LOW LEVEL DESIGN DOCUMENT

BANK CREDIT RISK PREDICTION

DOCUMENT VERSION CONTROL

DATE ISSUED	VERSION	DESCRIPTION	AUTHOR
	V1.0	LLD-V1.0	ADHIRAJ SINGH SHEKHAWAT

1.0 Introduction

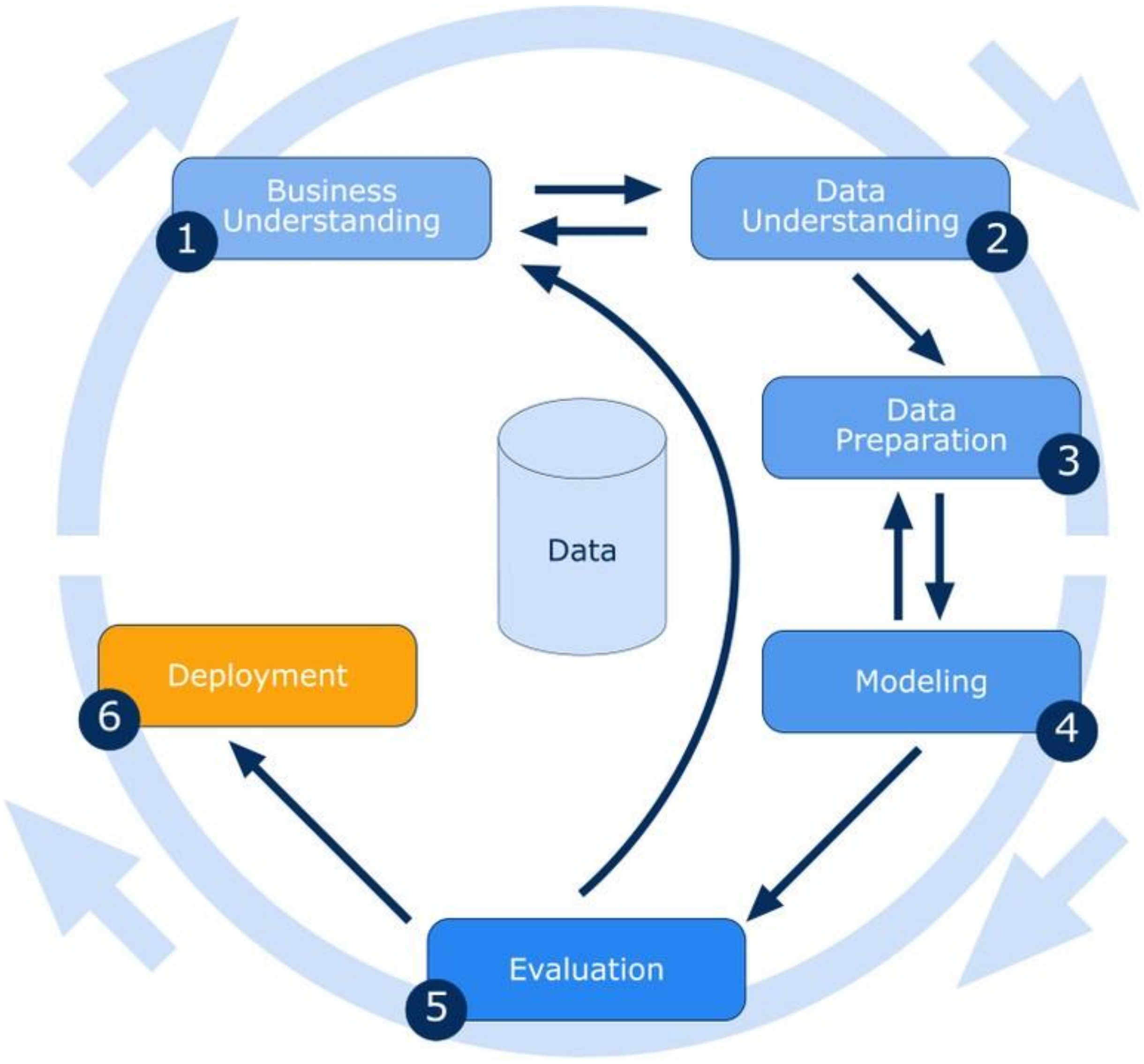
1. 1 What is Low-Level Design Document?

The goal of LLD or Low-Level design document (LLDD) is to give the internal logical design of the actual program code. Low-Level design is created based on the High-Level design. LLD describes the class diagrams with the methods and relations between classes and programspecs. It describes the modules so that the programmer can directly candirectly code the program from the document.

1.2 Scope

Low-level design (LLD) is a component-level design process that followsa step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code, and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work.

2.0 Architecture

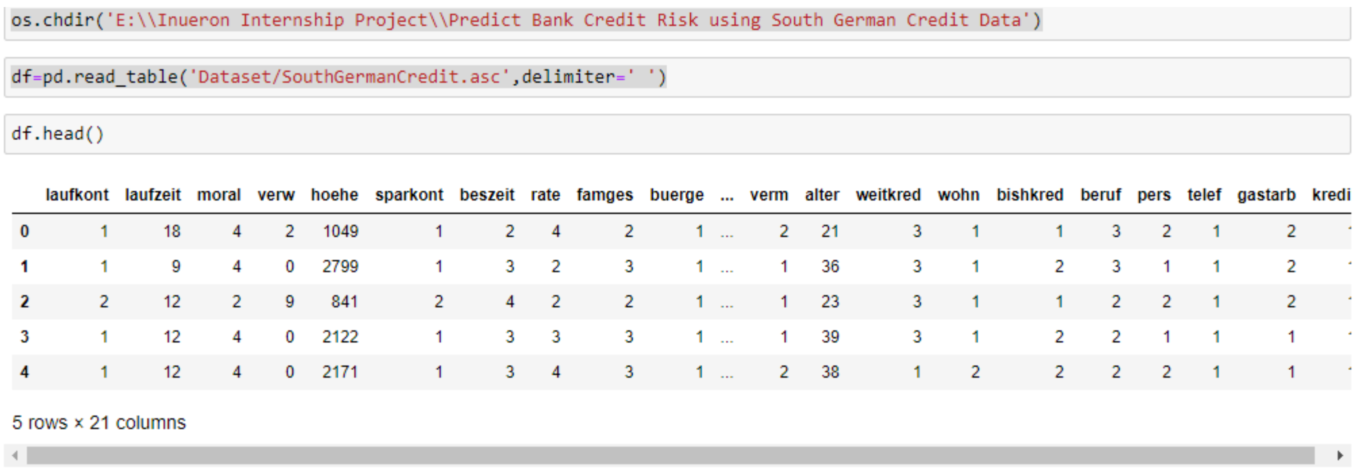


3. Architecture Description

The architecture for predicting credit risk of a bank project using machine learning involves several stages:

3.1 Data Description

The primary source of data for this project is taken UCI repository. The data contains of 1000 records and 21 attributes. The data is in structured format and stored in asc file



```
os.chdir('E:\\Inueron Internship Project\\Predict Bank Credit Risk using South German Credit Data')
```

```
df=pd.read_table('Dataset/SouthGermanCredit.asc',delimiter=' ')
```

```
df.head()
```

```
   laufkont  laufzeit  moral  verw  hoehe  sparkont  beszeit  rate  famges  buerge  ...  verm  alter  weitkred  wohn  bishkred  beruf  pers  telef  gastarb  kredi
0         1        18      4     2   1049         1        2     4       2       1  ...     2     21         3     1         1      3     2      1        2
1         1         9      4     0   2799         1        3     2       3       1  ...     1     36         3     1         2      3     1      1        2
2         2        12      2     9    841         2        4     2       2       1  ...     1     23         3     1         1      2     2      1        2
3         1        12      4     0   2122         1        3     3       3       1  ...     1     39         3     1         2      2     1      1        1
4         1        12      4     0   2171         1        3     4       3       1  ...     2     38         1     2         2      2     2      1        1

5 rows × 21 columns
```

3.2. Preprocessing

In this stage data is prepared for analysis. This includes cleaning the data, removing duplicates, handling missing values, and transforming the data into a suitable format for machine learning models. In the input South German bank data there were no duplicate values or null values which need to handled.

3.3. Feature Engineering

In this stage, relevant features are selected and engineered to improve the performance of the machine learning model. This can involve techniques such as feature selection, dimensionality reduction, and feature scaling. Some of the features were showing skewness and even after removing the skewness using log transformation 4 columns were dropped. The outliers were also removed by assigning upper limit values to outliers above the limit and similarly lower limit values to outliers below the lower limit.

3.4. Model Selection and Training

In this stage, a suitable machine learning model is selected based on the problem statement and data characteristics. The selected model is then trained on the preprocessed and engineered data to learn the patterns and relationships between the input and output variables. In this problem Random forest, Xgboost, Gradient Boosting, Adaboost and Bagging Classifier algorithms were hyper tuned to give the maximum accuracy.

3.5. Model Evaluation

The performance of the trained model is evaluated using various metrics such as accuracy, precision, recall, F1 score, and AUC-ROC curve. This helps to determine the effectiveness of the model and identify areas for improvement.

3.6. Deployment and Monitoring:

The final stage involves deploying the model into a production environment and continuously monitoring its performance. This includes tracking the model's accuracy and detecting any deviations from the expected behavior.

4.0 Unit Test Cases

Test Case Description	Pre - Requisites	Expected Results
Verify whether the Webpage is accessible to the User or not.	Webpage URL should be defined.	Webpage should be accessible to the User.
Verify whether the webpage is completely loads for the User or not.	1. Webpage URL is accessible. 2. Webpage is deployed.	The Webpage should be completely loads for the User when it is accessed.
Verify whether the user is able to enter data in input fields or not.	1. Webpage URL is accessible. 2. Webpage is deployed. 3. Webpage input fields are editable.	The User is able to enter data in input fields.
Verify whether the user is able to submit details or not.	1. Webpage URL is accessible. 2. Webpage is deployed. 3. Webpage input fields are editable.	The User is able to submit details to process.
Verify whether the user gets recommended Results on submitting the details or not.	1. Webpage URL is accessible. 2. Webpage is deployed. 3. Webpage input fields are editable.	The User gets recommended results on submitting the details.