Services_SupportAdmin
Transport
Financial
Mining
Industry

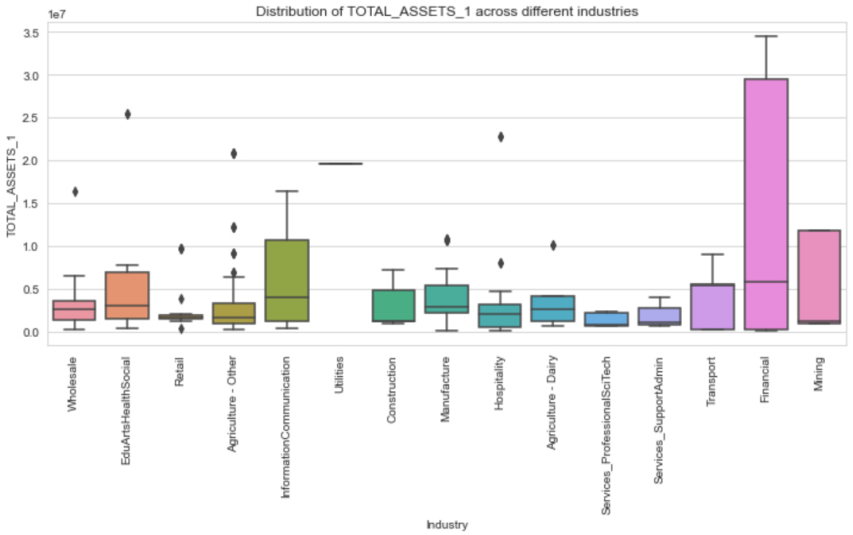

Distribution of NET_INTANGIBLES_1 across different industries
1e6
NET_INTANGIBLES_1
Wholesale
EduArtsHealthSocial
Retail
Agriculture - Other
InformationCommunication
Utilities
Construction
Manufacture
Hospitality
Agriculture - Dairy
Services_ProfessionalSciTech
Services_SupportAdmin
Transport
Financial
Mining
Industry


Distribution of NET_TRADE_RECV_1 across different industries
1e7
NET_TRADE_RECV_1

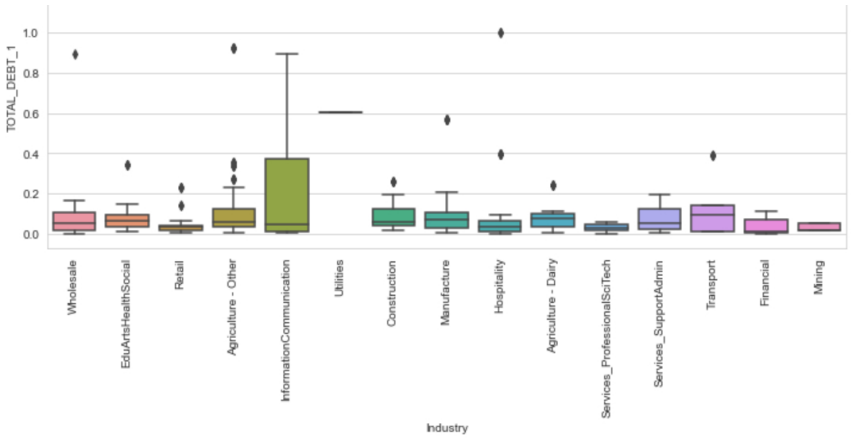

Wholesale
EduArtsHealthSocial
Retail
Agriculture - Other
InformationCommunication
Utilities
Construction
Manufacture
Hospitality
Agriculture - Dairy
Services_ProfessionalSciTech
Services_SupportAdmin
Transport
Financial
Mining
Industry

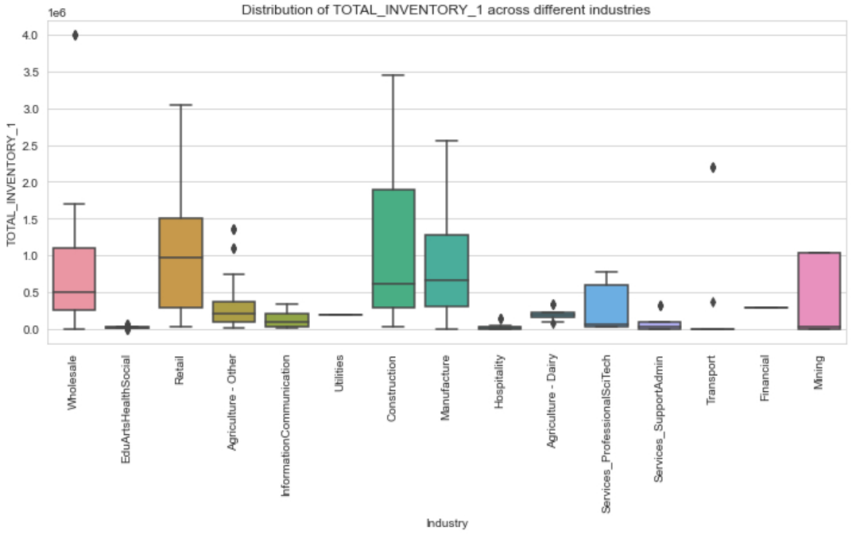

Distribution of NON_CURRENT_LIABILITIES_1 across different industries
1e7
NON_CURRENT_LIABILITIES_1
Industry

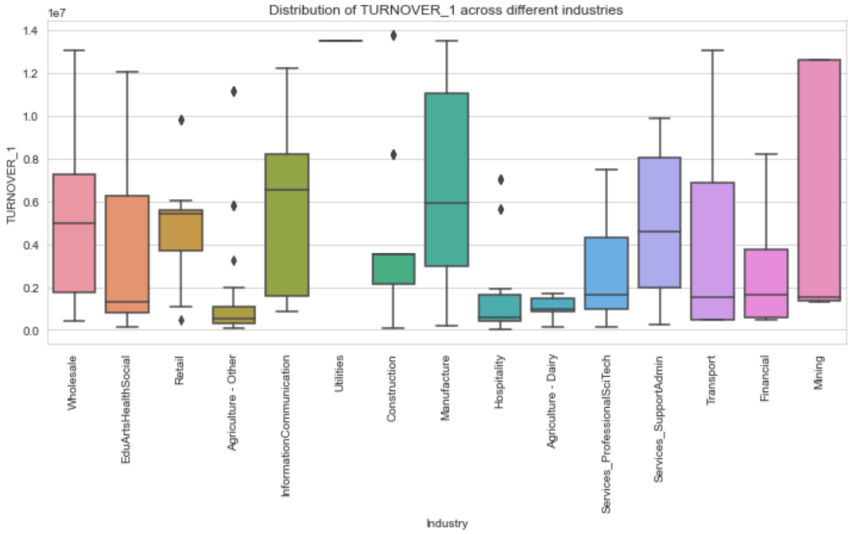

Distribution of RETAINED_EARNINGS_1 across different industries
1e7
RETAINED_EARNINGS_1
Industry

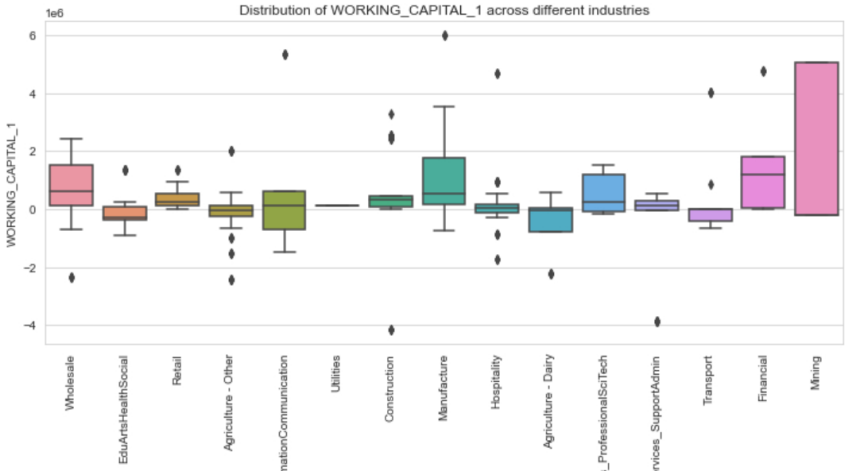

Distribution of TOTAL_ASSETS_1 across different industries
1e7
TOTAL_ASSETS_1
Industry


Distribution of TOTAL_DEBT_1 across different industries
1e7

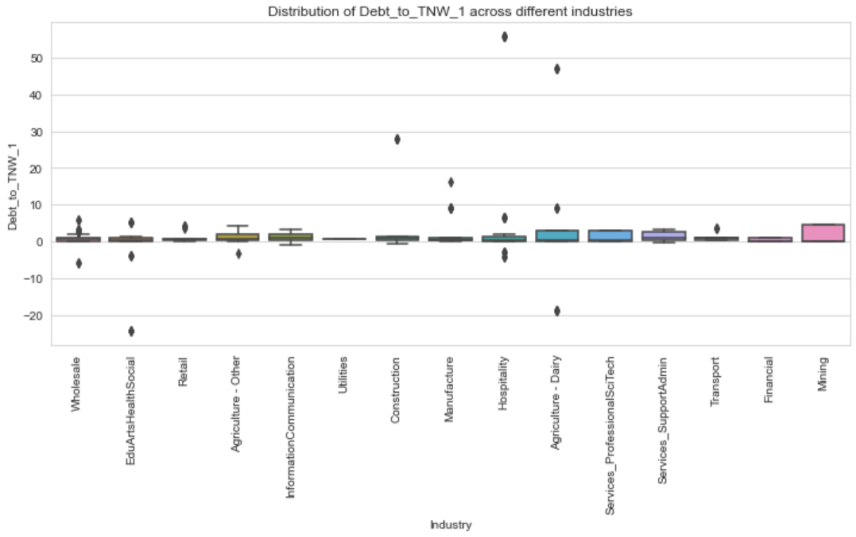

TOTAL_DEBT_1
1.0
0.8
0.6
0.4
0.2
0.0
Wholesale
EduArtsHealthSocial
Retail
Agriculture - Other
InformationCommunication
Utilities
Construction
Manufacture
Hospitality
Agriculture - Dairy
Services_ProfessionalSciTech
Services_SupportAdmin
Transport
Financial
Mining
Industry

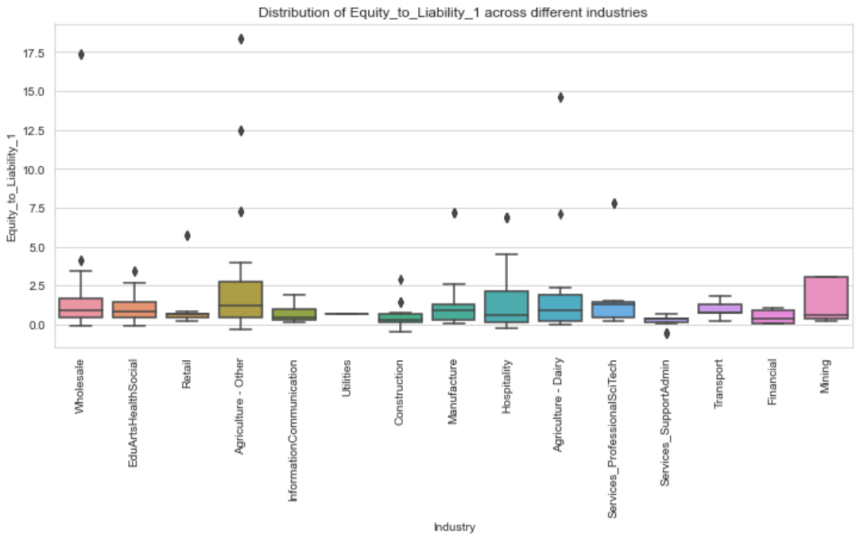

Distribution of TOTAL_INVENTORY_1 across different industries
1e6
TOTAL_INVENTORY_1
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Wholesale
EduArtsHealthSocial
Retail
Agriculture - Other
InformationCommunication
Utilities
Construction
Manufacture
Hospitality
Agriculture - Dairy
Services_ProfessionalSciTech
Services_SupportAdmin
Transport
Financial
Mining
Industry

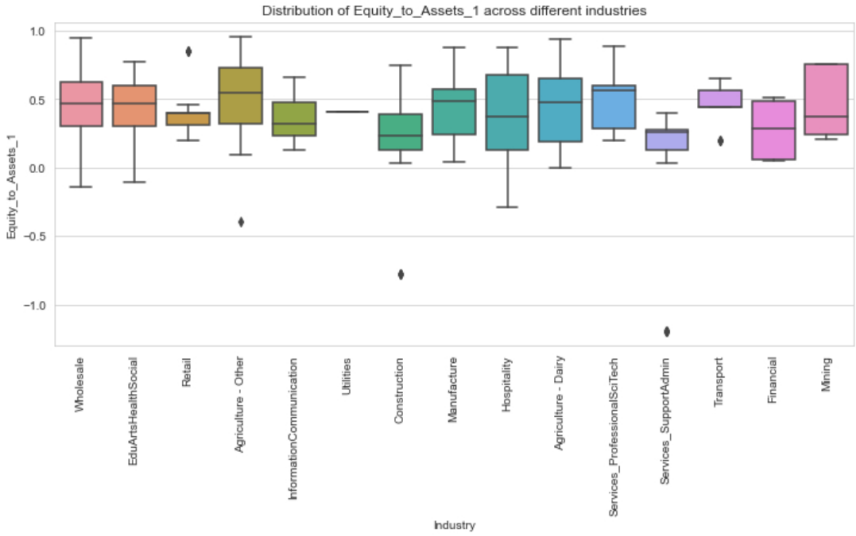

Distribution of TURNOVER_1 across different industries
1e7
TURNOVER_1
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Wholesale
EduArtsHealthSocial
Retail
Agriculture - Other
InformationCommunication
Utilities
Construction
Manufacture
Hospitality
Agriculture - Dairy
Services_ProfessionalSciTech
Services_SupportAdmin
Transport
Financial
Mining
Industry

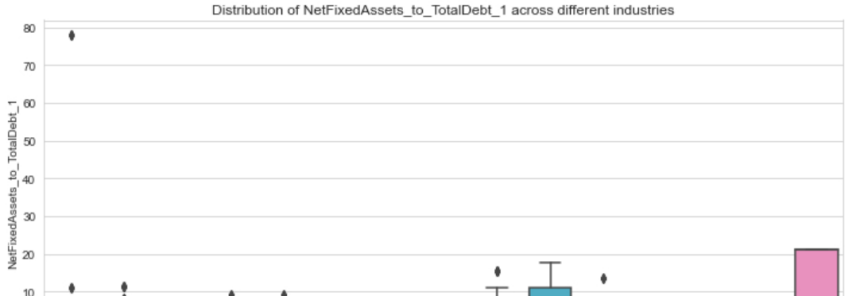

Distribution of WORKING_CAPITAL_1 across different industries
1e6
WORKING_CAPITAL_1
6
4
2
0
−2
−4
Wholesale
EduArtsHealthSocial
Retail
Agriculture - Other
Utilities
Construction
Manufacture
Hospitality
Agriculture - Dairy
Transport
Financial
Mining

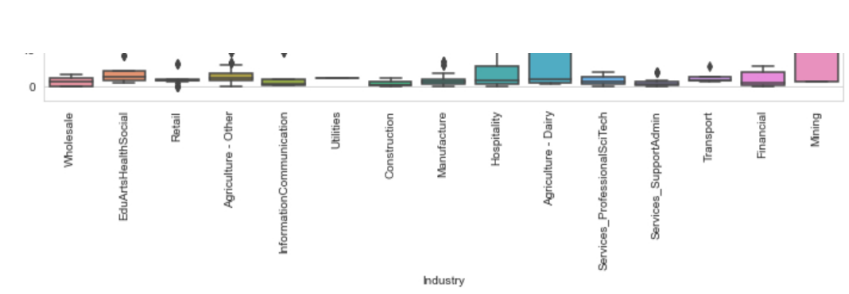

Industry

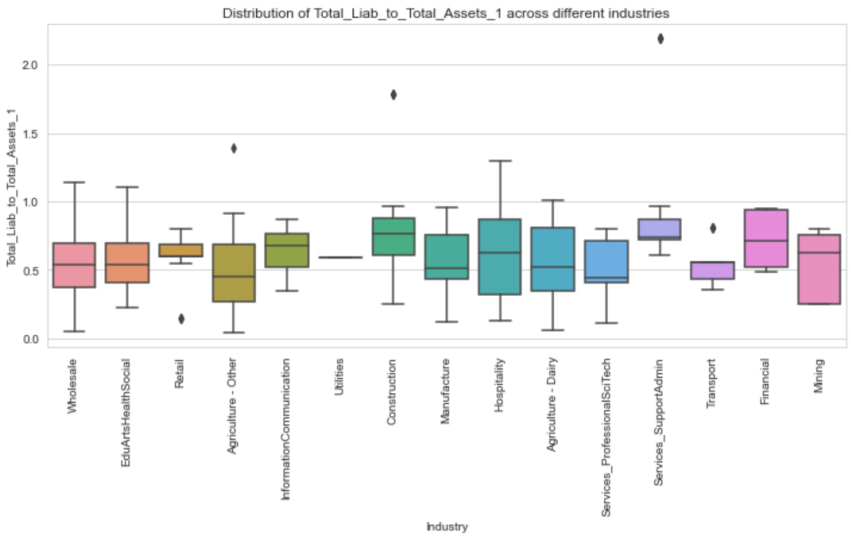

Distribution of Debt_to_TNW_1 across different industries
Debt_to_TNW_1
50
40
30
20
10
0
−10
−20
Wholesale
EduArtsHealthSocial
Retail
Agriculture - Other
InformationCommunication
Utilities
Construction
Manufacture
Hospitality
Agriculture - Dairy
Services_ProfessionalSciTech
Services_SupportAdmin
Transport
Financial
Mining
Industry

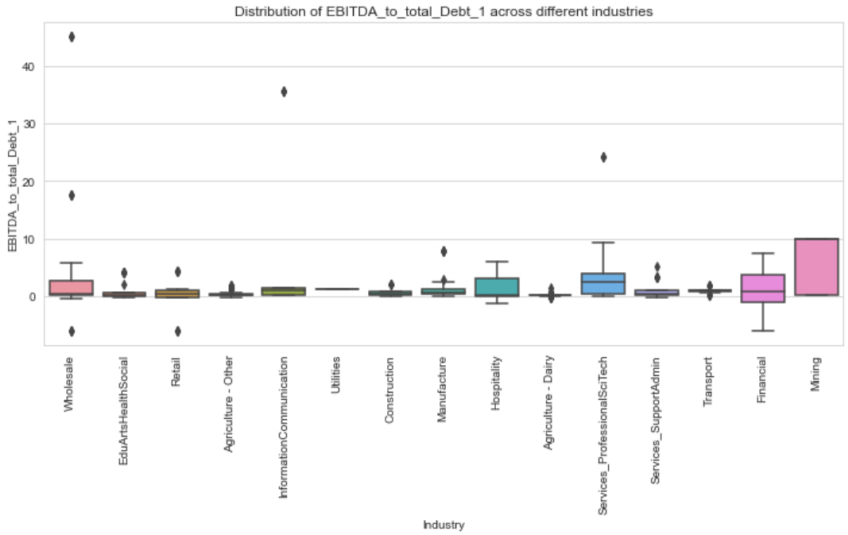

Distribution of Equity_to_Liability_1 across different industries
Equity_to_Liability_1
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
Wholesale
EduArtsHealthSocial
Retail
Agriculture - Other
InformationCommunication
Utilities
Construction
Manufacture
Hospitality
Agriculture - Dairy
Services_ProfessionalSciTech
Services_SupportAdmin
Transport
Financial
Mining
Industry

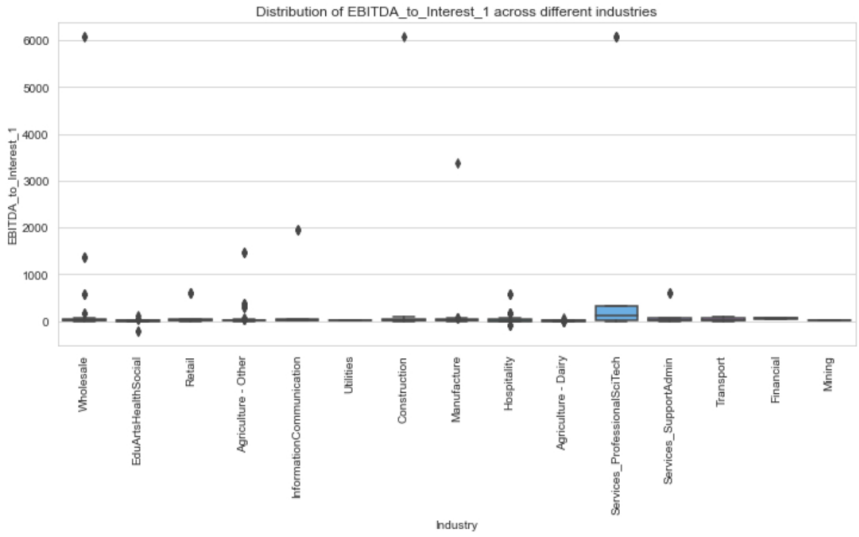

Distribution of Equity_to_Assets_1 across different industries
Equity_to_Assets_1
1.0
0.5
0.0
−0.5
−1.0
Wholesale
EduArtsHealthSocial
Retail
Agriculture - Other
InformationCommunication
Utilities
Construction
Manufacture
Hospitality
Agriculture - Dairy
Services_ProfessionalSciTech
Services_SupportAdmin
Transport
Financial
Mining
Industry


Distribution of NetFixedAssets_to_TotalDebt_1 across different industries
NetFixedAssets_to_TotalDebt_1
80
70
60
50
40
30
20
10

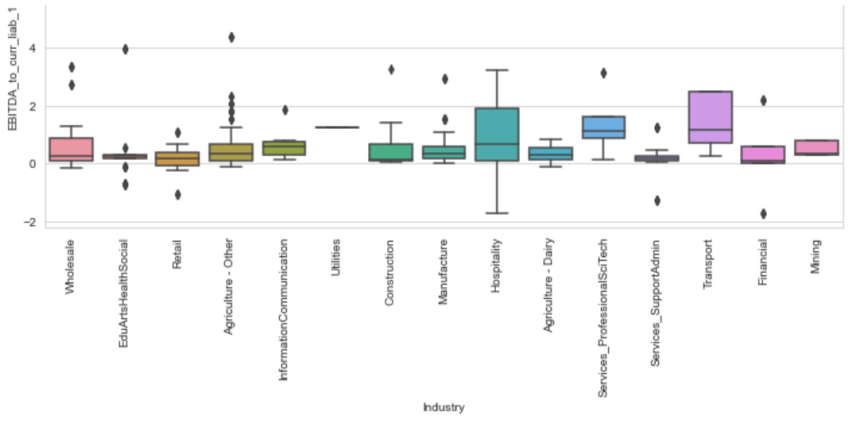

Wholesale
EduArtsHealthSocial
Retail
Agriculture - Other
InformationCommunication
Utilities
Construction
Manufacture
Hospitality
Agriculture - Dairy
Services_ProfessionalSciTech
Services_SupportAdmin
Transport
Financial
Mining
Industry

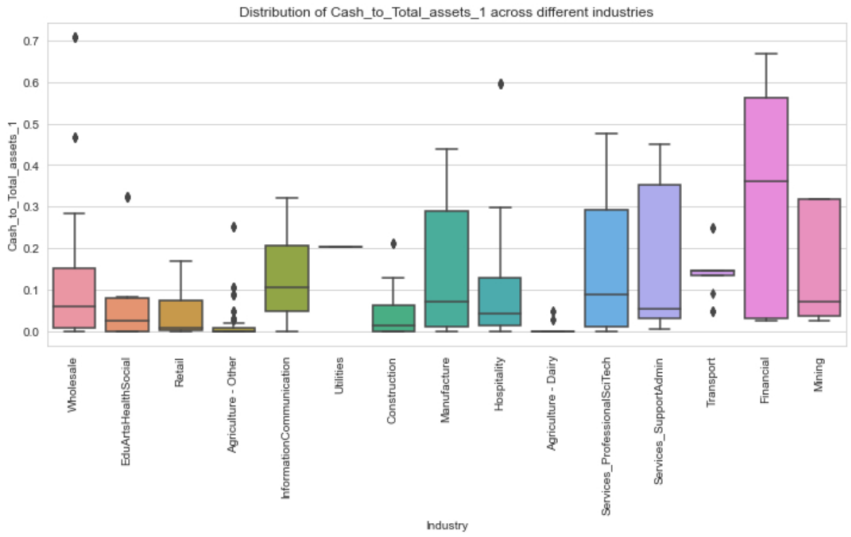

Distribution of Total_Liab_to_Total_Assets_1 across different industries
Total_Liab_to_Total_Assets_1
Industry

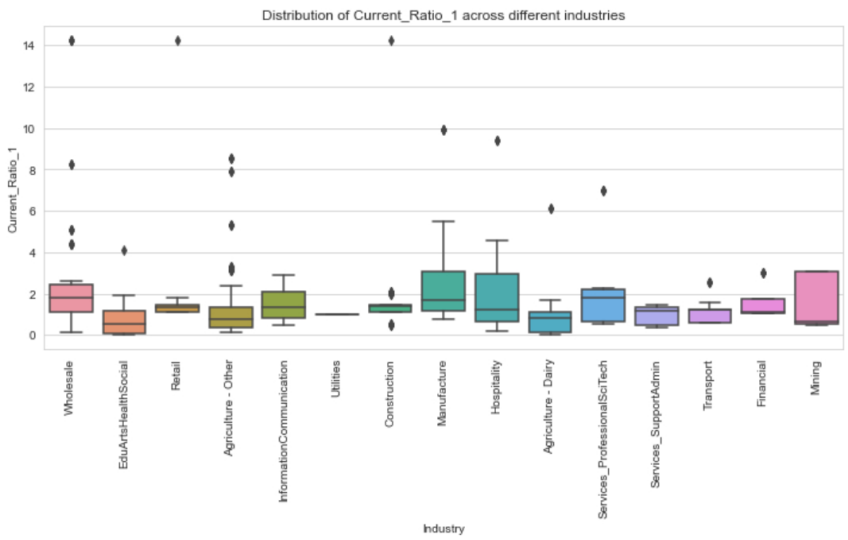

Distribution of EBITDA_to_total_Debt_1 across different industries
EBITDA_to_total_Debt_1
Industry

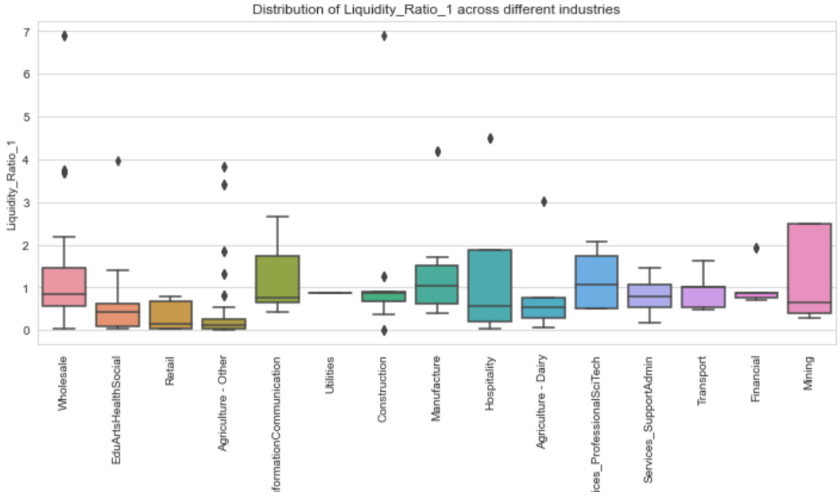

Distribution of EBITDA_to_Interest_1 across different industries
EBITDA_to_Interest_1
Industry

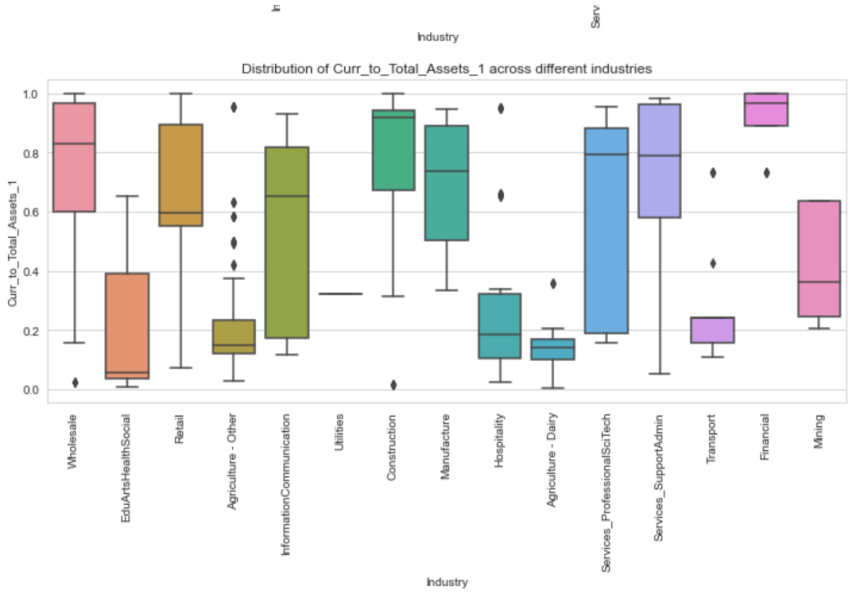

Distribution of EBITDA_to_curr_liab_1 across different industries

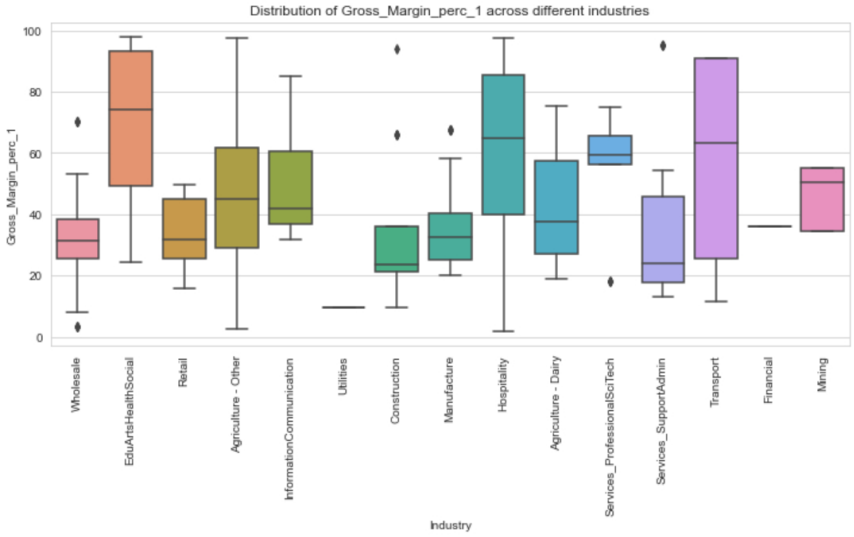

EBITDA_to_curr_liab_1
Wholesale
EduArtsHealthSocial
Retail
Agriculture - Other
InformationCommunication
Utilities
Construction
Manufacture
Hospitality
Agriculture - Dairy
Services_ProfessionalSciTech
Services_SupportAdmin
Transport
Financial
Mining
Industry

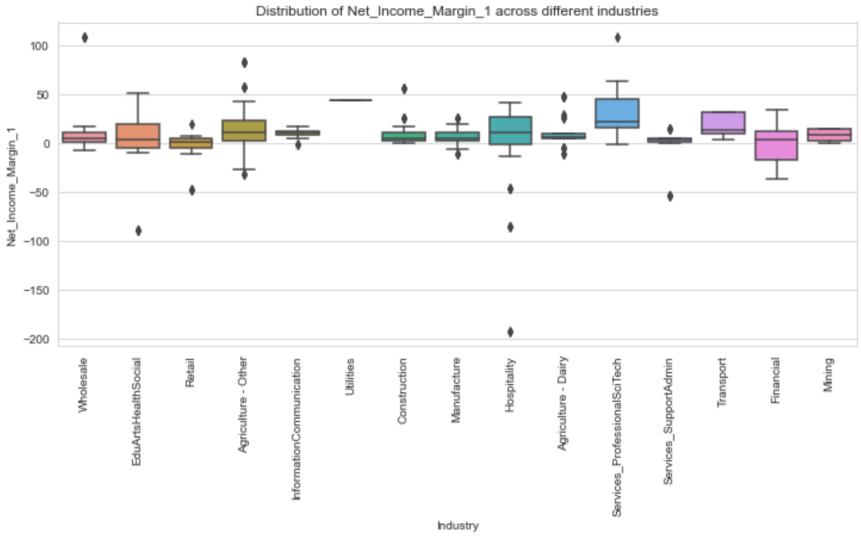

Distribution of Cash_to_Total_assets_1 across different industries
Cash_to_Total_assets_1
Industry

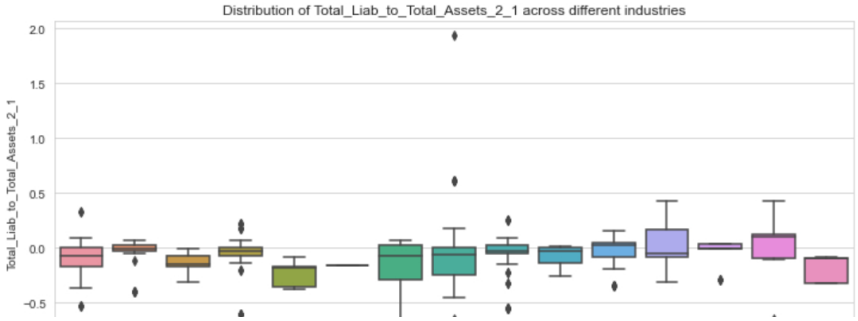

Distribution of Current_Ratio_1 across different industries
Current_Ratio_1
Industry

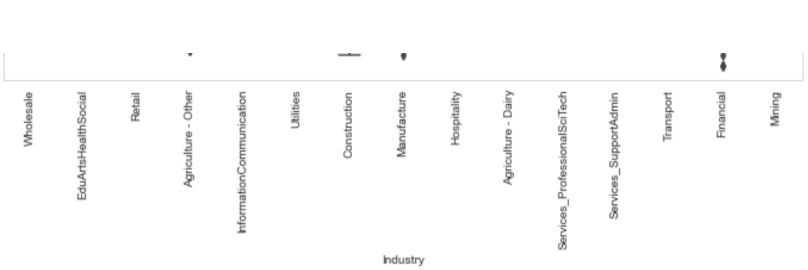

Distribution of Liquidity_Ratio_1 across different industries
Liquidity_Ratio_1


Industry
Distribution of Curr_to_Total_Assets_1 across different industries
Curr_to_Total_Assets_1
1.0
0.8
0.6
0.4
0.2
0.0
Wholesale
EduArtsHealthSocial
Retail
Agriculture - Other
InformationCommunication
Utilities
Construction
Manufacture
Hospitality
Agriculture - Dairy
Services_ProfessionalSciTech
Services_SupportAdmin
Transport
Financial
Mining
Industry


Distribution of Gross_Margin_perc_1 across different industries
Gross_Margin_perc_1
100
80
60
40
20
0
Wholesale
EduArtsHealthSocial
Retail
Agriculture - Other
InformationCommunication
Utilities
Construction
Manufacture
Hospitality
Agriculture - Dairy
Services_ProfessionalSciTech
Services_SupportAdmin
Transport
Financial
Mining
Industry


Distribution of Net_Income_Margin_1 across different industries
Net_Income_Margin_1
100
50
0
−50
−100
−150
−200
Wholesale
EduArtsHealthSocial
Retail
Agriculture - Other
InformationCommunication
Utilities
Construction
Manufacture
Hospitality
Agriculture - Dairy
Services_ProfessionalSciTech
Services_SupportAdmin
Transport
Financial
Mining
Industry


Distribution of Total_Liab_to_Total_Assets_2_1 across different industries
Total_Liab_to_Total_Assets_2_1
2.0
1.5
1.0
0.5
0.0
−0.5



- The numerical features varies across different industries
- so we will replace the missing values with the median of their respective industries

[11]:
```
#missing value imputation with median
for feature in numerical_features:
    data[feature].fillna(data.groupby('Industry')[feature].transform('median'),inplace=True)
```

[12]:
```
data.isna().any().any()
```

[12]:
```
False
```

Outlier Handling

handling the outliers by quantile based flooring and capping with their respective industries

[13]:
```
#outlier treatment
data = data.groupby(data.Industry.values).transform(lambda x: np.clip(x,x.quantile(0.05),x.quantile(0.95))if x.name in numerical_features else x)
data.head()
```

[13]:

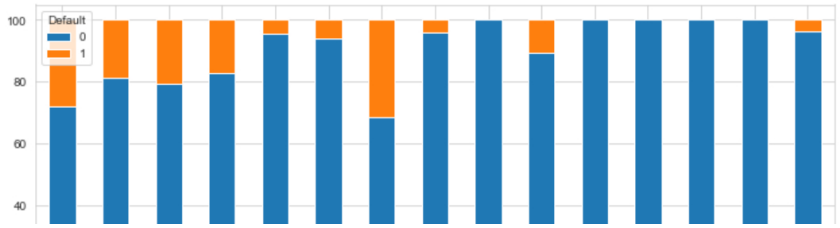
	Industry	CASH_AND_CASH_EQUIVALENTS_floored1	COST_OF_GOODS_SOLD_1	CURRENT_ASSETS_1	CURRENT_LIABILITIES_1	DEPRECIATION_1	EBITDA_1	FIXED
0	Wholesale	323773.0	2995379.0	476381.0	516192.0	78247.0	357842.0	
1	EduArtsHealthSocial	18593.0	11753.0	21913.0	257405.0	6988.0	42274.0	
2	Retail	202720.0	3420276.0	665190.0	373352.0	76899.0	406223.0	
3	Agriculture - Other	87515.0	2349333.7	1753108.7	2153519.0	134105.0	657162.4	
4	EduArtsHealthSocial	272938.0	237933.0	381958.0	681622.0	81766.0	-515748.0	

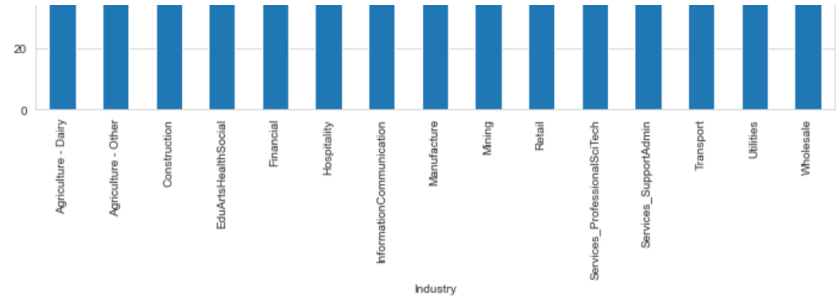
EDA

[15]:
```
prop_by_industry = pd.crosstab(data['Industry'],data['Default']).apply(lambda x:x/x.sum()*100,axis=1)
print(prop_by_industry)
prop_by_industry.plot(kind='bar',stacked=True)
plt.xticks(rotation=90)
```

```
Default                             0          1
Industry
Agriculture - Dairy            72.000000  28.000000
Agriculture - Other            81.300813  18.699187
Construction                   79.310345  20.689655
EduArtsHealthSocial            82.926829  17.073171
Financial                      95.454545   4.545455
Hospitality                    94.000000   6.000000
InformationCommunication       68.421053  31.578947
Manufacture                    95.774648   4.225352
Mining                        100.000000   0.000000
Retail                         89.285714  10.714286
Services_ProfessionalSciTech  100.000000   0.000000
Services_SupportAdmin         100.000000   0.000000
Transport                     100.000000   0.000000
Utilities                     100.000000   0.000000
Wholesale                      96.153846   3.846154
```

[15]:
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14]),
 [Text(0, 0, 'Agriculture - Dairy'),
  Text(1, 0, 'Agriculture - Other'),
  Text(2, 0, 'Construction'),
  Text(3, 0, 'EduArtsHealthSocial'),
  Text(4, 0, 'Financial'),
  Text(5, 0, 'Hospitality'),
  Text(6, 0, 'InformationCommunication'),
  Text(7, 0, 'Manufacture'),
  Text(8, 0, 'Mining'),
  Text(9, 0, 'Retail'),
  Text(10, 0, 'Services_ProfessionalSciTech'),
  Text(11, 0, 'Services_SupportAdmin'),
  Text(12, 0, 'Transport'),
  Text(13, 0, 'Utilities'),
  Text(14, 0, 'Wholesale')])
```

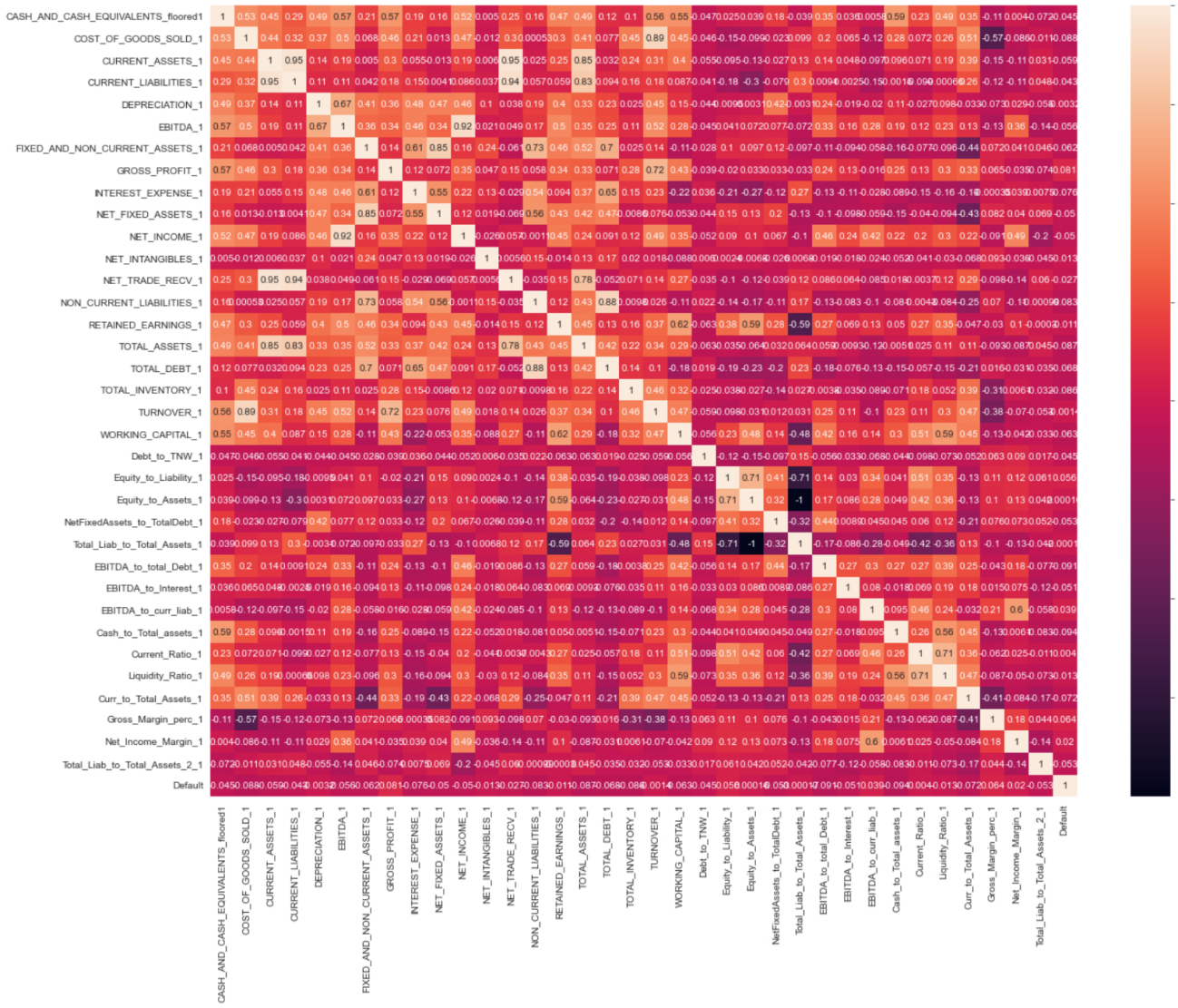




- Among all the industries InformationCommunication and Agriculture-diary has the highest default rate
- Agriculture other , construction and EduArtsHealthSocial have default rate around 20%

[16]:
```
plt.figure(figsize=(20,15))
sns.heatmap(data.corr(),annot=True)
```

[16]:
```
<AxesSubplot:>
```



some of the features are highly correlated to each other i,e Multicollinearity

[17]:
```
#scaling the data using standard scaler

scaler = StandardScaler()
data_trf = data.copy()
data_trf[numerical_features] = scaler.fit_transform(data_trf[numerical_features])
data_trf.head()
```

[17]:

	Industry	CASH_AND_CASH_EQUIVALENTS_floored1	COST_OF_GOODS_SOLD_1	CURRENT_ASSETS_1	CURRENT_LIABILITIES_1	DEPRECIATION_1	EBITDA_1	FIXED
0	Wholesale	0.082289	0.227524	-0.353960	-0.277880	-0.099619	-0.017990	
1	EduArtsHealthSocial	-0.293261	-0.877496	-0.472153	-0.350854	-0.563862	-0.460186	
2	Retail	-0.066677	0.384889	-0.304856	-0.318159	-0.108401	0.049805	
3	Agriculture - Other	-0.208447	-0.011747	-0.021921	0.183818	0.264288	0.401438	

4	EduArtsHealthSocial	0.019732	-0.793728	-0.378516	-0.231232	-0.076693	-1.242125

[18]:
```
#seperating independent and dependent variables
X =data_trf.loc[:,data_trf.columns != 'Default']
X.drop(['Industry'],axis=1,inplace=True)
y = data_trf['Default']
print('shape of X',X.shape)
print('shape of y',y.shape)
```

```
shape of X (573, 35)
shape of y (573,)
```

[19]:
```
# using variance inflation factor vif for removing highly correlated features
# using a threshold of 4 for VIF
from statsmodels.stats.outliers_influence import variance_inflation_factor
features = list(X.columns)
while True:
    vif = pd.DataFrame()
    df = X[features]
    vif['variables'] = df.columns
    vif['VIF'] = [variance_inflation_factor(df.values,i)for i in range(df.shape[1])]
    vif.sort_values(by='VIF',ascending=False,inplace=True,ignore_index=True)
    if vif['VIF'][0] > 4:
        print(vif['variables'][0], 'has been removed with a vif of' , vif['VIF'][0])
        features.remove(vif['variables'][0])
    else:
        break
print('NO of Remaining features after VIF',len(features))
```

```
Equity_to_Assets_1 has been removed with a vif of 14735685.088588918
CURRENT_LIABILITIES_1 has been removed with a vif of 759.1328261534444
CURRENT_ASSETS_1 has been removed with a vif of 247.62206643656504
TOTAL_ASSETS_1 has been removed with a vif of 73.88002943534978
EBITDA_1 has been removed with a vif of 32.962967868964235
TURNOVER_1 has been removed with a vif of 19.8759711335175
FIXED_AND_NON_CURRENT_ASSETS_1 has been removed with a vif of 13.991232694600122
TOTAL_DEBT_1 has been removed with a vif of 8.815008158029718
WORKING_CAPITAL_1 has been removed with a vif of 5.551574543389508
Current_Ratio_1 has been removed with a vif of 4.611448736375268
NET_INCOME_1 has been removed with a vif of 4.484343164759755
CASH_AND_CASH_EQUIVALENTS_floored1 has been removed with a vif of 4.047231973943805
NO of Remaining features after VIF 23
```

[20]:
```
X =X[features]
X.shape
```

[20]:
```
(573, 23)
```

MODEL BUILDING

[21]:
```
#building the baseline model

x_train,x_test,y_train,y_test=train_test_split(X,y,test_size=0.1,random_state=122)
print("Target Class distribution in Training split:",Counter(y_train))
print("Target Class distribution in Testing split:",Counter(y_test))
```

```
Target Class distribution in Training split: Counter({0: 460, 1: 55})
Target Class distribution in Testing split: Counter({0: 51, 1: 7})
```

[22]:
```
def metrics(model,testdata,actual,predicted):
    ''' This function returns evaluation metrics by comparing actual and predicted outcomes'''
    print('Testing F1-Score:',f1_score(actual,predicted))
    probpred = model.predict_proba(testdata)
    print('AUC Score:',roc_auc_score(actual,probpred[:,-1]))
    print('Classification Report \n',classification_report(actual,predicted))
    fpr , tpr ,thresh = roc_curve(actual,probpred[:,1],pos_label=1)
    plt.figure(figsize=(12,5))
    plt.subplot(1,2,1)
    sns.heatmap(confusion_matrix(actual,predicted),annot=True,fmt='.1f')
    plt.title('Confusion matrix')
    plt.subplot(1,2,2)
    plt.plot(fpr,tpr,linestyle='--',color='r')
    plt.title('ROC Curve')
```

Training the model

[23]:
```
classifier = RandomForestClassifier()
classifier.fit(x_train,y_train)
Trainpred = classifier.predict(x_train)
print('Training F1-Score:',f1_score(y_train,Trainpred))
```

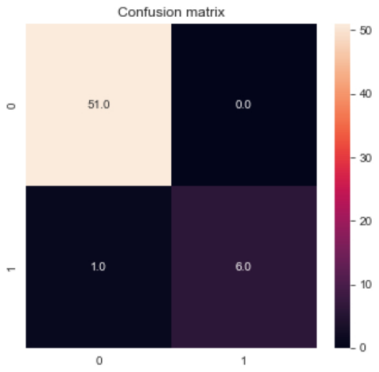
```
Training F1-Score: 0.9734513274336283
```

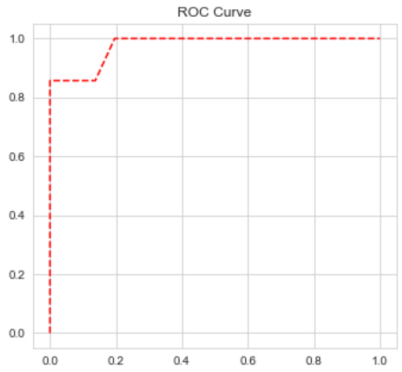
[24]:
```
#predicting on the test data
pred = classifier.predict(x_test)
metrics(classifier,x_test,y_test,pred)
```

```
Testing F1-Score: 0.923076923076923
AUC Score: 0.9761904761904762
Classification Report
              precision    recall  f1-score   support

           0       0.98      1.00      0.99        51
           1       1.00      0.86      0.92         7
```

```
    accuracy                           0.98        58
   macro avg       0.99      0.93      0.96        58
weighted avg       0.98      0.98      0.98        58
```





- The testing F1-score is 92% which is good but,
- the training F1-Score is higher i,e 98% , this shows that our model is slightly overfitting

K-FOLD CROSS VALIDATION to reduce the overfitting

[25]:
```
cv = StratifiedKFold(n_splits=5,random_state=22,shuffle=True)
scores = cross_val_score(classifier,X,y,scoring='f1',n_jobs=-1)
print("F1_Score:",np.mean(scores),"std:",np.std(scores))
```

```
F1_Score: 0.9426666666666668 std: 0.05681940200710008
```

- The testing f1-score now has been improved from 92% to 94.1%
- This slightly reduces the problem of overfitting

Hyperparameter tuning

[26]:
```
params=param_grid = {'bootstrap': [True],
'max_depth': [1,4,10, 20, 30, 40, 50, 60, 70, 80, 90, 100, None],
'max_features': ['auto', 'sqrt'],
'min_samples_leaf': [1, 2, 4],
'min_samples_split': [2, 5, 10],
'n_estimators': [200, 400, 600, 800, 1000]
}
grid = GridSearchCV(classifier,params,scoring='f1',n_jobs=-1,cv=5,verbose=2)
grid.fit(X,y)
```

```
Fitting 5 folds for each of 1170 candidates, totalling 5850 fits
```

[26]:
```
GridSearchCV(cv=5, estimator=RandomForestClassifier(), n_jobs=-1,
             param_grid={'bootstrap': [True],
                         'max_depth': [1, 4, 10, 20, 30, 40, 50, 60, 70, 80, 90,
                                       100, None],
                         'max_features': ['auto', 'sqrt'],
                         'min_samples_leaf': [1, 2, 4],
                         'min_samples_split': [2, 5, 10],
                         'n_estimators': [200, 400, 600, 800, 1000]},
             scoring='f1', verbose=2)
```

[27]:
```
print('Best params:',grid.best_params_)
print('Best Score:',grid.best_score_)
grid.best_estimator_
```

```
Best params: {'bootstrap': True, 'max_depth': 20, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 400}
Best Score: 0.9426666666666668
```

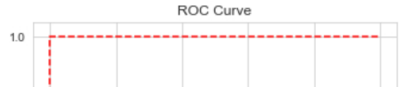
[27]:
```
RandomForestClassifier(max_depth=20, n_estimators=400)
```

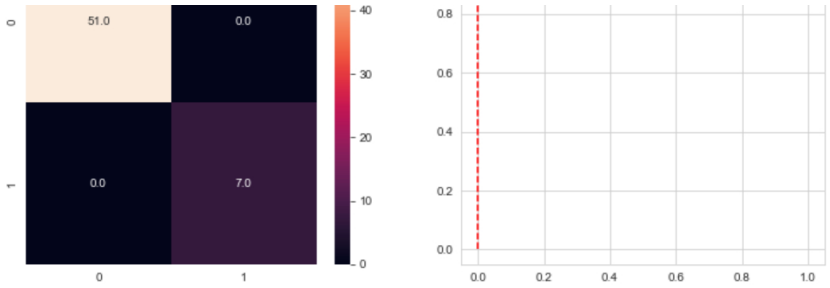
[28]:
```
train_pred = grid.predict(x_train)
test_pred = grid.predict(x_test)
print('Training score after hyperparameter tuning:',f1_score(y_train,train_pred))
metrics(grid,x_test,y_test,test_pred)
```

```
Training score after hyperparameter tuning: 0.9734513274336283
Testing F1-Score: 1.0
AUC Score: 1.0
Classification Report
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        51
           1       1.00      1.00      1.00         7

    accuracy                           1.00        58
   macro avg       1.00      1.00      1.00        58
weighted avg       1.00      1.00      1.00        58
```

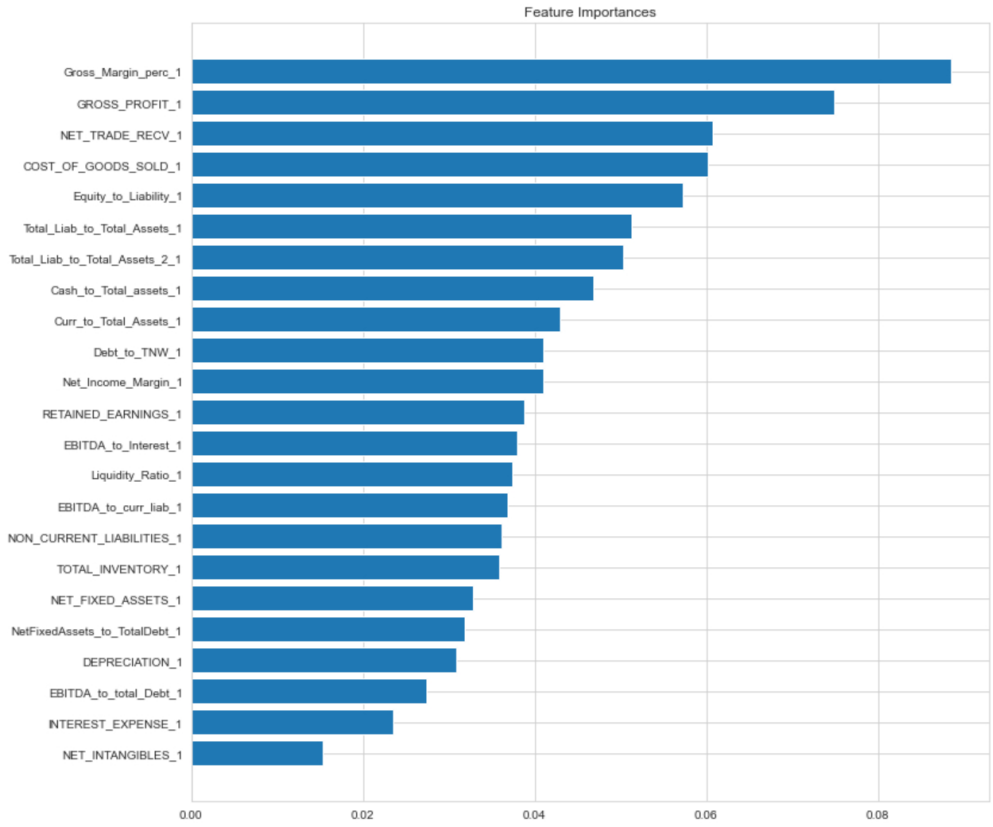




- *The testing f1_score improved gradually from 94% to 99%*
- *The Training f1_score is 97%*
- *The overfitting problem has been removed*

[29]:
```
feat_imps=pd.DataFrame(classifier.feature_importances_,index=x_train.columns,columns=['Scores'])
feat_imps.sort_values(by='Scores',inplace=True,ascending=True)
plt.figure(figsize=(12,12))
plt.barh(width=feat_imps['Scores'],y=feat_imps.index)
plt.title('Feature Importances')
```

[29]:
```
Text(0.5, 1.0, 'Feature Importances')
```



[32]:
```
print('The most important features for predicting the default are:',list(feat_imps.index[:5]))
```

```
The most important features for predicting the default are: ['NET_INTANGIBLES_1', 'INTEREST_EXPENSE_1', 'EBITDA_to_total_Debt_1', 'DEPRECIATION_1', 'NetFixedAssets_to_TotalDebt_1']
```