iNeuron

Drug Activity Prediction

LOW LEVEL DESIGN DOCUMENT (LLD)

Project By: -

Ashfaque Jamal

Beauty Atara

Date: - 22/09/2022

Table of Content:

Introduction:

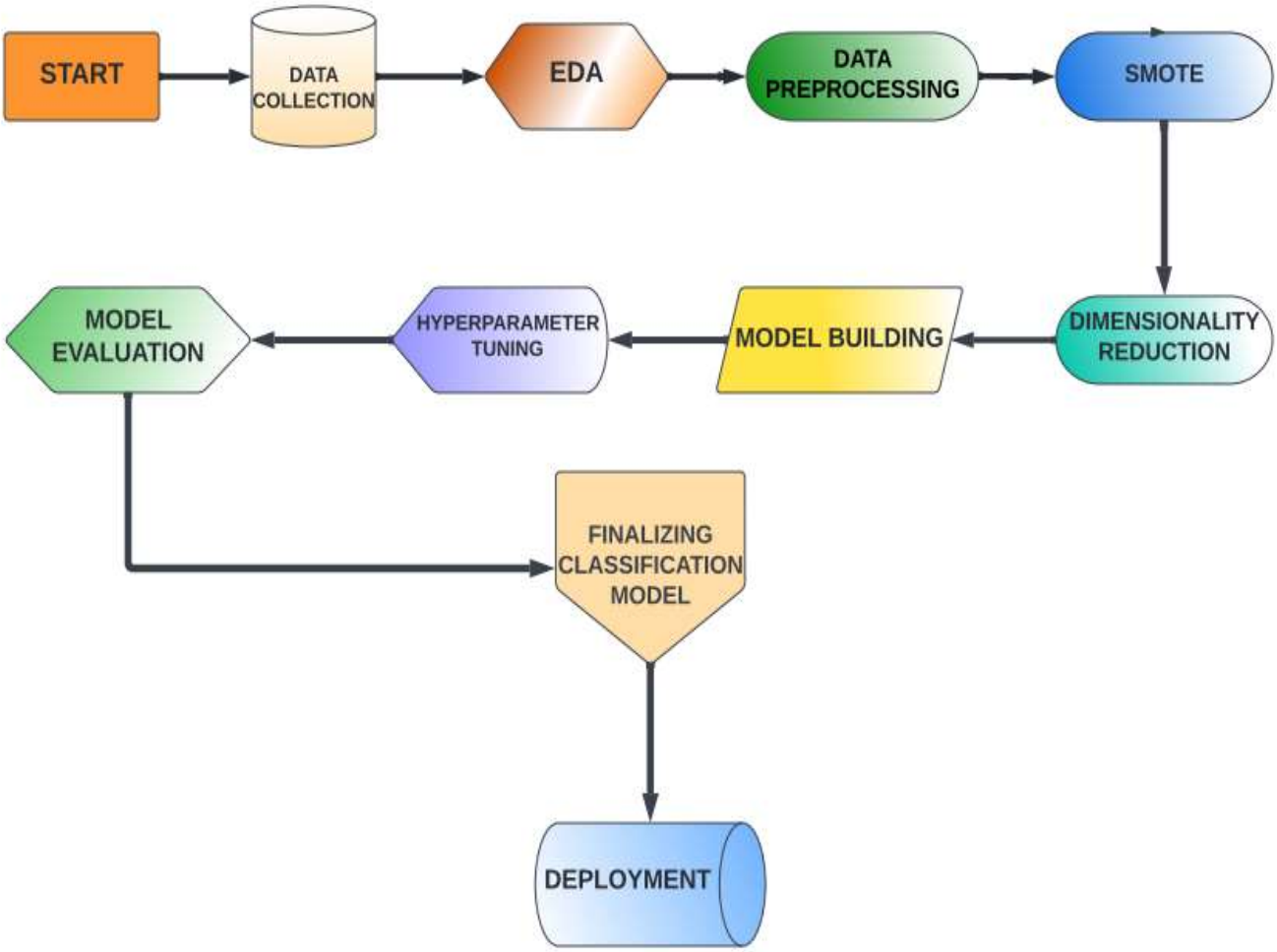
1.1 What is LLD?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for **Drug Activity Prediction**. LLD describes the class diagrams with the methods and relations between classes and program 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. 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.

Architecture:



Architecture Description:

1. **Data Collection:**

   The collected data was in two parts, training data and testing data. The data was in .dat format, so first of all we converted it into .csv format. The training data was containing 800 rows and the testing data was containing 350 rows. The data was then expanded to a dimension of 800*1000000 and 350*1000000 for training and testing data respectively.

2. **Reading the data:**

   The data was read with the help of panda's library.

3. **EDA:**

   In Exploratory Data Analysis, we checked whether the data is having any null values or not. After that we also checked for any outliers present in the data.

4. **Data Pre-processing:**

   The first step in data pre-processing was to deal with the null values. Our data contained many null values, so we kept only those columns for which the number of null values is less than 15 % of the total values in column. Then after, the null values were replaced with the median of the specific feature. After dealing with null values, the next step was to scale the data. The data was then standardised using standard scaler. The last step in data pre-processing was applying One Hot Encoding in order to expand the data and convert it into sparse matrix.

5. **Resampling:**

   The dataset provided contains 800 drug activities out of which, there were only 78 actives (+1) and 722 inactive (0). This distribution is referred to as Class-Imbalance. To deal with the class imbalance, SMOTE (Over Sampling technique) was applied to increase the samples of the minority-class to have equal distribution of minority vs majority so that there is no bias between the classes while classification.

6. **Dimensionality Reduction:**

   As the data uses was high-dimensional in nature, we tried to reduce the dimensionality of the data using the algorithm of PCA to facilitate faster development

7. **Model Building:**

   The data was now ready to be fed to the classification model. The following classification models were used to make the predictions:

   - Logistic Regression
   - Decision Tree Classifier
   - Random Forest Classifier
   - Perceptron
   - Naive Bayes Classifier
   - K Nearest Neighbours
   - Support Vector Classifier

8. **Hyperparameter tuning:**

Using GridSearchCV the hyperparameters for each of the model were tuned, and those parameters for which the best results were obtained, were used for final model building.

9. **Model Evaluation:**

All the models were then evaluated on the test data. Since the data was imbalanced, the main focus was on the precision, recall and the f1 score for evaluating the model.

10. **Finalizing Model:**

After trying all the experiments with different combinations of all parameters, the best model was arrived at. The resulting model was then used to compute the activities of the drugs in test dataset into active or inactive classes.

Results:

1. With PCA=100 and with SMOTE:

	Model	f1_scores
1	DT	0.579235
3	NB	0.543087
2	RF	0.500632
6	SVC	0.493414
5	KNN	0.482183
0	LR	0.467331
4	Perceptron	0.436200

2. PCA=350 and with SMOTE:

	Model	f1_scores
3	NB	0.532086
6	SVC	0.494800
1	DT	0.493467
0	LR	0.484183
5	KNN	0.482183
4	Perceptron	0.478814
2	RF	0.474474

3.Without SMOTE, with PCA:

	Models	f1_scores
3	Gaussian NB	0.545383
0	Decision Tree Classifier	0.514865
6	Perceptron	0.513147
1	Random Forest Classifier	0.503759
5	KNN	0.500632
7	SVC	0.500632
4	Bernoullie NB	0.496212
2	Logistic Regression	0.474474

4.Without PCA and with SMOTE:

	Models	f1_scores
0	Decision Tree Classifier	0.549122
3	Gaussian NB	0.542942
4	Bernoullie NB	0.526137
2	Logistic Regression	0.524681
1	Random Forest Classifier	0.521933
5	KNN	0.495355
7	SVC	0.490715
6	Perceptron	0.389606

5. With One Hot Encoding, PCA=100, with SMOTE:

	Models	f1_scores
7	SVC	0.528741
6	Perceptron	0.516189
2	Logistic Regression	0.500632
1	Random Forest Classifier	0.490382
4	Bernoullie NB	0.412599
0	Decision Tree Classifier	0.396008
5	KNN	0.125423
3	Gaussian NB	0.088542

6. With One Hot Encoding, PCA=350, with SMOTE:

	Models	f1_scores
1	Random Forest Classifier	0.503759
2	Logistic Regression	0.500632
6	Perceptron	0.499125
7	SVC	0.481273
0	Decision Tree Classifier	0.463333
4	Bernoullie NB	0.417839
3	Gaussian NB	0.088542
5	KNN	0.088542

Conclusion:

Maximum f1 score was 0.579, obtained from Decision Tree Classifier with PCA (n_components=100) and with resampling. We also came to know that without resampling the models are not performing well. Decision Tree Classifier and Naive Bayes are performing good compared to other classification models.