Low Level Design
Thyroid Disease Prediction

Revision Number: 1.0 Last
date of revision: 01/05/2024

Dhaksin S

Document Version Control

Date Issued	Version	Description	Author
01 May 2024	1.0	First Version of Complete LLD	Dhaksin S

iNeuron

Contents

1. Introduction

1.1 What is Low-Level design document?

The goal of the LDD or Low-level design document (LLDD) is to give the internal logic design of the actual program code for the House Price Prediction dashboard. LDD describes the class diagrams with the methods and relations between classes and programs specs. It describes the modules so that the programmer can directly code the program from the document.

1.2 Scope

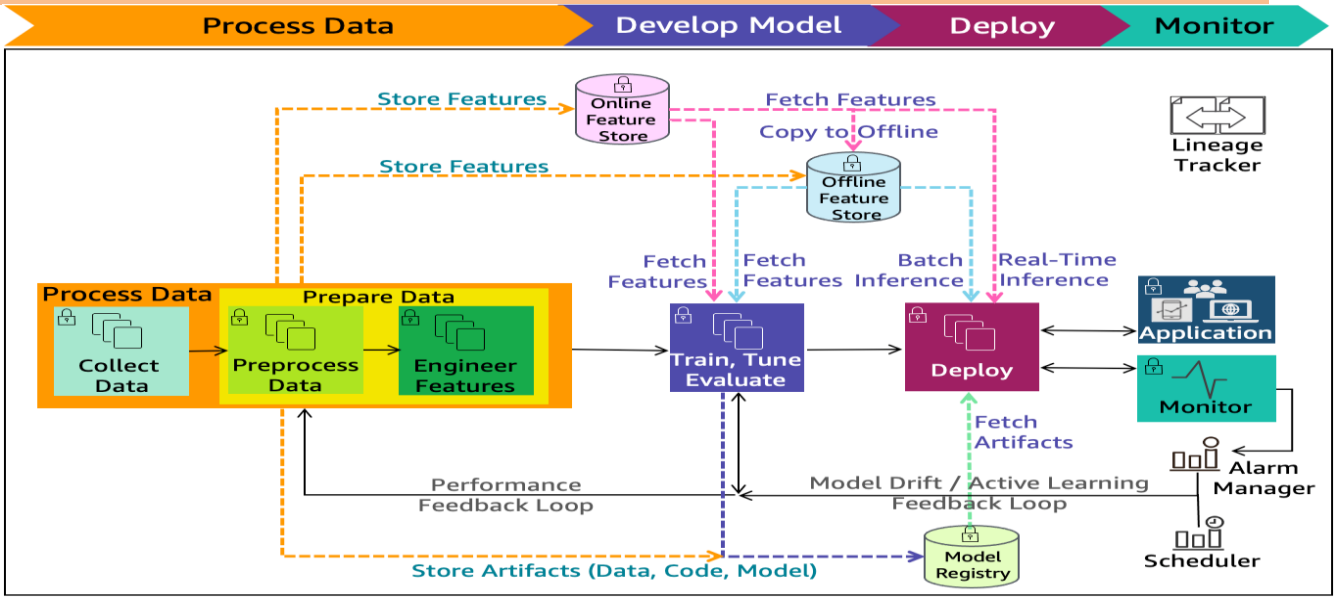
Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. The process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement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

2. Architecture

The architecture for predicting thyroid disease is designed to handle data ingestion, cleaning, exploratory data analysis (EDA), preprocessing, model building, and evaluation.

It starts with loading the dataset and cleaning it by removing unnecessary columns and handling missing values. EDA is then performed to understand the dataset, followed by preprocessing steps such as converting categorical data to numerical format, imputing missing values, and scaling features.

The dataset is split into training and testing sets for model building, where various machine learning algorithms are trained and evaluated. The best-performing model is selected based on accuracy scores for predicting thyroid disease. This model can be deployed as a tool for healthcare professionals to aid in early detection and management of thyroid disorders.



1. Data Ingestion

For data ingestion, the dataset contains the following columns:

1. **Age:** The age of the patient.
2. **Gender:** The gender of the patient (male or female).
3. **On Thyroxine:** Whether the patient is on thyroxine medication (yes or no).
4. **Query on Thyroxine:** Whether there is a query about thyroxine medication (yes or no).
5. **On Antithyroid Medication:** Whether the patient is on antithyroid medication (yes or no).
6. **Sick:** Whether the patient is sick (yes or no).
7. **Pregnant:** Whether the patient is pregnant (yes or no).
8. **Thyroid Surgery:** Whether the patient has undergone thyroid surgery (yes or no).
9. **I131 Treatment:** Whether the patient has undergone I131 treatment (yes or no).
10. **Query Hypothyroid:** Whether there is a query about hypothyroidism (yes or no).
11. **Query Hyperthyroid:** Whether there is a query about hyperthyroidism (yes or no).
12. **Lithium:** Whether the patient is taking lithium medication (yes or no).
13. **Goitre:** Whether the patient has goitre (yes or no).
14. **Tumor:** Whether the patient has a tumor (yes or no).
15. **Hypopituitary:** Whether the patient has hypopituitarism (yes or no).
16. **Psych:** Whether the patient has a psychiatric condition (yes or no).
17. **TSH Measured:** Whether TSH levels have been measured (yes or no).
18. **TSH:** Thyroid-stimulating hormone (TSH) level.
19. **T3 Measured:** Whether T3 levels have been measured (yes or no).
20. **T3:** Triiodothyronine (T3) level.
21. **TT4 Measured:** Whether TT4 levels have been measured (yes or no).
22. **TT4:** Total thyroxine (TT4) level.
23. **T4U Measured:** Whether T4U levels have been measured (yes or no).
24. **T4U:** Thyroxine-binding globulin (T4U) level.
25. **FTI Measured:** Whether FTI levels have been measured (yes or no).

26. **FTI:** Free thyroxine index (FTI) level.
27. **TBG Measured:** Whether TBG levels have been measured (yes or no).
28. **TBG:** Thyroxine-binding globulin (TBG) level.
29. **Target:** The diagnosis or target variable indicating the thyroid disease condition.

When ingesting this dataset, it is important to ensure that each column is correctly parsed and handled according to its data type (e.g., categorical, numerical). Missing values and any inconsistencies should be addressed during the ingestion process to ensure data quality for subsequent analysis.

2. Data Cleaning

During data cleaning, the dataset is processed to remove unnecessary columns and handle missing values. Columns like "TBG Measured" and "TBG" are removed due to many null values. Missing values are imputed using the KNNImputer, ensuring the dataset is clean and ready for analysis.

3. Exploratory Data Analysis

Exploratory Data Analysis (EDA) is crucial for understanding the dataset and identifying patterns or trends that can inform the modeling process. In the context of the thyroid disease dataset, EDA involves visualizing the distribution of variables such as age, gender, and various thyroid-related measurements. This helps in identifying any outliers or anomalies in the data. Additionally, EDA can reveal relationships between variables, such as the correlation between different thyroid hormones.

Visualizations such as histograms, box plots, and scatter plots are used to explore the data. For example, a histogram of age can show the age distribution of patients in the dataset. A box plot can reveal the distribution of TSH levels among different genders. These visualizations provide insights into the data, helping to inform preprocessing steps and model selection for predicting thyroid disease.

4. Preprocessing

Data preprocessing is essential to ensure the dataset is ready for analysis and model building. In the context of predicting thyroid disease, preprocessing involves handling missing values, converting categorical variables to numerical format, and scaling features. Missing values are imputed using the KNNImputer to maintain the integrity of the dataset. Categorical variables like gender and medication status are encoded using LabelEncoder to convert them into numerical values. Feature scaling is applied using MinMaxScaler to normalize the features to a common scale, ensuring each feature contributes equally to the model. These preprocessing steps are crucial for building accurate predictive models for thyroid disease.

5. Model Building

Model building for predicting thyroid disease involves training several machine learning algorithms on the preprocessed dataset. Algorithms such as Decision Trees, Random Forests, K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Logistic Regression are trained and evaluated using metrics like accuracy, precision, recall, and F1-score. The model with the highest accuracy is selected

as the final model for predicting thyroid disease. Hyperparameter tuning is also performed to optimize the model's performance. The selected model can then be deployed as a predictive tool for healthcare professionals to aid in the early detection and management of thyroid disorders.

6. Model Evaluation

Model evaluation involves assessing the performance of trained models using metrics like accuracy, precision, recall, and F1-score. The model with the highest accuracy is selected for predicting thyroid disease.

7. Model Deployment

The selected model can be deployed as a predictive tool for healthcare professionals to aid in diagnosing thyroid disease.

3. ARCHITECTURE DESCRIPTION

3.1 Data Description

The dataset for predicting thyroid disease contains a comprehensive set of clinical and laboratory variables that are crucial for diagnosing and managing thyroid disorders. These variables provide valuable insights into the patients' health status and help healthcare professionals make informed decisions regarding their treatment.

The dataset includes demographic information such as age and gender, which are important factors in assessing the risk of thyroid disease. Age is a significant predictor, as thyroid disorders are more common in older individuals. Gender also plays a role, as thyroid disease is more prevalent in females.

Medication status variables like "On Thyroxine," "Query on Thyroxine," and "On Antithyroid Medication" indicate the patients' current medication regimen, which is essential for understanding their treatment history and managing their condition effectively.

Variables such as "Sick," "Pregnant," and "Thyroid Surgery" provide information about the patients' overall health status and any specific conditions that may impact their thyroid function. For example, pregnancy can affect thyroid hormone levels, requiring adjustments in medication.

The dataset also includes diagnostic query variables like "Query Hypothyroid" and "Query Hyperthyroid," which indicate whether there is a suspicion or concern about these conditions. These variables help in identifying patients who require further evaluation and monitoring.

Laboratory measurements such as "TSH," "T3," "TT4," "T4U," "FTI," and "TBG" are crucial for diagnosing and monitoring thyroid disorders. Thyroid-stimulating hormone (TSH), triiodothyronine (T3), and thyroxine (TT4) levels are key indicators of thyroid function, while thyroxine-binding globulin (TBG) levels can affect hormone transport. Free thyroxine index (FTI) and thyroxine-binding globulin (T4U) levels provide additional insights into thyroid function.

Variables like "Goitre," "Tumor," "Hypopituitary," and "Psych" indicate the presence of other conditions or factors that may influence thyroid function. These variables help in identifying patients with complex medical histories who may require specialized care.

Overall, this dataset provides a comprehensive view of patients' health status and thyroid function, making it valuable for predicting thyroid disease and guiding treatment decisions. Proper analysis and modeling of this dataset can help in developing accurate predictive models and improving the management of thyroid disorders.

3.2 Data Loading

Step 1:- file = open("dataset/thyroid.csv") df = pd.read_csv(file)

Step 2:- df

Out[7]:

	age	gender	onthyroxine	query_on_thyroxine	on_antithyroid_medication	sick	pregnant	thyroid_surgery	I131_treatment	query_hypothyroid	...	T3	TT4
0	29	F	f	f	f	f	f	f	f	f	...	1.9	
1	41	F	f	f	f	f	f	f	f	f	...	?	
2	36	F	f	f	f	f	f	f	f	f	...	?	
3	32	F	f	f	f	f	f	f	f	f	...	?	
4	60	F	f	f	f	f	f	f	f	f	...	?	
...	...	...	...	...	...	...	...	...	...	...	...	...	...
9166	56	M	f	f	f	f	f	f	f	f	...	?	
9167	22	M	f	f	f	f	f	f	f	f	...	?	
9168	69	M	f	f	f	f	f	f	f	f	...	?	
9169	47	F	f	f	f	f	f	f	f	f	...	?	
9170	31	M	f	f	f	f	f	f	f	t	...	?	

9171 rows × 29 columns

```
  Letter        Diagnosis
    ------      ---------

hyperthyroid conditions:

    A   hyperthyroid
    B   T3 toxic
    C   toxic goitre
    D   secondary toxic

hypothyroid conditions:

    E   hypothyroid
    F   primary hypothyroid
    G   compensated hypothyroid
    H   secondary hypothyroid

binding protein:

    I   increased binding protein
    J   decreased binding protein

general health:

    K   concurrent non-thyroidal illness

replacement therapy:
```

```
    L   consistent with replacement therapy
    M   underreplaced
    N   overreplaced

antithyroid treatment:

    O   antithyroid drugs
    P   I131 treatment
    Q   surgery

miscellaneous:

    R   discordant assay results
    S   elevated TBG
    T   elevated thyroid hormones


hyperthyroid conditions (A, B, C, D)
hypothyroid conditions (E, F, G, H)
binding protein (I, J)
general health (K)
replacement therapy (L, M, N)
discordant results (R)
```

.

3.3 Data Transformation

Data transformation in this dataset involves converting categorical variables into numerical format using LabelEncoder. This transformation is necessary for machine learning algorithms to process the data effectively. Additionally, missing values are imputed using the KNNImputer to ensure the dataset is complete. Feature scaling is applied using MinMaxScaler to normalize the features to a common scale, preventing any bias in the model. These transformations prepare the dataset for model building and evaluation, ensuring that the algorithms can accurately predict thyroid disease based on the available data.

3.3 Data Modelling

Data Modelled and the accuracy obtained.

Decisiontree

```
]:  1  from sklearn.tree import DecisionTreeClassifier
    2  tree = DecisionTreeClassifier(max_depth=3)
    3  clf = tree.fit(X_train,y_train)
    4  treepredict = clf.predict(X_test)
```

```
]:  1  accuracy_score(treepredict,y_test)
```

```
]:  0.7975077881619937
```

Random Forest

```
]:  1  from sklearn.ensemble import RandomForestClassifier
    2  rf = RandomForestClassifier(max_depth=2,n_estimators=200)
    3  rclf = rf.fit(X_train,y_train)
    4  rfpred = rclf.predict(X_test)
    5  accuracy_score(rfpred,y_test)
```

```
]:  0.742601246105919
```

3.4 Evaluation:

Model evaluation is a crucial step in assessing the performance of machine learning models trained on the dataset for predicting thyroid disease. Several metrics are used to evaluate the models, including accuracy, precision, recall, and F1-score.

Accuracy measures the proportion of correctly classified instances out of the total instances. Precision measures the proportion of correctly predicted positive instances out of all predicted positive instances, indicating the model's ability to avoid false positives. Recall measures the proportion of correctly predicted positive instances out of all actual positive instances, indicating the model's ability to identify all relevant cases.

F1-score is the harmonic mean of precision and recall, providing a balance between the two metrics. It is useful when there is an imbalance between the classes in the dataset.

Additionally, the models can be evaluated using a confusion matrix, which provides a detailed breakdown of true positives, true negatives, false positives, and false negatives. This matrix helps in understanding the performance of the model across different classes and identifying any specific areas for improvement. Overall, model evaluation ensures that the selected model is accurate and reliable for predicting thyroid disease based on the dataset.

Decision Tree

```
]:  1 from sklearn.tree import DecisionTreeClassifier
    2 tree = DecisionTreeClassifier(max_depth=3)
    3 clf = tree.fit(X_train,y_train)
    4 y_pred = clf.predict(X_test)
    5 accuracy_score(y_pred,y_test)
```

```
]: 0.8430685358255452
```

4. Accuracy

Model	Accuracy
Decision Tree	0.8430685358255452

Random Forest Classifier	0.7387071651090342
KNN Classifier	0.7387071651090342
SVM	0.7422118380062306
Decision Tree with the highest accuracy here	