**EDGE – Technical document**

**Module Name : XXXXXXXX [ Eg : Data balancing ]**

Table of Contents

[1.Introduction 1](#_Toc46768999)

[2.Purpose of the document 2](#_Toc46769000)

[3.Abbrevations and definitions 2](#_Toc46769001)

[4.Module work flow 2](#_Toc46769002)

[5. technical summary 3](#_Toc46769003)

[6.Handling checks 3](#_Toc46769004)

[7.Common and coding standards 4](#_Toc46769005)

[8.Class and functions skeleton 5](#_Toc46769006)

[9.Try catch blocks 6](#_Toc46769007)

[10.Performance 6](#_Toc46769008)

[11.GIT repository 7](#_Toc46769009)

[12.Affected modules 7](#_Toc46769010)

[13.Reference documents 7](#_Toc46769011)

# 1.Introduction

The document contains the detailed technical description of the EDGE module XXXXX [ Eg -Class balancing ] and its functional over view, integration with all other modules and details as needed.

# 2.Purpose of the document

The document explains a module’s functional work, how it is communicating with other modules and what should be the correct behaviour of the module to complete a flow . Document address the below sections in depth

1. Functional
2. Integration
3. Standard to be followed
4. Performance
5. Check points
6. Maintainability

# 3.Abbrevations and definitions

|  |  |
| --- | --- |
| Abbreviation | **Definition** |
| SVM | Solverminds Solutions and Technology Pvt Ltd |
| DB | Database |

# 4.Module work flow

Data from statistics / Visualization

NO YES

Up sampling

Down sampling

SMOTE

Cat target

Algorithms

NO Balancing

With/Without Feature selection

# 5. technical summary

1. Data comes from statistics page or Visualization
2. System is validating the target , if the target is categorical and not exceeding more than 10 classes or less than 1 class , balancing is enabled , if not balancing will not be enabled
3. If the target is numerical , balancing is disabled
4. Suggestion is displayed whether a balancing is required or Not
5. Based on user selection, balancing is applied to the training data set and model is developed
6. Once the best model is saved to predict the test data, same pipeline which includes balancing is applied to the new/test data in the prediction process

# 6.Handling checks

System is handling various scenarios as below but not limited to

1. Numerical column to balance
2. More class or only 1 class as target
3. Date column or Text column will not be given to balancing module [ Try catch – exception handling ]
4. Missing values is handled by mode [ Mean and mode for all other columns]
5. Complete row or column empty is handled during upload itself
6. Special characters and symbol in target is handled
7. Special strings is handled
8. Any introduction of error due to balancing is handled
9. When we have few minority class with the same count [very less % comparatively to majority class ] when unbalancing is applied, one random minority class will be over sampled
10. Point 9 is applicable for majority class – Down sampling

# 7.Common and coding standards

* Project file or module should have the below
  + Name of the module
  + Date of module creation
  + Author of the module
  + Modification history
  + Synopsis of the module about what the module does
  + Global variables and Constant names accessed by the module
* Variable name should be meaningful and understandable
* Local variables should be named using camel case lettering starting with small letter (e.g. localData)
* Global variables names should start with a capital letter (e.g. GlobalData).
* Constant names should be formed using capital letters only (e.g. CONSDATA).
* It is better to avoid the use of digits in variable names.
* Declare only necessary variables and make sure declared variables are used [ IDE has these alerts]
* Remove the assignment of variables before the return if it is used outside the function
* Proper and consistency in the naming convention. [Eg: underscore between the words]
* Name the functions according to what they do.
* Length of functions should not be very large
* Code should be understandable with comments [Do not comment unnecessarily].
* DO NOT PRINT UNNECESSARILY
* Use multiple nested loops only when absolutely necessary.
* Make sure indentation structure is correct.
* Write the functions separately and call inside the main action.
* Error return values and exception handling conventions [ try and catch blocks]
* Maintain appropriate organization of files and folder for each project.

# 8.Class and functions skeleton

Module is designed as below

Main class is balancing and appropriate functions are retuned to get the output.

Same class and functions can be called for the new dataset [ Test dataset ]

class DataFrameImputer(TransformerMixin):  
  
 def \_\_init\_\_(self):  
 *"""Impute missing values.  
  
 Columns of dtype object are imputed with the most frequent value  
 in column.  
  
 Columns of other types are imputed with mean of column.  
  
 """* def fit(self, X, y=None):  
 self.fill = pd.Series([X[c].value\_counts().index[0] if X[c].dtype == np.dtype('O') else X[c].mean() for c in X],  
 index=X.columns)  
  
 return self  
  
 def transform(self, X, y=None):  
 return X.fillna(self.fill)

class classbalancing:

def \_\_init\_\_(self, r=SMOTE, i=40):

self.type = SMOTE

self.perccentage = 40

def train\_data(self,sampling\_perc):

train\_test\_split(x,y,test size=\_, random\_seed=123)

# 9.Try catch blocks

Try catch block is added everywhere as to capture any exceptions

[Copy & paste the code and make the return statement as consistent ]

Return Keys - success , failure , validation

try:  
 finalmlmodel.objects.filter(filename=filename).delete()  
 except:  
 pass  
 connection.close()  
 deletelog(filename)  
 error = {"status": "failure", "acc": "", "finalalgo": "", "modelfile": "","error": str(e)}  
 return Response(error)  
except Exception as e:  
 traceback.print\_exc()  
 return Response({"status": "failure", "error": str(e)})

# 10.Performance

1. Module should is sending an estimated time to the user for the process to be completed
2. Unhandled scenarios as conversion into the list, tuples or array is handled system is not hanging in such scenario
3. An estimated usage of memory is calculated and returned as exception to the user that “memory limit is exceed”.

# 11.GIT repository

GIT repository of the module code placed

# 12.Affected modules

List of affected modules

1. Start learning - Training of algorithms takes into the account of balancing
2. Prediction – Pipeline will be applied to the test data and if there is a class balancing , same should be applied to the new data

# 13.Reference documents

Lit of reference documents

1.Previous module document

2.Affected module document

3.Other miscellaneous documents referred in this document