Loan Prediction Analysis

Dissertation submitted in fulfilment of the requirements for the Degree of

B. Tech Computer Science and Engineering Data Science (AI and ML)

By

K.Ranjith Kumar Reddy

12111200

Supervisor

Ved Prakash Chaubey

School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab (India)

Month…………… Year ………

DECLARATION STATEMENT

I hereby declare that the research work reported in the dissertation/dissertation proposal entitled "**LOAN PREDICTION ANALYSIS**" in partial fulfilment of the requirement for the award of Degree for Bachelor of Technology in Computer Science and Engineering Data Science (AI and ML) at Lovely Professional University, Phagwara, Punjab isan authentic work carried out under supervision of my research supervisor Mr. Ved Prakash Chaubey. I have not submitted this work elsewhere for any degree or diploma.

I understand that the work presented herewith is in direct compliance with Lovely Professional University’s Policy on plagiarism, intellectual property rights, and highest standards of moral and ethical conduct. Therefore, to the best of my knowledge, the content of this dissertation represents authentic and honest research effort conducted, in its entirety, by me.I am fully responsible for the contents of my dissertation work.

Signature of Candidate

K.Ranjith Kumar Reddy

RK21UTA17

SUPERVISOR'S CERTIFICATE

This is to certify that the work reported in the B. Tech Dissertation proposal entitled "**LOAN PREDICTION ANALYSIS"**, submitted by **K.Ranjith Kumar Reddy** at **Lovely Professional University, Phagwara, India** is a bonafide record of his original work carried out under my supervision. This work hasnot been submitted elsewhere for any other degree.

Signature of Supervisor

(Name of Supervisor)

Date:

Counter Signed by:

1) ***Concerned HOD:***
   HoD's Signature: _________________

   HoD Name: _____________________

   Date: ____________________

2) *Neutral Examiners:*

   External Examiner

   Signature:_________________

   Name:____________________

   Affiliation:________________

   Date:_____________________

   Internal

   Examiner

   Signature:_________________

   Name:____________________

   Date: ____________________

TABLE OF CONTENTS

DATASET DESCRIPTION:

The dataset frequently includes details about the customer's gender, marital status, income, employment history, educational background, loan amount, loan term, and credit history. Other elements that could influence the decision to approve the loan include the customer's age, the loan's objective, and the nature of the given collateral.

The loan prediction dataset is frequently used to train machine learning models that forecast the probability of loan approval based on the information provided by the consumer. This can assist banks and other financial organizations in automating and streamlining the loan approval process.

Description of Columns

- ID = Customer ID of Applicant
- year = Year of Application
- loan limit = maximum avaliable amount of the loan allowed to be taken
- Gender = sex type
- approv_in_adv = Is loan pre-approved or not
- loan_type = Type of loan
- loan_purpose = the reason you want to borrow money
- Credit_Worthiness = is how a lender determines that you will default on your debt obligations, or how worthy you are to receive new credit.
- open_credit = is a pre-approved loan between a lender and a borrower. It allows the borrower to make repeated withdrawals up to a certain limit.
- business_or_commercial = Usage type of the loan amount
- loan_amount = The exact loan amount
- rate_of_interest = is the amount a lender charges a borrower and is a percentage of the principal—the amount loaned.
- Interest_rate_spread = the difference between the interest rate a financial institution pays to depositors and the interest rate it receives from loans
- Upfront_charges = Fee paid to a lender by a borrower as consideration for making a new loan

- term = the loan's repayment period
- Neg_ammortization = refers to a situation when a loan borrower makes a payment less than the standard installment set by the bank.
- interest_only = amount of interest only without principles
- lump_sum_payment = is an amount of money that is paid in one single payment rather than in installments.
- property_value = the present worth of future benefits arising from the ownership of the property
- construction_type = Collateral construction type
- occupancy_type = classifications refer to categorizing structures based on their usage
- Secured_by = Type of Collatoral
- total_units = number of unites
- income = refers to the amount of money, property, and other transfers of value received over a set period of time
- credit_type = type of credit
- co-applicant_credit_type = is an additional person involved in the loan application process. Both applicant and co-applicant apply and sign for the loan
- age = applicant's age
- submission_of_application = Ensure the application is complete or not
- LTV = life-time value (LTV) is a prognostication of the net profit
- Region = applicant's place
- Security_Type = Type of Collatoral
- status = Loan status (Approved/Declined)
- dtir1 = debt-to-income ratio

Objective of the Project:

The objective of the dataset is based on the customer's financial and demographic data, loan prediction aims to determine whether a loan application should be approved or rejected. Machine learning algorithms are frequently used for this, which analyse historical loan data to spot trends and forecast how future loans will turn out.

Loan prediction's major objective is to assist financial institutions in making more knowledgeable loan approval decisions, while also lowering the chance of default and raising profitability. Financial firms may determine which consumers are most likely to repay their loans and which ones are high-risk borrowers by using machine learning to analyse vast volumes of data.

The most important variables that determine whether a loan will be approved or rejected can be found using loan prediction models. For instance, a loan prediction model might find that borrowers are more likely to be authorised for loans if they have higher credit ratings, stable employment histories, and lower debt-to-income ratios. Financial institutions can use this information to enhance their lending practises and create future loan approval decisions that are more precise.

Statistical Insights of the Dataset:

Importing libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from imblearn.under_sampling import RandomUnderSampler
from imblearn.over_sampling import SMOTE
```

Accessing the Dataset and reading it

```
df=pd.read_csv('Loan.csv')
df
```

	ID	year	loan_limit	Gender	approv_in_adv	loan_type	loan_purpose	Credit_Worthiness	open_credit	business_or_commercial	...	credit_type
0	24890	2019	cf	Sex Not Available	nopre	type1	p1	l1	nopc	nob/c	...	EXP
1	24891	2019	cf	Male	nopre	type2	p1	l1	nopc	b/c	...	EQUI
2	24892	2019	cf	Male	pre	type1	p1	l1	nopc	nob/c	...	EXP
3	24893	2019	cf	Male	nopre	type1	p4	l1	nopc	nob/c	...	EXP
4	24894	2019	cf	Joint	pre	type1	p1	l1	nopc	nob/c	...	CRIF
...	...	...	...	...	...	...	...	...	...	...	...	...

Number of Columns and the Column Info

```
In [4]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 148670 entries, 0 to 148669
Data columns (total 34 columns):
 #   Column                     Non-Null Count   Dtype
---  ------                     --------------   -----
 0   ID                         148670 non-null  int64
 1   year                       148670 non-null  int64
 2   loan_limit                 145326 non-null  object
 3   Gender                     148670 non-null  object
 4   approv_in_adv              147762 non-null  object
 5   loan_type                  148670 non-null  object
 6   loan_purpose               148536 non-null  object
 7   Credit_Worthiness          148670 non-null  object
 8   open_credit                148670 non-null  object
 9   business_or_commercial     148670 non-null  object
 10  loan_amount                148670 non-null  int64
 11  rate_of_interest           112231 non-null  float64
 12  Interest_rate_spread       112031 non-null  float64
 13  Upfront_charges            109028 non-null  float64
 14  term                       148629 non-null  float64
 15  Neg_ammortization          148549 non-null  object
 16  interest_only              148670 non-null  object
 17  lump_sum_payment           148670 non-null  object
 18  property_value             133572 non-null  float64
 19  construction_type          148670 non-null  object
 20  occupancy_type             148670 non-null  object
 21  Secured_by                 148670 non-null  object
 22  total_units                148670 non-null  object
 23  income                     139520 non-null  float64
 24  credit_type                148670 non-null  object
 25  Credit_Score               148670 non-null  int64
 26  co-applicant_credit_type   148670 non-null  object
 27  age                        148470 non-null  object
 28  submission_of_application  148470 non-null  object
 29  LTV                        133572 non-null  float64
 30  Region                     148670 non-null  object
 31  Security_Type              148670 non-null  object
 32  Status                     148670 non-null  int64
 33  dtir1                      124549 non-null  float64
dtypes: float64(8), int64(5), object(21)
memory usage: 38.6+ MB
```

We can see that the dataset contains 34 columns and more than 1 lakh rows.

Statistical information of all the features

In [94]:
```
df.describe()
```

Out[94]:

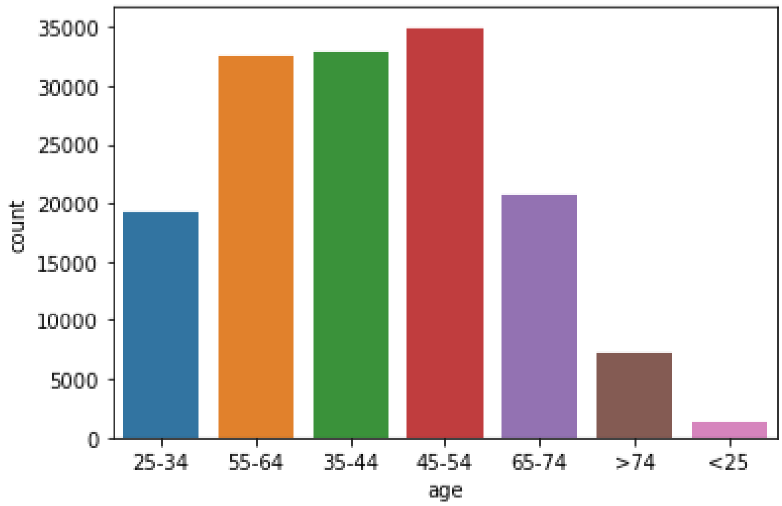
	ID	year	loan_amount	rate_of_interest	Interest_rate_spread	Upfront_charges	term	property_value	income	Credit_Sc
count	148670.000000	148670.0	1.486700e+05	112231.000000	112031.000000	109028.000000	148629.000000	1.335720e+05	139520.000000	148670.000
mean	99224.500000	2019.0	3.311177e+05	4.045476	0.441656	3224.996127	335.136582	4.978935e+05	6957.338876	699.789
std	42917.476598	0.0	1.839093e+05	0.561391	0.513043	3251.121510	58.409084	3.599353e+05	6496.586382	115.875
min	24890.000000	2019.0	1.650000e+04	0.000000	-3.638000	0.000000	96.000000	8.000000e+03	0.000000	500.000
25%	62057.250000	2019.0	1.965000e+05	3.625000	0.076000	581.490000	360.000000	2.680000e+05	3720.000000	599.000
50%	99224.500000	2019.0	2.965000e+05	3.990000	0.390400	2596.450000	360.000000	4.180000e+05	5760.000000	699.000
75%	136391.750000	2019.0	4.365000e+05	4.375000	0.775400	4812.500000	360.000000	6.280000e+05	8520.000000	800.000
max	173559.000000	2019.0	3.576500e+06	8.000000	3.357000	60000.000000	360.000000	1.650800e+07	578580.000000	900.000

Graphs:

Histogram

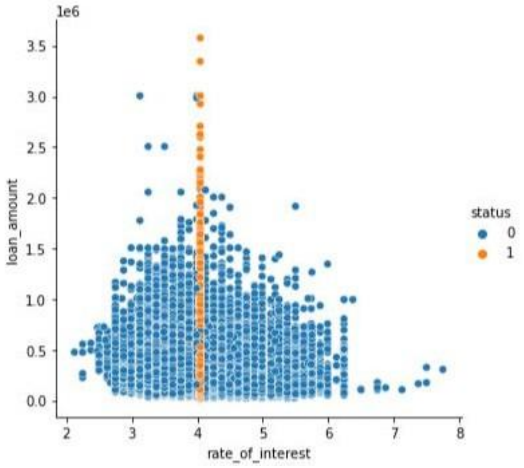
```
sns.countplot(data=df , x='age')
```

<AxesSubplot:xlabel='age', ylabel='count'>



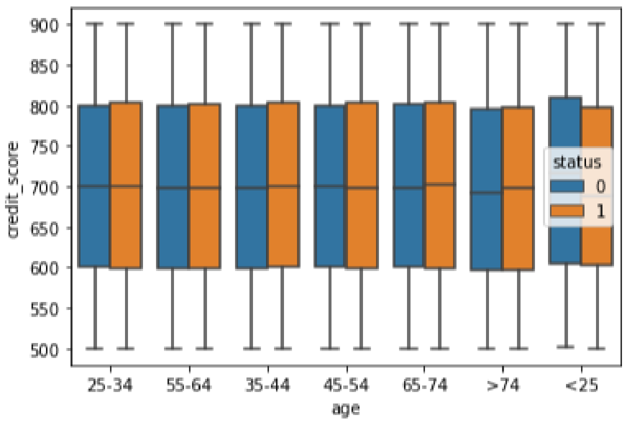
Relative Plot

```
In [60]: sns.relplot(x ="rate_of_interest", y ="loan_amount", hue='status' ,data = df)
Out[60]: <seaborn.axisgrid.FacetGrid at 0x211df41a130>
```



Box Plot

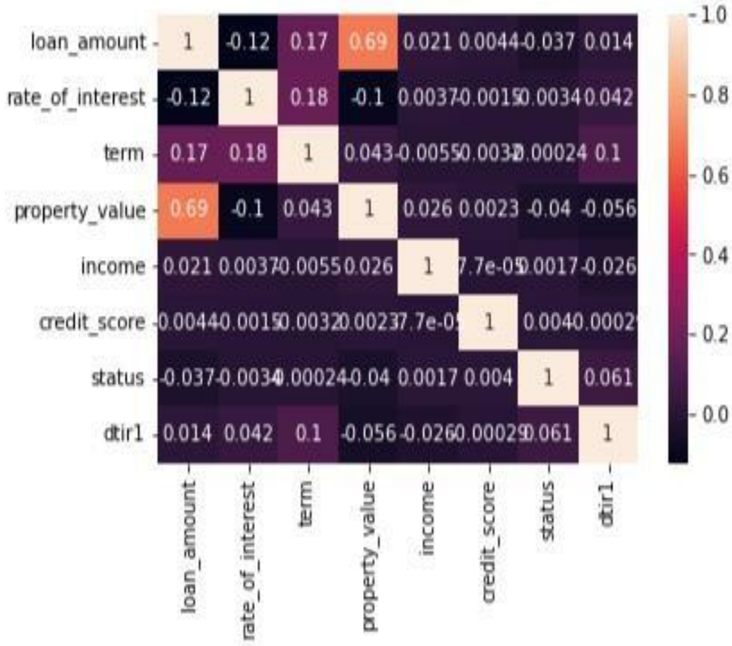
```
In [67]: sns.boxplot(data=df , x='age' , y ='credit_score' , hue='status')
Out[67]: <AxesSubplot:xlabel='age', ylabel='credit_score'>
```



Correlation Matrix of all the features

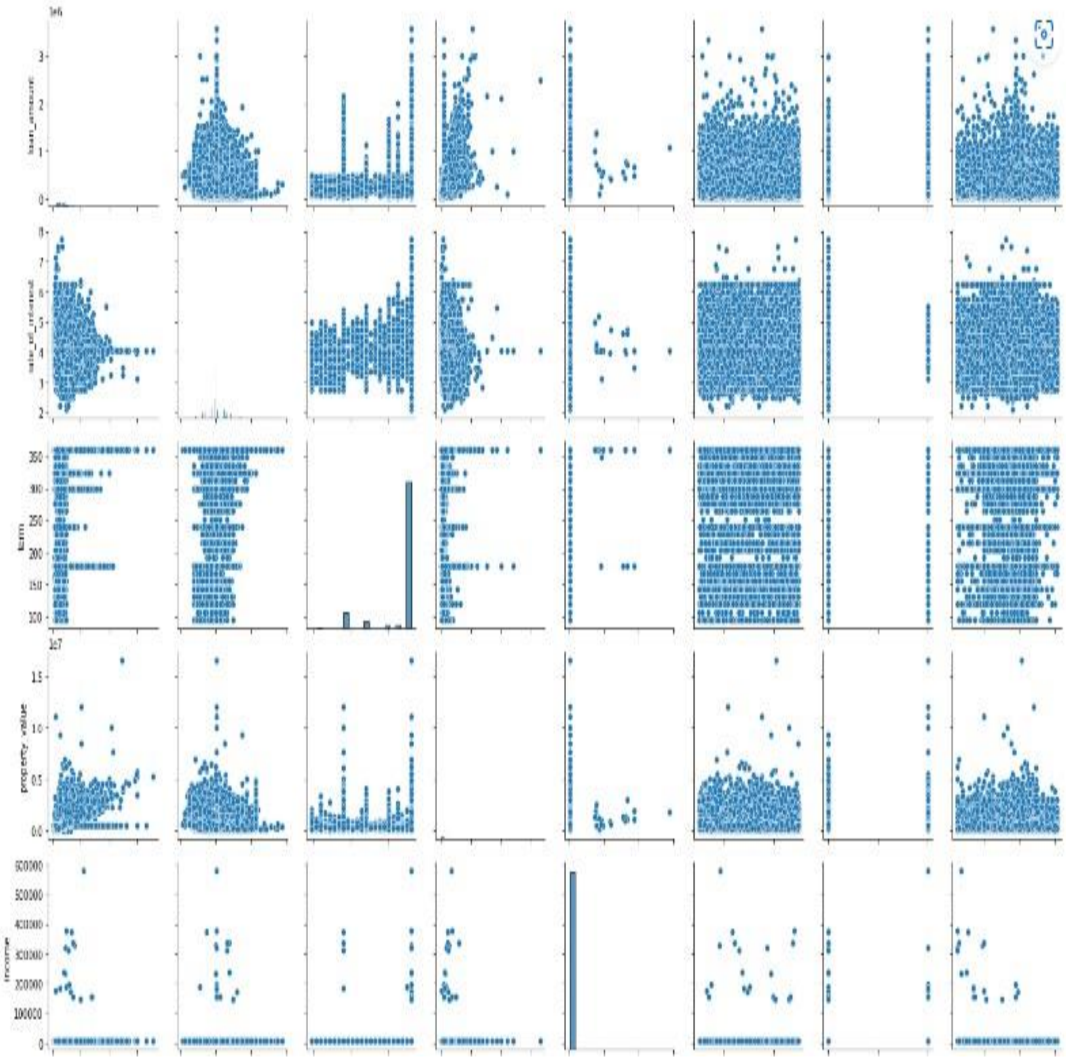
age

```
8]: sns.heatmap(df.corr() , annot = True)

8]: <AxesSubplot:>
```



Pair Plot

```
In [63]: sns.pairplot(df)

Out[63]: <seaborn.axisgrid.PairGrid at 0x211e3df6490>
```



LOVELY PROFESSIONAL UNIVERSITY
Transforming Education Transforming India

Model Building:

Random Forest Classification Algorithm

```
]: from sklearn.ensemble import RandomForestClassifier
   from sklearn.metrics import classification_report

   model=RandomForestClassifier()
   model.fit(x_train,y_train)
   y_pred=model.predict(x_test)

   conf = confusion_matrix(y_test,y_pred)
   acc= accuracy_score(y_test,y_pred)

   print('Accuracy of RandomForest: ',acc)
   print(10*'==========')
   print('Confusion Matrix: \n',conf)
   print(10*'==========')
   print('Classification Report: \n',classification_report(y_test,y_pred))
```

```
Accuracy of RandomForest:  0.977152466367713
====================================================================================================
Confusion Matrix:
 [[32660   857]
 [  162 10921]]
====================================================================================================
Classification Report:
               precision    recall  f1-score   support

           0       1.00      0.97      0.98     33517
           1       0.93      0.99      0.96     11083

    accuracy                           0.98     44600
   macro avg       0.96      0.98      0.97     44600
weighted avg       0.98      0.98      0.98     44600
```

Decision Tree Classification Algorithm

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score,confusion_matrix,ConfusionMatrixDisplay

model=DecisionTreeClassifier()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)

conf = confusion_matrix(y_test,y_pred)
acc= accuracy_score(y_test,y_pred)

print('Accuracy of DecisionTree: ',acc)
print(10*'===========')
print('Confusion Matrix: \n',conf)
print(10*'===========')
print('Classification Report: \n',classification_report(y_test,y_pred))
```

```
Accuracy of DecisionTree:  0.968542600896861
==============================================================================================================
Confusion Matrix:
 [[32758   759]
 [  644 10439]]
==============================================================================================================
Classification Report:
               precision    recall  f1-score   support

           0       0.98      0.98      0.98     33517
           1       0.93      0.94      0.94     11083

    accuracy                           0.97     44600
   macro avg       0.96      0.96      0.96     44600
weighted avg       0.97      0.97      0.97     44600
```

Logistic Regression Algorithm

```
: from sklearn.linear_model import LogisticRegression

model=LogisticRegression()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)

conf = confusion_matrix(y_test,y_pred)
acc= accuracy_score(y_test,y_pred)

print('Accuracy of Logistic Regression: ',acc)
print(10*'==========')
print('Confusion Matrix: \n',conf)
print(10*'==========')
print('Classification Report: \n',classification_report(y_test,y_pred))
```

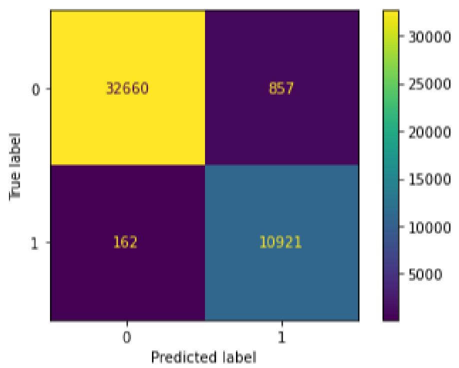
```
Accuracy of Logistic Regression:  0.48838565022421526
====================================================================================================
Confusion Matrix:
 [[15779 17738]
 [ 5080  6003]]
====================================================================================================
Classification Report:
               precision    recall  f1-score   support

           0       0.76      0.47      0.58     33517
           1       0.25      0.54      0.34     11083

    accuracy                           0.49     44600
   macro avg       0.50      0.51      0.46     44600
weighted avg       0.63      0.49      0.52     44600
```

Model Evaluation

Confusion Matrices of Logistic Regression and Random Forest Classification.

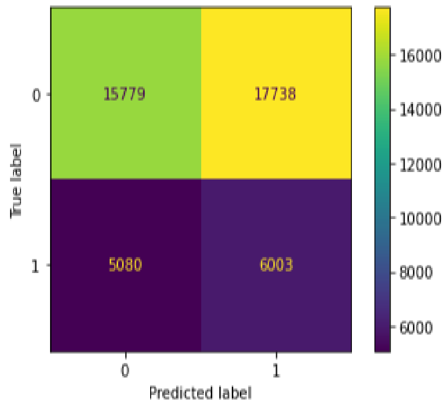
```
: # Confusion matrix of RandomForest
confdisplay=ConfusionMatrixDisplay(conf)
confdisplay.plot()
```

```
: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x211edce61f0>
```



```
confdisplay=ConfusionMatrixDisplay(conf)
confdisplay.plot()
```

<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x211de27c460>



Conclusion

As we conclude, the Decision Tree and Random Forest Classification algorithms have outperformed other Machine Learning Algorithms.The accuracy of the Decision Tree Classification Algorithm has come out to be 96.8542% The accuracy of the Logistic Regression Algorithm has come out to be 48.8385 % The accuracy of the Random ForestClassification Algorithm has come out to be 97.7152%

In this classification of Loan Prediction Analysis, we found that Random Forest Classification Algorithm has outperformed all the other classification algorithms with the highest accuracy.

References:

Sci Kit Learn References: https://scikit-learn.org/stable/

Kaggle: https://www.kaggle.com/code/manarzaitoon/simple-loan-default-prediction/input

Checklist for Dissertation Supervisor

Name:____________________________UID: ___________________________Domain:_____________

Registration No:_____________Name of student: __________________

Title of Dissertation:

____

- ☐ **Front pages are as per the format.**
- ☐ **The topic on the PAC form and title page are the same.**
- ☐ **Front page numbers are in roman and for report, it is like 1, 2, 3…….**
- ☐ **TOC, List of Figures, etc. match the actual page numbers in the report.**
- ☐ **Font, Font Size, Margins, line Spacing, Alignment, etc. are as per the guidelines.**
- ☐ **Color prints are used for images and implementation snapshots.**
- ☐ **Captions and citations are provided for all the figures, tables etc. and are numbered andcenter aligned.**
- ☐ **All the equations used in the report are numbered.**
- ☐ **Citations are provided for all the references.**
- ☐ **Objectives are clearly defined.**
- ☐ **The minimum number of pages of the report is 50.**
- ☐ **Minimum references in the report are 30.**

Here by, I declare that I had verified the above-mentioned points in the finaldissertation **report.**

Signature of Supervisor with UID