**Prediction Model for Students’ Future Development by Deep Learning and Tensorflow Artificial Intelligence Engine**

**Abstract**

Classification and prediction of students’ performance in examination are the typical challenges for educators. Various traditional data mining methods such as decision tree and association rules were used to perform classification. In recent years, the rapid development of artificial intelligence and deep learning algorithm provided another approach for intelligent classification and result prediction. In this paper, a research on how to use Tensorflow artificial intelligence engine for classifying students’ performance and forecasting their future universities degree program is studied. An appropriate and accurate forecast is important for providing prompt advice to student on program and university selection. For a more comprehensive consideration of an all rounded factors, the deep learning model analysed not only the traditional academic performance including Mathematic, Chinese, English, Physics, Chemistry, Biology and History, but also non-academic performance such as service, Conduct, Sport and Art. A few parameters in Tensorflow engine including the number of intermediate nodes and number of deep learning layers are adjusted and compared. With a data set of two thousands students, 75% of these data are used as the training data and 25% are used as the testing data, the accuracy ranged from 80% to 91%. The optimal configuration of the Tensorflow deep learning model that achieves highest prediction accuracy is determined. This study determined the factors affecting the accuracy of the prediction model.

**Existing System**

How to predict students’ performance is always a question concerned by the students’ teachers and parents. Based on the past examination results and in-class assessments, it is possible to forecast the future development of the students. It is a challenging and important matters as it involves the large volume of data in educational databases and the result could impact the future development of a young kid. A good and accuracy prediction could bring the benefits and impacts to students, educators and academic institutions. Various type of data mining techniques had been used for performance prediction for past decades but not satisfactory.

**Disadvantages:**

* There is limited research in students’ performance.
* Not suitable results.

**Proposed System:**

In this paper, we will investigate how to use artificial intelligence and deep learning algorithm for pattern recognition and correlation of assessment results. There are some traditional data mining techniques that have been used to predict students’ performance. Some researches educational data mining method had been done to identify those important attributes in students data. Neural network is another emerging technique used in educational data mining. The advantage of neural network is that it has the ability to detect all possible interactions between predictors variables. When more computing power is nowadays available, more layers of neural network can be implemented and deep learning analysis can be practically implemented. Deep learning could perform detection even in a complex nonlinear relationship between dependent and independent variables. It is considered as one of the best prediction method.

**Advantages:**

* Best Prediction for student performance.
* Calculating student performance using ML techniques.

**SYSTEM CONFIGURATION:**

**Hardware requirements:**

Processer                     :           Any Update Processer

Ram                             :           Min 4 GB

Hard Disk                   :           Min 100 GB

**Software requirements:**

Operating System       :           Windows family

Technology                 :           Python 3.6

IDE : PyCharm

**INTRODUCTION**

How to predict students’ performance is always aquestion concerned by the students’ teachers and parents.Based on the past examination results and in-classassessments, it is possible to forecast the future developmentof the students. It is a challenging and important matters as it

involves the large volume of data in educational databasesand the result could impact the future development of ayoung kid. A good and accuracy prediction could bring thebenefits and impacts to students, educators and academicinstitutions. Various type of data mining techniques hadbeen used for performance prediction for decades, e.g.decision tree, Naive Bayes, K-Nearest Neighbor and SupportVector Machine [1]. However, with the rise of artificialintelligence and deep learning application, using AI enginesuch as Google Tensorflow for pattern recognition has nowbeen rising it importance. In this paper, we will investigatehow to use artificial intelligence and deep learning algorithmfor pattern recognition and correlation of assessment results.There are some traditional data mining techniques that havebeen used to predict students’ performance. Some researcheseducational data mining method had been done to identifythose important attributes in students data.II. TRADITIONAL METHOD FOR CLASSIFICATION ANDPREDICTIONIn order to build the predictive modeling, there are

several traditional tasks used for example classification,regression and categorization. Algorithm such as associationrules and decision tree are commonly used.Association rule reflects the interdependence andcorrelation between different things. It is commonly used in

physical stores and e-commerce recommendation systems.Likewise, it can be used for recommending schools andsubjects to students. Support, confidence are two keyconcepts of them. In a given database, each transactioncontains a set of items. Every rule is composed by twodifferent itemsets X and Y. Support is the percentage of thetransaction contains both X and Y, confidence is thepercentage of Y that contains X. In other words, support isthe probability while confidence is the conditionalprobability. If the minimum thresholds of them, which have

to be set by us, are satisfied, there is an association thatexists between X and Y. Association rule mining is moreapplicable to the situation where the index in the record arediscrete value. If the indicator values in the original databaseare continuous data, and appropriate discretization should beperformed previously.The classification decision tree model is a tree structure

that describes the classification of instances. The decisiontree consists of nodes and directed edges. There are twotypes of nodes: internal nodes and leaf nodes, internal nodesrepresent a feature or attribute, and leaf nodes represent aclass.When categorizing, a certain feature of the instance istested starting from the root node, and the instance isassigned to its child node according to the test result; at thistime, each child node corresponds to a value of the feature.

This recursively moves down until it reaches the leaf node,and finally assigns the instance to the class of the leaf node.The decision tree can be seen as a set of if-then rules: a ruleis constructed from the root node of the decision tree to eachpath of the leaf nodes; the characteristics of the internalnodes on the path correspond to the conditions of the rules,and the leaf nodes Corresponds to the classification of theconclusions.

**Literature survey**

Authors:-A.O.Ogunde, and D.A .Ajibade.

Paper:-" Data Mining System for Predicting University Students"Graduation Grades Using ID3 Decision Tree Algorithm. Journal of Computer Science and Information Technology, 2015.

* The desire of every organization is to extract hidden but useful knowledge from
* their data through data mining tools. Also, the recent decline in the standard of
* education in most developing countries has necessitated researches that will help
* proffer solutions to some of the problems. From the literature, different analysis has
* been carried out on university data, which includes student’s university entrance
* examination and Ordinary level results but the relationship between these entry
* results and students’ final graduation grades has been in isolation. Therefore, in this
* work, a new system that will predict students’ graduation grades based on entry
* results data using the Iterative Dichotomiser 3 (ID3) decision tree algorithm was
* developed. ID3 decision tree algorithm was used to train the data of the graduated
* sets. The knowledge represented by decision trees were extracted and presented in
* form of IF-THEN rules. The trained data were then used to develop a model for
* making future prediction of students’ graduation grades. The developed system
* could be very useful in predicting students’ final graduation grades even from the
* point of entry into the university. This will help management staff, academic
* planners to properly counsel students in order to improve their overall performance.

Authors:-R.S.J.D.Baker.

paper:-"Data mining for education. International encyclopedia of

education", 7, 112-118, 2010.

* Data mining techniques are used to extract useful knowledge from raw data. The extracted knowledge is valuable and significantly affects the decision maker. Educational data mining (EDM) is a method for extracting useful information that could potentially affect an organization.
* The increase of technology use in educational systems has led to the storage of large amounts of student data, which makes it important to use EDM to improve teaching and learning processes.
* EDM is useful in many different areas including identifying at-risk students, identifying priority learning needs for different groups of students, increasing graduation rates, effectively assessing institutional performance, maximizing campus resources, and optimizing subject curriculum renewal.
* This paper surveys the relevant studies in the EDM field and includes the data and methodologies used in those studies.

Authors:-R.S.J.D.Baker, and KalinaYacef.

Paper:-"The state of educational data mining in 2009: A review and future visions". JEDM-Journal of Educational Data Mining, 3-17, 2009.

* We review the history and current trends in the field of Educational Data Mining (EDM). We consider the methodological profile of research in the early years of EDM, compared to in 2008 and 2009, and discuss trends and shifts in the research conducted by this community.
* In particular, we discuss the increased emphasis on prediction, the emergence of work using existing models to make scientific discoveries (discovery with models), and the reduction in the frequency of relationship mining within the EDM community.
* We discuss two ways that researchers have attempted to categorize the diversity of research in educational data mining research, and review the types of research problems that these methods have been used to address.
* The mostcited papers in EDM between 1995 and 2005 are listed, and their influence on the EDM community (and beyond the EDM community) is discussed.

Authors;-P.Cortez and A.Silva.

paper:-"Using Data Mining to Predict Secondary School Student Performance", 2008. In A. Brito and J. Teixeira Eds. Proceedings of 5th FUtureBUsinessTEChnology Conference,5-12, Porto, Portugal EUROSIS, ISBN 978-9077381-39-7, 2008.

* Although the educational level of the Portuguese pop-ulation has improved in the last decades, the statistics keep Portugal at Europe's tail end due to its high stu-dent failure rates. In particular, lack of success in the core classes of Mathematics and the Portuguese lan-guage is extremely serious
* . On the other hand, the fields of Business Intelligence (BI)/Data Mining (DM), which aim at extracting high-level knowledge from raw data, offer interesting automated tools that can aid the education domain.
* The present work intends to ap-proach student achievement in secondary education us-ing BI/DM techniques. Recent real-world data (e.g. student grades, demographic, social and school related features) was collected by using school reports and ques-tionnaires.
* The two core classes (i.e. Mathematics and Portuguese) were modeled under binary/five-level clas-sification and regression tasks. Also, four DM mod-els (i.e. Decision Trees, Random Forest, Neural Net-works and Support Vector Machines) and three input selections (e.g. with and without previous grades) were tested.
* The results show that a good predictive accuracy can be achieved, provided that the first and/or second school period grades are available. Although student achievement is highly influenced by past evaluations, an explanatory analysis has shown that there are also other relevant features (e.g. number of absences, parent's job and education, alcohol consumption).
* As a direct out-come of this research, more efficient student prediction tools can be be developed, improving the quality of ed-ucation and enhancing school resource management.

Authors:-Mr.M.N.Quadri and Dr.N.V.Kalyankar.

Paper:-"Drop Out characteristics of Student Data for Academic Performance Using Decision Tree Techniques". Global Journal of Computer Science and Technology, 2010.

* Students’ academic performance is critical for
* educational institutions because strategic programs can be
* planned in improving or maintaining students’ performance
* during their period of studies in the institutions. The academic
* performance in this study is measured by their cumulative
* grade point average (CGPA) upon graduating. This study
* presents the work of data mining in predicting the drop out
* feature of students. This study applies decision tree technique
* to choose the best prediction and analysis. The list of students
* who are predicted as likely to drop out from college by data
* mining is then turned over to teachers and management for
* direct or indirect intervention.

**3. System analysis**

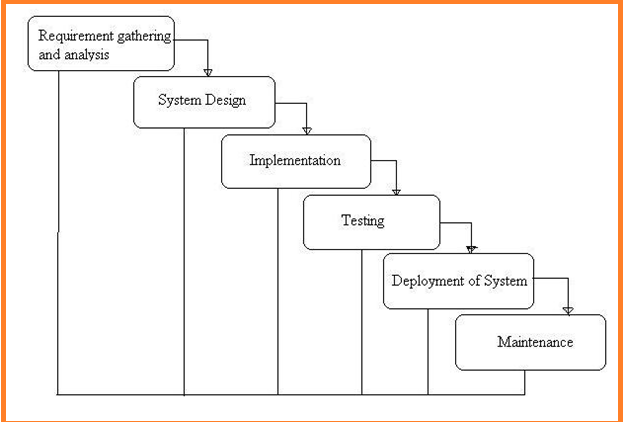


Fig:-1 Project SDLC

• Project Requisites Accumulating and Analysis

• Application System Design

• Practical Implementation

• Manual Testing of My Application

• Application Deployment of System

• Maintenance of the Project

**Requisites Accumulating and Analysis**

It’s the first and foremost stage of the any project as our is a an academic leave for requisites amassing we followed of IEEE Journals and Amassed so many IEEE Relegated papers and final culled a Paper designated “Individual web revisitation by setting and substance importance input and for analysis stage we took referees from the paper and did literature survey of some papers and amassed all the Requisites of the project in this stage

**System Design**

In System Design has divided into three types like GUI Designing, UML Designing with avails in development of project in facile way with different actor and its utilizer case by utilizer case diagram, flow of the project utilizing sequence, Class diagram gives information about different class in the project with methods that have to be utilized in the project if comes to our project our UML Will utilizable in this way The third and post import for the project in system design is Data base design where we endeavor to design data base predicated on the number of modules in our project

**Implementation**

The Implementation is Phase where we endeavor to give the practical output of the work done in designing stage and most of Coding in Business logic lay coms into action in this stage its main and crucial part of the project

**Testing**

**Unit Testing**

It is done by the developer itself in every stage of the project and fine-tuning the bug and module predicated additionally done by the developer only here we are going to solve all the runtime errors

**Manual Testing**

As our Project is academic Leave we can do any automatic testing so we follow manual testing by endeavor and error methods

**Deployment of System**

Once the project is total yare we will come to deployment of client system in genuinely world as its academic leave we did deployment i our college lab only with all need Software’s with having Windows OS

**Maintenance**

The Maintenance of our Project is one time process only

**FUNCTIONAL REQUIREMENT**

* Admin login
* Classification
* Predict
* Upload Data Set
* View Data Set

**NON-FUNCTIONAL REQUIREMENT**

**Expanded System admin security:** overseer to eschew the abuse of the application by PC ought to be exceptionally secured and available.

**Compactness:** The Presentation of this application is facile to utilize so it is looks simple for the using client to comprehend and react to identically tantamount.

**Unwavering quality:** and the functionalities accessible in the application thissubstructure has high probability to convey us the required inquiries.

**Time take for Reaction:** The time taken by the application to culminate an undertaking given by the client is very fast.

**Multifariousness:** Our application can be stretched out to incorporate the vicissitudes done by applications present now to enhance the performance of the item. This is implicatively insinuated for the future works that will be done on the application.

**Vigor:** The project is blame tolerant concerning illicit client/beneficiary sources of info. Blunder checking has been worked in the platforms to avert platforms disappointment.

**FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

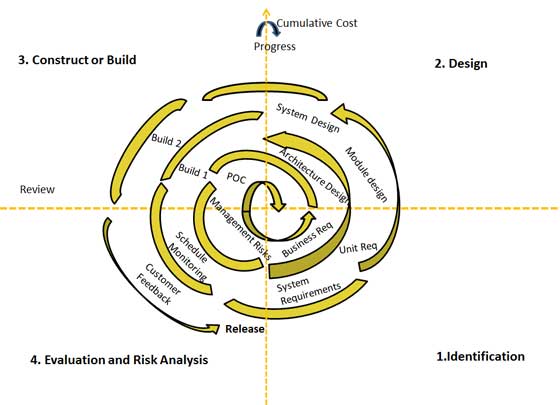
**SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**PROCESS MODEL USED WITH JUSTIFICATION**

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

**SDLC (Spiral Model):**

****

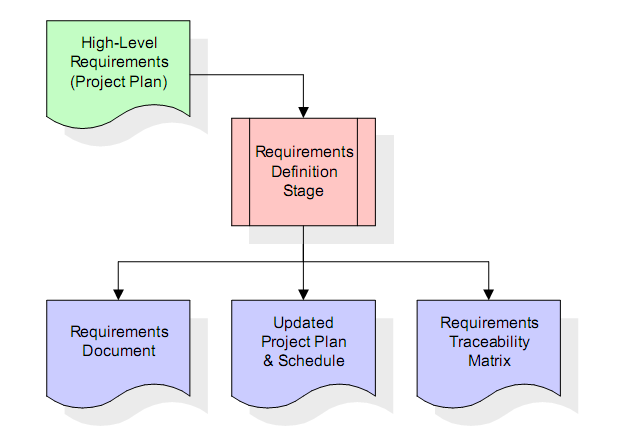
**Stages of SDLC:**

Requirement Gathering and Analysis

* Designing
* Coding
* Testing
* Deployment

**Requirements Definition Stage and Analysis:**

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.

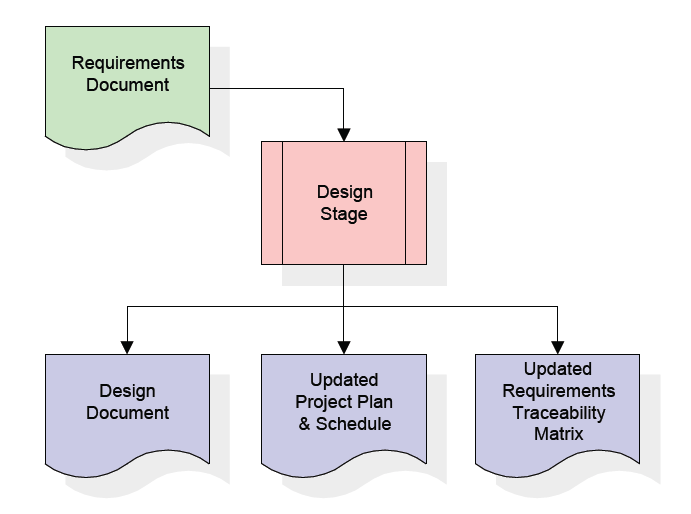


These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). the requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are *not* included in the requirements document. The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term *requirements traceability*. The outputs of the requirements definition stage include the requirementsdocument, the RTM, and an updated project plan.

**Design Stage:**

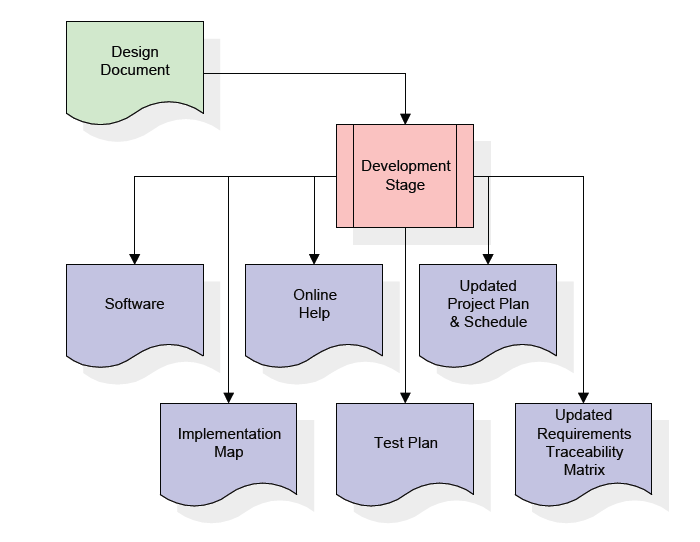
The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.



When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

**Development Stage:**

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.

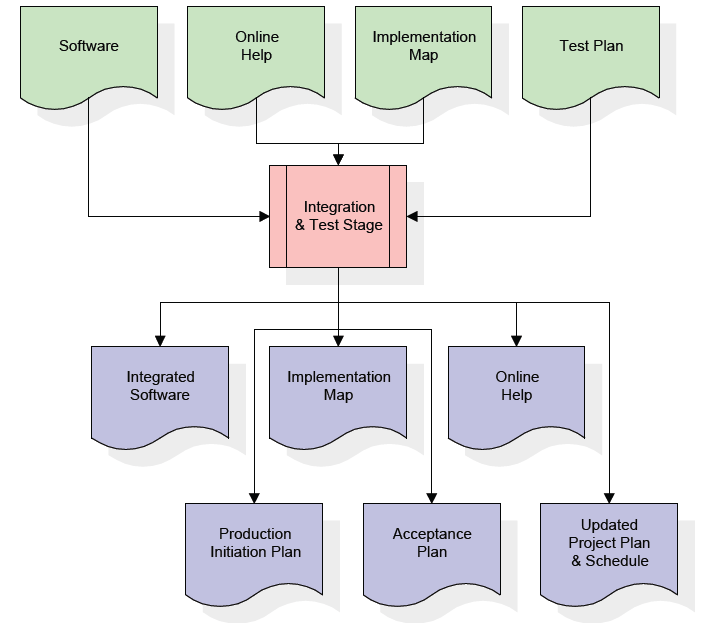


The RTM will be updated to show that each developed artifact is linked to a specific design element, and that each developed artifact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

**Integration & Test Stage:**

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability.

During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled intothe Production Initiation Plan.

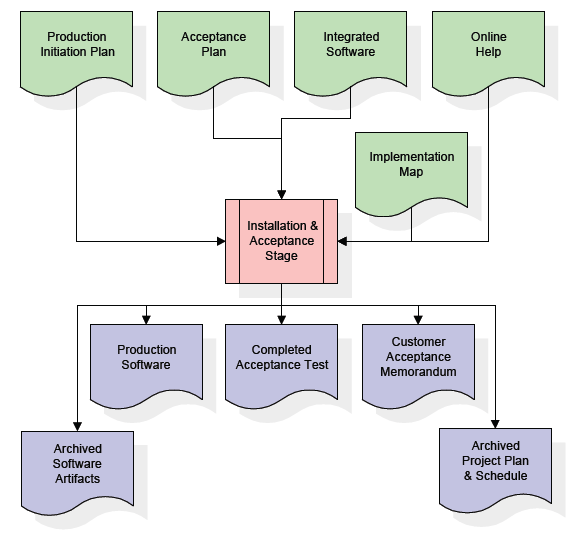


The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

**Installation & Acceptance Stage**

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.



The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labor data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

**SOFTWARE OVER VIEW:**

**History of Python**

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

**Input as CSV File**

Reading data from CSV(comma separated values) is a fundamental necessity in Data Science. Often, we get data from various sources which can get exported to CSV format so that they can be used by other systems. The Panadas library provides features using which we can read the CSV file in full as well as in parts for only a selected group of columns and rows.

The CSV file is a text file in which the values in the columns are separated by a comma. Let's consider the following data present in the file named input.csv.You can create this file using windows notepad by copying and pasting this data. Save the file as input.csv using the save As All files(\*.\*) option in notepad.

import pandas aspd

data =pd.read\_csv('path/input.csv')

print(data)

**Operations using NumPy**

NumPy is a Python package which stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

Using NumPy, a developer can perform the following operations −

* Mathematical and logical operations on arrays.
* Fourier transforms and routines for shape manipulation.
* Operations related to linear algebra. NumPy has in-built functions for linear algebra and random number generation.

## Key Features of Pandas

* Fast and efficient DataFrame object with default and customized indexing.
* Tools for loading data into in-memory data objects from different file formats.
* Data alignment and integrated handling of missing data.
* Reshaping and pivoting of date sets.
* Label-based slicing, indexing and subsetting of large data sets.
* Columns from a data structure can be deleted or inserted.
* Group by data for aggregation and transformations.
* High performance merging and joining of data.
* Time Series functionality.

**Implementation**

**Sample Code**

**DBConnection.py**

import mysql.connector

class DBConnection:

@staticmethod

defgetConnection():

database = mysql.connector.connect(host="localhost", user="root", passwd="root", db='student')

return database

if \_\_name\_\_=="\_\_main\_\_":

print(DBConnection.getConnection())

**Classification.py**

from PyQt5 import QtCore, QtGui, QtWidgets

from DBConnection import DBConnection

from sklearn.ensemble import RandomForestClassifier

import numpy as np

import pandas as pd

import sys

from sklearn.metrics import classification\_report, f1\_score, accuracy\_score, confusion\_matrix

import time

from sklearn import metrics

import xlrd

class Classification(object):

defbrowsefile(self):

fileName, \_ = QtWidgets.QFileDialog.getOpenFileName(None, "Select File")

print(fileName)

self.lineEdit.setText(fileName)

defanalyz(self):

try:

database = DBConnection.getConnection()

cursor = database.cursor()

trainset = []

fname=self.lineEdit.text()

df = pd.read\_csv("heart.csv")

X = np.array(df.drop(['Disease status'], 1)) # Train DataSet

print("X=", X)

y = np.array(df['Disease status']) # Train Class

print("y=", y)

tf = pd.read\_csv(fname)

testdata = np.array(tf.drop(['Disease status'], 1))

#print("td=", testdata)

testdata = testdata.reshape(len(testdata), -1)

testdata1 = np.array(tf['Disease status'])

#print("td1=", testdata1)

rf = RandomForestClassifier()

rf.fit(X, y)

s = time.clock()

result = rf.predict(testdata)

print("ress1", result)

print("ress",testdata1)

accuracy = self.model\_assessment\_dt(testdata1, result)

query = "insert into graph values(%s,%s)"

values = (len(tf), accuracy)

cursor.execute(query, values)

database.commit()

self.showMessageBox("Information", "Analysed Successfully")

self.lineEdit.setText("")

return result

except Exception as e:

print("Error=" + e.args[0])

tb = sys.exc\_info()[2]

print(tb.tb\_lineno)

defmodel\_assessment\_dt(self,y\_test, predicted\_class):

print('accuracy')

# Accuracy = (TP + TN) / ALL

accuracy = accuracy\_score(y\_test, predicted\_class) \* 100

print(accuracy)

return accuracy

defshowMessageBox(self, title, message):

msgBox = QtWidgets.QMessageBox()

msgBox.setIcon(QtWidgets.QMessageBox.Information)

msgBox.setWindowTitle(title)

msgBox.setText(message)

msgBox.setStandardButtons(QtWidgets.QMessageBox.Ok)

msgBox.exec\_()

defsetupUi(self, Dialog):

Dialog.setObjectName("Dialog")

Dialog.resize(544, 388)

Dialog.setStyleSheet("background-color: rgb(85, 170, 127);")

self.label = QtWidgets.QLabel(Dialog)

self.label.setGeometry(QtCore.QRect(200, 70, 141, 20))

self.label.setStyleSheet("font: 75 16pt \"Times New Roman\";")

self.label.setObjectName("label")

self.label\_2 = QtWidgets.QLabel(Dialog)

self.label\_2.setGeometry(QtCore.QRect(110, 130, 71, 16))

self.label\_2.setStyleSheet("font: 75 12pt \"Times New Roman\";")

self.label\_2.setObjectName("label\_2")

self.lineEdit = QtWidgets.QLineEdit(Dialog)

self.lineEdit.setGeometry(QtCore.QRect(110, 160, 251, 31))

self.lineEdit.setText("")

self.lineEdit.setObjectName("lineEdit")

self.pushButton = QtWidgets.QPushButton(Dialog)

self.pushButton.setGeometry(QtCore.QRect(380, 160, 75, 31))

self.pushButton.setStyleSheet("font: 75 12pt \"Times New Roman\";")

self.pushButton.setObjectName("pushButton")

self.pushButton\_2 = QtWidgets.QPushButton(Dialog)

self.pushButton\_2.setGeometry(QtCore.QRect(110, 210, 111, 31))

self.pushButton\_2.setStyleSheet("background-color: rgb(85, 85, 0);\n"

"color: rgb(255, 255, 255);\n"

"font: 75 12pt \"Times New Roman\";")

self.pushButton\_2.setObjectName("pushButton\_2")

self.pushButton\_2.clicked.connect(self.analyz)

#self.pushButton\_3 = QtWidgets.QPushButton(Dialog)

#self.pushButton\_3.setGeometry(QtCore.QRect(240, 210, 111, 31))

#self.pushButton\_3.setStyleSheet("background-color: rgb(85, 0, 0);\n"

#"font: 75 12pt \"Times New Roman\";\n"

#"color: rgb(255, 255, 255);")

#self.pushButton\_3.setObjectName("pushButton\_3")

#self.pushButton\_3.clicked.connect(self.randomtree)

#self.pushButton\_4 = QtWidgets.QPushButton(Dialog)

#self.pushButton\_4.setGeometry(QtCore.QRect(200, 270, 75, 23))

#self.pushButton\_4.setStyleSheet("background-color: rgb(170, 85, 0);\n"

#"color: rgb(0, 0, 0);\n"

#"font: 75 12pt \"Times New Roman\";")

#self.pushButton\_4.setObjectName("pushButton\_4")

self.retranslateUi(Dialog)

QtCore.QMetaObject.connectSlotsByName(Dialog)

defretranslateUi(self, Dialog):

\_translate = QtCore.QCoreApplication.translate

Dialog.setWindowTitle(\_translate("Dialog", "Classification"))

self.label.setText(\_translate("Dialog", "Classification"))

self.label\_2.setText(\_translate("Dialog", "Select File"))

self.pushButton.setText(\_translate("Dialog", "Browse"))

self.pushButton.clicked.connect(self.browsefile)

self.pushButton\_2.setText(\_translate("Dialog", "Analysis"))

if \_\_name\_\_ == "\_\_main\_\_":

import sys

app = QtWidgets.QApplication(sys.argv)

Dialog = QtWidgets.QDialog()

ui = Classification()

ui.setupUi(Dialog)

Dialog.show()

sys.exit(app.exec\_())

**About MySQL:**

**MySQL** is a relational database management system (RDBMS)[]](http://en.wikipedia.org/wiki/MySQL#cite_note-1) that runs as a server providing multi-user access to a number of databases.  The SQL phrase stands for Structured Query Language.Free-software-open source projects that require a full-featured database management system often use MySQL. For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, Joomla, WordPress, phpBB, Drupal and other software built on the LAMP software stack. MySQL is also used in many high-profile, large-scale World Wide Web products, including Wikipedia, Google  , Facebook, and Twitter.

MySQL is the world's most popular open source database software, with over 100 million copies of its software downloaded or distributed throughout it's history. With its superior speed, reliability, and ease of use, MySQL has become the preferred choice for Web, Web 2.0, SaaS, ISV, Telecom companies and forward-thinking corporate IT Managers because it eliminates the major problems associated with downtime, maintenance and administration for modern, online applications.

Many of the world's largest and fastest-growing organizations use MySQL to save time and money powering their high-volume Web sites, critical business systems, and packaged software — including industry leaders such as Yahoo!, Alcatel-Lucent, Google, Nokia, YouTube, Wikipedia, and Booking.com.

The flagship MySQL offering is MySQL Enterprise, a comprehensive set of production-tested software, proactive monitoring tools, and premium support services available in an affordable annual subscription.

MySQL is a key part of LAMP (Linux, Apache, MySQL, PHP / Perl / Python), the fast-growing open source enterprise software stack. More and more companies are using LAMP as an alternative to expensive proprietary software stacks because of its lower cost and freedom from platform lock-in.

MySQL was originally founded and developed in Sweden by two Swedes and a Finn: David Axmark, Allan Larsson and Michael "Monty" Widenius, who had worked together since the 1980's. More historical information on MySQL is

**Project Data Base**

CREATE DATABASE

USE `student`;

DROP TABLE IF EXISTS `dataset`;

CREATE TABLE `dataset` (

`math` varchar(100) DEFAULT NULL,

`chinese` varchar(100) DEFAULT NULL,

`eng` varchar(100) DEFAULT NULL,

`phy` varchar(100) DEFAULT NULL,

`che` varchar(100) DEFAULT NULL,

`bio` varchar(100) DEFAULT NULL,

`hist` varchar(100) DEFAULT NULL,

`condt` varchar(100) DEFAULT NULL,

`sprt` varchar(100) DEFAULT NULL,

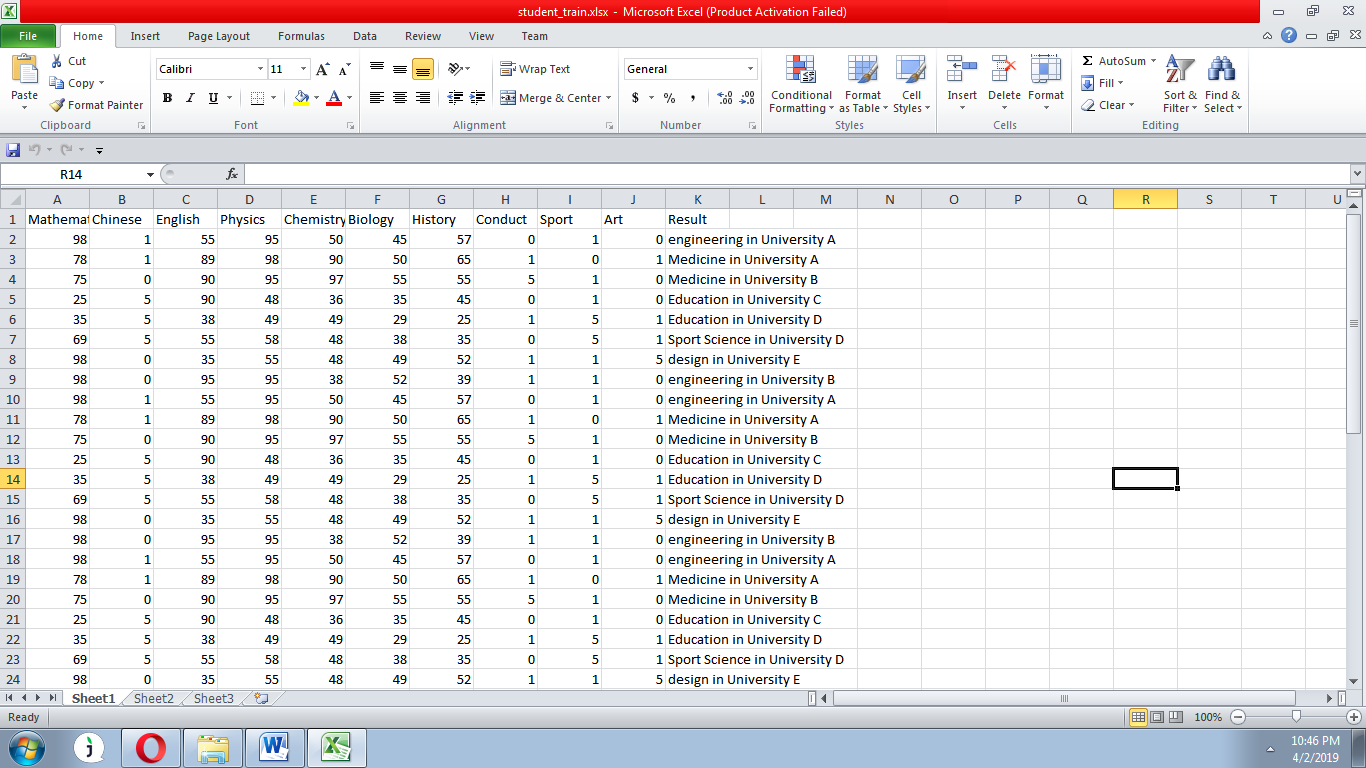
`art` varchar(100) DEFAULT NULL,

`result` varchar(100) DEFAULT NULL

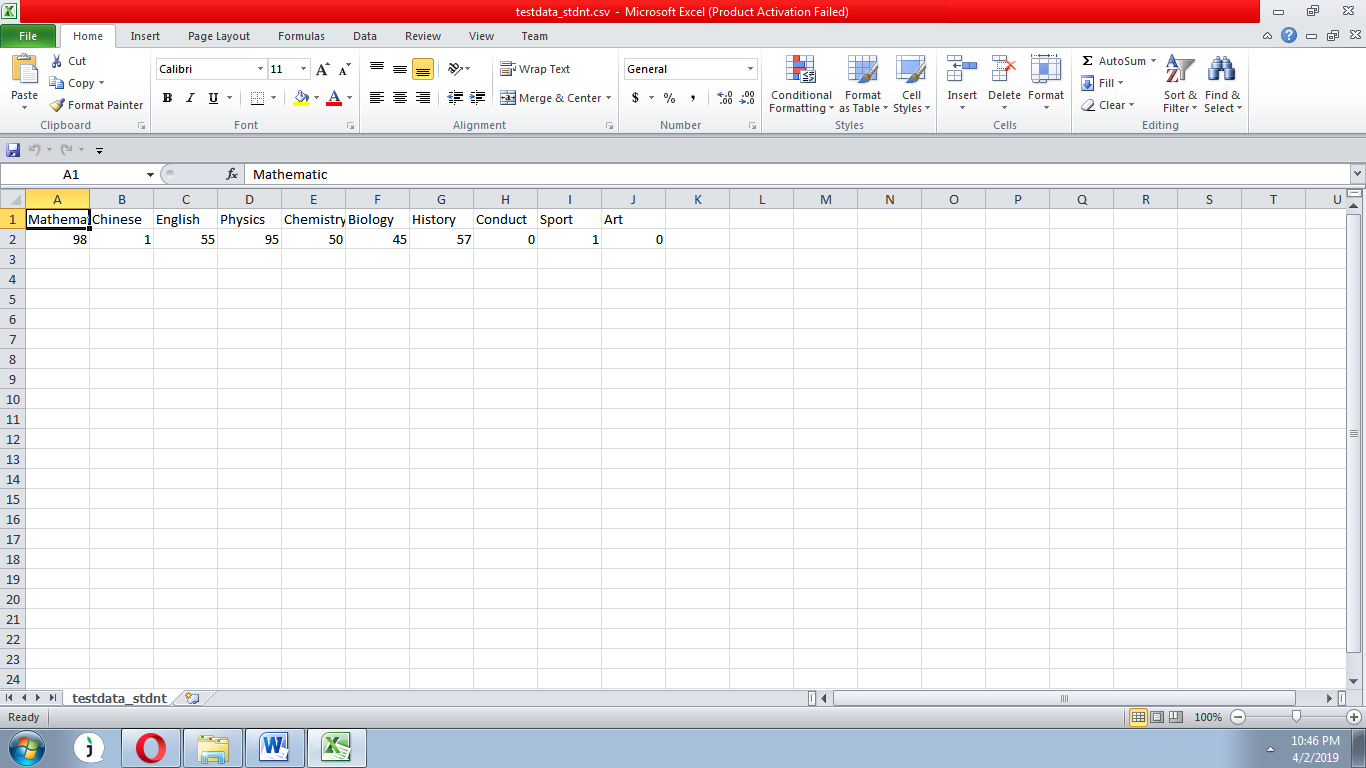
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

**Project Data Sets**

**student\_train.xlsx**



**testdata\_stdnt.csv**



**5. System Design:**

**5.2. UML DIAGRAMS**

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

**Global Use Case Diagrams:**

Identification of actors:

**Actor:** Actor represents the role a user plays with respect to the system. An actor interacts with, but has no control over the use cases.

Graphical representation:



<<Actor name>>

An actor is someone or something that:

Interacts with or uses the system.

* Provides input to and receives information from the system.
* Is external to the system and has no control over the use cases.

Actors are discovered by examining:

* Who directly uses the system?
* Who is responsible for maintaining the system?
* External hardware used by the system.
* Other systems that need to interact with the system.

Questions to identify actors:

* + Who is using the system? Or, who is affected by the system? Or, which groups need help from the system to perform a task?
  + Who affects the system? Or, which user groups are needed by the system to perform its functions? These functions can be both main functions and secondary functions such as administration.
  + Which external hardware or systems (if any) use the system to perform tasks?
  + What problems does this application solve (that is, for whom)?
  + And, finally, how do users use the system (use case)? What are they doing with the system?

The actors identified in this system are:

1. **System Administrator**
2. **Customer**
3. **Customer Care**

Identification of usecases:

**Usecase:** A use case can be described as a specific way of using the system from a user’s (actor’s) perspective.

**Graphical representation:**



A more detailed description might characterize a use case as:

* Pattern of behavior the system exhibits
* A sequence of related transactions performed by an actor and the system
* Delivering something of value to the actor

Use cases provide a means to:

* capture system requirements
* communicate with the end users and domain experts
* test the system

Use cases are best discovered by examining the actors and defining what the actor will be able to do with the system.

Guide lines for identifying use cases:

* For each actor, find the tasks and functions that the actor should be able to perform or that the system needs the actor to perform. The use case should represent a course of events that leads to clear goal
* Name the use cases.
* Describe the use cases briefly by applying terms with which the user is familiar.

This makes the description less ambiguous

Questions to identify use cases:

* What are the tasks of each actor?
* Will any actor create, store, change, remove or read information in the system?
* What use case will store, change, remove or read this information?
* Will any actor need to inform the system about sudden external changes?
* Does any actor need to inform about certain occurrences in the system?
* What usecases will support and maintains the system?

**Flow of Events**

A flow of events is a sequence of transactions (or events) performed by the system. They typically contain very detailed information, written in terms of what the system should do, not how the system accomplishes the task. Flow of events are created as separate files or documents in your favorite text editor and then attached or linked to a use case using the Files tab of a model element.

A flow of events should include:

* When and how the use case starts and ends
* Use case/actor interactions
* Data needed by the use case
* Normal sequence of events for the use case
* Alternate or exceptional flows

Construction of Usecase diagrams:

Use-case diagrams graphically depict system behavior (use cases). These diagrams present a high level view of how the system is used as viewed from an outsider’s (actor’s) perspective. A use-case diagram may depict all or some of the use cases of a system.

A use-case diagram can contain:

* actors ("things" outside the system)
* use cases (system boundaries identifying what the system should do)
* Interactions or relationships between actors and use cases in the system including the associations, dependencies, and generalizations.

Relationships in use cases:

**1. Communication:**

The communication relationship of an actor in a usecase is shown by connecting the actor symbol to the usecase symbol with a solid path. The actor is said to communicate with the usecase.

**2. Uses:**

A Uses relationship between the usecases is shown by generalization arrow from the usecase.

**3. Extends:**

The extend relationship is used when we have one usecase that is similar to another usecase but does a bit more. In essence it is like subclass.

**SEQUENCE DIAGRAMS**

A sequence diagram is a graphical view of a scenario that shows object interaction in a time-based sequence what happens first, what happens next. Sequence diagrams establish the roles of objects and help provide essential information to determine class responsibilities and interfaces.

There are two main differences between sequence and collaboration diagrams: sequence diagrams show time-based object interaction while collaboration diagrams show how objects associate with each other. A sequence diagram has two dimensions: typically, vertical placement represents time and horizontal placement represents different objects.

**Object:**

An object has state, behavior, and identity. The structure and behavior of similar objects are defined in their common class. Each object in a diagram indicates some instance of a class. An object that is not named is referred to as a class instance.

The object icon is similar to a class icon except that the name is underlined:

An object's concurrency is defined by the concurrency of its class.

**Message:**

A message is the communication carried between two objects that trigger an event. A message carries information from the source focus of control to the destination focus of control. The synchronization of a message can be modified through the message specification. Synchronization means a message where the sending object pauses to wait for results.

**Link:**

A link should exist between two objects, including class utilities, only if there is a relationship between their corresponding classes. The existence of a relationship between two classes symbolizes a path of communication between instances of the classes: one object may send messages to another. The link is depicted as a straight line between objects or objects and class instances in a collaboration diagram. If an object links to itself, use the loop version of the icon.

**CLASS DIAGRAM:**

Identification of analysis classes:

A class is a set of objects that share a common structure and common behavior (the same attributes, operations, relationships and semantics). A class is an abstraction of real-world items.

There are 4 approaches for identifying classes:

1. Noun phrase approach:
2. Common class pattern approach.
3. Use case Driven Sequence or Collaboration approach.
4. Classes , Responsibilities and collaborators Approach
5. **Noun Phrase Approach:**

The guidelines for identifying the classes:

* + Look for nouns and noun phrases in the usecases.
  + Some classes are implicit or taken from general knowledge.
  + All classes must make sense in the application domain; Avoid computer

implementation classes – defer them to the design stage.

* + Carefully choose and define the class names After identifying the classes we have to eliminate the following types of classes:
* Adjective classes.

1. **Common class pattern approach:**

The following are the patterns for finding the candidate classes:

* + Concept class.
  + Events class.
  + Organization class
  + Peoples class
  + Places class
  + Tangible things and devices class.

1. **Use case driven approach:**

We have to draw the sequence diagram or collaboration diagram. If there is need for some classes to represent some functionality then add new classes which perform those functionalities.

1. **CRC approach:**

The process consists of the following steps:

* + Identify classes’ responsibilities ( and identify the classes )
  + Assign the responsibilities
  + Identify the collaborators.

Identification of responsibilities of each class:

The questions that should be answered to identify the attributes and methods of a class respectively are:

1. What information about an object should we keep track of?
2. What services must a class provide?

Identification of relationships among the classes:

Three types of relationships among the objects are:

Association: How objects are associated?

Super-sub structure: How are objects organized into super classes and sub classes?

Aggregation: What is the composition of the complex classes?

Association:

The **questions** that will help us to identify the associations are:

1. Is the class capable of fulfilling the required task by itself?
2. If not, what does it need?
3. From what other classes can it acquire what it needs?

Guidelines for identifying the tentative associations:

* A dependency between two or more classes may be an association. Association often corresponds to a verb or prepositional phrase.
* A reference from one class to another is an association. Some associations are implicit or taken from general knowledge.

Some common association patterns are:

Location association like part of, next to, contained in…..

Communication association like talk to, order to ……

We have to eliminate the unnecessary association like implementation associations, ternary or n-ary associations and derived associations.

Super-sub class relationships:

Super-sub class hierarchy is a relationship between classes where one class is the parent class of another class (derived class).This is based on inheritance.

Guidelines for identifying the super-sub relationship, a generalization are

1***.* Top-down*:***

Look for noun phrases composed of various adjectives in a class name. Avoid excessive refinement. Specialize only when the sub classes have significant behavior.

2.**Bottom-up*:***

Look for classes with similar attributes or methods. Group them by moving the common attributes and methods to an abstract class. You may have to alter the definitions a bit.

3.**Reusability*:***

Move the attributes and methods as high as possible in the hierarchy.

4. **Multiple inheritances*:***

Avoid excessive use of multiple inheritances. One way of getting benefits of multiple inheritances is to inherit from the most appropriate class and add an object of another class as an attribute.

**Aggregation or a-part-of relationship:**

It represents the situation where a class consists of several component classes. A class that is composed of other classes doesn’t behave like its parts. It behaves very difficultly. The major properties of this relationship are transitivity and anti symmetry.

The **questions** whose answers will determine the distinction between the part and whole relationships are:

* Does the part class belong to the problem domain?
* Is the part class within the system’s responsibilities?
* Does the part class capture more than a single value?( If not then simply include it as an attribute of the whole class)
* Does it provide a useful abstraction in dealing with the problem domain?

There are three types of aggregation relationships. They are:

**Assembly:**

It is constructed from its parts and an assembly-partsituation physically exists.

**Container:**

A physical whole encompasses but is not constructed from physical parts.

**Collection member:**

A conceptual whole encompasses parts that may be physical or conceptual. The container and collection are represented by hollow diamonds but composition is represented by solid diamond.

**Class**

****

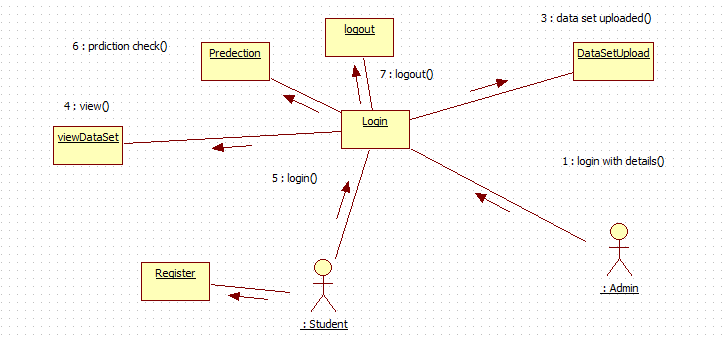
**Usecase**

****

**Seq**

****

**Collabrotion**

****

**State**

****

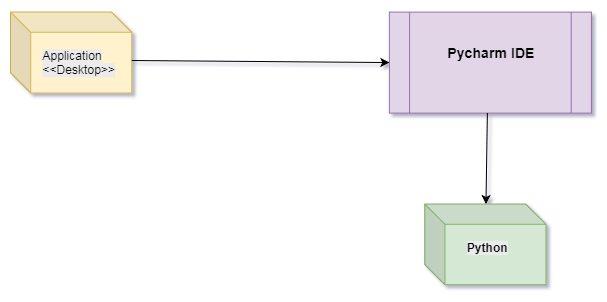
**Activity**

****

**Component**

****

**Deployment:**

****

**6. Software testing**

Software testing is one of the main stages of project development life cycle to provide our cessation utilizer with information about the quality of the application and ours, in our Project we have under gone some stages of testing like unit testing where it’s done in development stage of the project when we are in implementation of the application after the Project is yare we have done manual testing with different Case of all the different modules in the application we have even done browser compatibility testing in different web browsers in market, even we have done Client side validation testing on our application

**Unit testing**

The unit testing is done in the stage of implementation of the project only the error are solved in development stage some of the error we come across in development are given below

**TESTING**

Testing is the debugging program is one of the most critical aspects of the computer programming triggers, without programming that works, the system would never produce an output of which it was designed. Testing is best performed when user development is asked to assist in identifying all errors and bugs. The sample data are used for testing. It is not quantity but quality of the data used the matters of testing. Testing is aimed at ensuring that the system was accurately an efficiently before live operation commands.

**Testing objectives:**

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say, testing is a process of executing a program with intent of finding an error.

A successful test is one that uncovers an as yet undiscovered error.

A good test case is one that has probability of finding an error, if it exists.

The test is inadequate to detect possibly present errors.

The software more or less confirms to the quality and reliable standards.

**Levels of Testing:**

In order to uncover present in different phases we have the concept of levels of testing.

**The basic levels of Testing:**

Client needs acceptance testing

Requirements system testing

Design integration testing

Code unit testing

Figure: Levels of Testing

**Code testing:**

This examines the logic of the program. For example, the logic for updating various sample data and with the sample files and directories were tested and verified.

**Specification Testing:**

Executing this specification starting what the program should do and how it should performed under various conditions. Test cases for various situation and combination of conditions in all the modules are tested.

**Unit testing:**

In the unit testing we test each module individually and integrate with the overall system. Unit testing focuses verification efforts on the smallest unit of software design in the module. This is also known as module testing. The module of the system is tested separately. This testing is carried out during programming stage itself. In the testing step each module is found to work satisfactorily as regard to expected output from the module. There are some validation checks for fields also. For example the validation check is done for varying the user input given by the user which validity of the data entered. It is very easy to find error debut the system.

Each Module can be tested using the following two Strategies:

1. Black Box Testing
2. White Box Testing

**BLACK BOX TESTING**

**What is Black Box Testing?**

Black box testing is a software testing techniques in which **functionality of the software under test (SUT) is tested without looking at the internal code structure**, implementation details and knowledge of internal paths of the software. This type of testing is based entirely on the software requirements and specifications.

**In Black Box Testing we just focus on inputs and output of the software system** without bothering about internal knowledge of the software program.



The above Black Box can be any software system you want to test. For example : an operating system like Windows, a website like Google ,a database like Oracle or even your own custom application. Under Black Box Testing, you can test these applications by just focusing on the inputs and outputs without knowing their internal code implementation.

**Black box testing - Steps**

Here are the generic steps followed to carry out any type of Black Box Testing.

* Initially requirements and specifications of the system are examined.
* Tester chooses valid inputs (positive test scenario) to check whether SUT processes them correctly. Also some invalid inputs (negative test scenario) are chosen to verify that the SUT is able to detect them.
* Tester determines expected outputs for all those inputs.
* Software tester constructs test cases with the selected inputs.
* The test cases are executed.
* Software tester compares the actual outputs with the expected outputs.
* Defects if any are fixed and re-tested.

**Types of Black Box Testing**

There are many types of Black Box Testing but following are the prominent ones -

* **Functional testing** – This black box testing type is related to functional requirements of a system; it is done by software testers.
* **Non-functional testing** – This type of black box testing is not related to testing of a specific functionality, but non-functional requirements  such as performance, scalability, usability.
* **Regression testing** – Regression testing is done  after code fixes , upgrades or any other system maintenance to check the new code has not affected the existing code.

**WHITE BOX TESTING**

White Box Testing is the testing of a software solution's internal coding and infrastructure.It focuses primarily on strengthening security, the flow of inputs and outputs through the application, and improving design and usability.White box testing is also known as **clear, open, structural, and glass box testing**.

It is one of two parts of the **"box testing" approach** of software testing. Its counter-part, blackbox testing, involves testing from an external or end-user type perspective. On the other hand, Whitebox testing is based on the inner workings of an application and revolves around internal testing. The term "whitebox" was used because of the see-through box concept. The clear box or whitebox name symbolizes the ability to see through the software's outer shell (or "box") into its inner workings. Likewise, the "black box" in "black box testing" symbolizes not being able to see the inner workings of the software so that only the end-user experience can be tested

## What do you verify in White Box Testing ?

White box testing involves the testing of the software code for the following:

* Internal security holes
* Broken or poorly structured paths in the coding processes
* The flow of specific inputs through the code
* Expected output
* The functionality of conditional loops
* Testing of each statement, object and function on an individual basis

The testing can be done at system, integration and unit levels of software development. One of the basic goals of whitebox testing is to verify a working flow for an application. It involves testing a series of predefined inputs against expected or desired outputs so that when a specific input does not result in the expected output, you have encountered a bug.

**How do you perform White Box Testing?**

To give you a simplified explanation of white box testing, we have divided it into **two basic steps**. This is what testers do when testing an application using the white box testing technique:

**STEP 1) UNDERSTAND THE SOURCE CODE**

The first thing a tester will often do is learn and understand the source code of the application. Since white box testing involves the testing of the inner workings of an application, the tester must be very knowledgeable in the programming languages used in the applications they are testing. Also, the testing person must be highly aware of secure coding practices. Security is often one of the primary objectives of testing software. The tester should be able to find security issues and prevent attacks from hackers and naive users who might inject malicious code into the application either knowingly or unknowingly.

**Step 2) CREATE TEST CASES AND EXECUTE**

The second basic step to white box testing involves testing the application’s source code for proper flow and structure. One way is by writing more code to test the application’s source code. The tester will develop little tests for each process or series of processes in the application. This  method requires that the tester must have intimate knowledge of the code and is often done by the developer. Other methods include manual testing, trial and error testing and the use of testing tools as we will explain further on in this article.

**System testing:**

Once the individual module testing is completed, modules are assembled and integrated to perform as a system. The top down testing, which began from upper level to lower level module, was carried out to check whether the entire system is performing satisfactorily.

There are three main kinds of System testing:

1. Alpha Testing
2. Beta Testing
3. Acceptance Testing

**Alpha Testing:**

This refers to the system testing that is carried out by the test team with the Organization.

**Beta Testing**:

This refers to the system testing that is performed by a selected group of friendly customers

**Acceptance Testing:**

This refers to the system testing that is performed by the customer to determine whether or not to accept the delivery of the system.

**Integration Testing:**

Data can be lost across an interface, one module can have an adverse effort on the other sub functions, when combined, may not produce the desired major functions. Integrated testing is the systematic testing for constructing the uncover errors within the interface. The testing was done with sample data. The developed system has run successfully for this sample data. The need for integrated test is to find the overall system performance.

**Output testing:**After performance of the validation testing, the next step is output testing. The output displayed or generated by the system under consideration is tested by asking the user about the format required by system.

**CONCLUSIONS**

Classification and prediction is a general problem.Traditional data mining techniques such as association rules,decision tree, clustering and so on had been use for a fewdecades for solving this problem. The raising popularity ofusing Tensorflow for deep learning and artificial intelligenceopened a new approach and direction for solvingclassification problem and prediction of non-linear results.In this research, the number of hidden layers, hiddennotes, the number of iteration and the learning rate areadjusted and compared. It is discovered that it is not alwaystrue that the deep the deep learning model, i.e. the morenumber of hidden layer, the more accurate will the result be.There is an optimal point at are required to be tested andidentified.For the learning rate, a higher learning rate could help tospeed up the convergence of the trained model. However, ifthe learning rate is too high, the result might overshoot theoptimal point. Therefore, the prediction performance couldbe improved with a low momentum and high learning rate.Then gradually, the momentum can be increased and thelearning rate can be decreased for ensuring convergence.This study demonstrated that deep learning could be aneffective tool for predicting the students’ performance. Theresult ranged from 80% to 91%. The prediction result isgood enough to provide appropriate recommendations forstudents, their teachers and parents to decide theirdevelopment pathway. It is believed that more applicationsof deep learning could be used for education and corporatestaff training in the future.

**REFERENCE**

[1] Amirah Mohamed, ShahiriWahidah, “A Review on PredictingStudent's Performance Using Data Mining Techniques”, ProcediaComputer Science, Volume 72, 2015, Pages 414-422,

<https://www.sciencedirect.com/science/article/pii/S18770509150361>82

[2] G. Gray, C. McGuinness, P. Owende, An application of classificationmodels to predict learner progression in tertiary education, in:Advance Computing Conference (IACC), 2014 IEEE International,IEEE, 2014, pp. 549–554.

[3] P. M. Arsad, N. Buniyamin, J.-l. A. Manan, A neural networkstudents’ performance prediction model (nnsppm), in: SmartInstrumentation, Measurement and Applications (ICSIMA), 2013IEEE International Conference on, IEEE, 2013, pp. 1–5.

[4] Joonhyuck Lee , Dongsik Jang and Sangsung Park, “Deep Learning-Based Corporate Performance Prediction Model ConsideringTechnical Capability” MDP <http://www.mdpi.com/2071->1050/9/6/899/pdf

[5] RianneConijn, Chris Snijders, Ad Kleingeld, UweMatzat,“Predicting Student Performance from LMS Data: A Comparison of17 Blended Courses Using Moodle LMS. October 2016IEEE

Transactions on Learning Technologies PP(99):1-1106