UNIVERSITY OF THE PUNJAB

JHELUM CAMPUS

Department of Computer Science

Project Title:

Transforming Alzheimer's Detection: *Deep Learning Solution for Early Diagnosis.*

Supervisor: Sir Adeel Ahmad

SYSTEM REQUIREMENTS SPECIFICATION

3.1 Functional Requirements – Dataset preparation and pre processing

3.1.1 Data Collection

Data collection is defined as the procedure of collecting, measuring and analysing accurate insights for research using standard validated techniques. A researcher can evaluate their hypothesis based on collected data. In most cases, data collection is the primary and most important step for research, irrespective of the field of research. The approach of data collection is different for different fields of study, depending on the required information. The most critical objective of data collection is ensuring that information-rich and reliable data is collected for statistical analysis so that data-driven decisions can be made for research.

3.1.2 Data Visualization

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. Example -

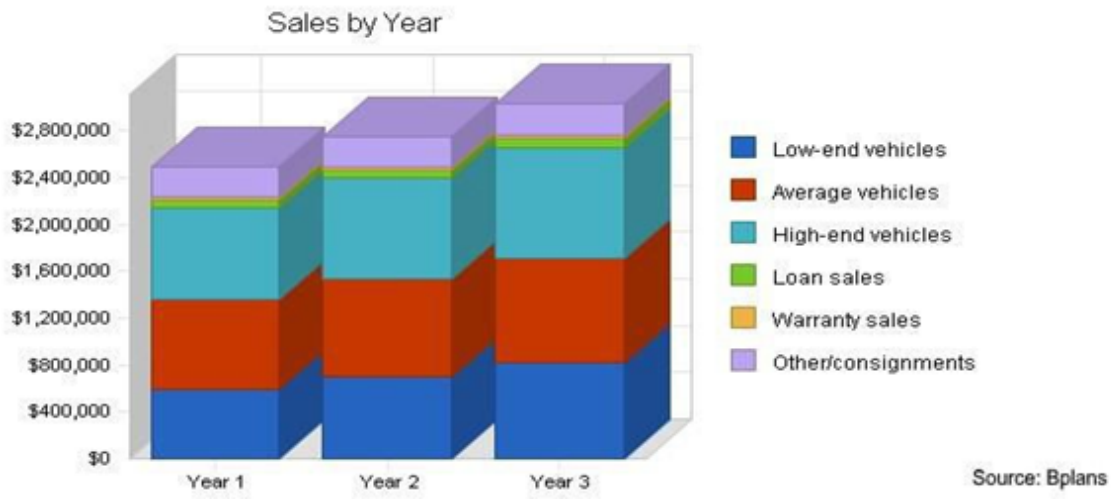

Fig 3.1 – Data Visualization

3.1.3 Data Labelling

Supervised machine learning, which we'll talk about below, entails training a predictive model on historical data with predefined target answers. An algorithm must be shown which target answers or attributes to look for. Mapping these target attributes in a dataset is called labelling. Data labelling takes much time and effort as datasets sufficient for machine learning may require thousands of records to be labelled. For instance, if your image recognition algorithm must classify types of bicycles, these types should be clearly defined and labelled in a dataset.

3.1.4 Data Selection

Data selection is defined as the process of determining the appropriate data type and source, as well as suitable instruments to collect data. Data selection precedes the actual practice of data collection. This definition distinguishes data selection from selective data reporting (selectively excluding data that is not supportive of a research hypothesis) and interactive/active data selection (using collected data for monitoring activities/events, or conducting secondary data analyses). The process of selecting suitable data for a research project can impact data integrity. After having collected all information, a *data analyst* chooses a subgroup of data to solve the

defined problem. For instance, if you save your customers' geographical location, you don't need to add their cell phones and bank card numbers to a dataset. But purchase history would be necessary. The selected data includes attributes that need to be considered when building a predictive model.

3.1.5 Data Pre-processing

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviours or trends, and is likely to contain many errors. Data pre- processing is a proven method of resolving such issues. The purpose of pre-processing is to convert raw data into a form that fits machine learning. Structured and clean data allows a data scientist to get more precise results from an applied machine learning model. The technique includes data formatting, cleaning, and sampling.

- Data formatting. The importance of data formatting grows when data is acquired from various sources by different people. The first task for a data scientist is to standardize record formats. A specialist checks whether variables representing each attribute are recorded in the same way. Titles of products and services, prices, date formats, and addresses are examples of variables. The principle of data consistency also applies to attributes represented by numeric ranges.
- Data cleaning. This set of procedures allows for removing noise and fixing inconsistencies in data. A data scientist can fill in missing data using imputation techniques, e.g. substituting missing values with

mean attributes. A specialist also detects outliers — observations that deviate significantly from the rest of distribution. If an outlier indicates erroneous data, a data scientist deletes or corrects them if possible. This stage also includes removing incomplete and useless data objects.

- Data anonymization**.** Sometimes a data scientist must anonymize or exclude attributes representing sensitive information (i.e. when working with healthcare and banking data).
- Data sampling. Big datasets require more time and computational power for analysis. If a dataset is too large, applying data sampling is the way to go. A data scientist uses this technique to select a smaller but representative data sample to build and run models much faster, and at the same time to produce accurate outcomes.

3.1.6 Data Transformation

Data transformation is the process of converting data from one format or structure into another format or structure. Data transformation is critical to activities such as data integration and data management. Data transformation can include a range of activities: you might convert data types, cleanse data by removing nulls or duplicate data, enrich the data, or perform aggregations, depending on the needs of your project.

- Scaling. Data may have numeric attributes (features) that span different ranges, for example, millimetres, meters, and kilometres. Scaling is about converting these attributes so that they will have the same scale, such as

between 0 and 1, or 1 and 10 for the smallest and biggest value for an attribute.

- Decomposition. Sometimes finding patterns in data with features representing complex concepts is more difficult. Decomposition technique can be applied in this case. During decomposition, a specialist converts higher level features into lower level ones. In other words, new features based on the existing ones are being added. Decomposition is mostly used in time series analysis. For example, to estimate a demand for air conditioners per month, a market research analyst converts data representing demand per quarters.
- Aggregation. Unlike decomposition, aggregation aims at combining several features into a feature that represents them all. For example, you have collected basic information about your customers and particularly their age. To develop a demographic segmentation strategy, you need to distribute them into age categories, such as 16-20, 21-30, 31-40, etc. You use aggregation to create large-scale features based on small- scale ones. This technique allows you to reduce the size of a dataset without the loss of information.

3.1.7 Data Splitting

A dataset used for machine learning should be partitioned into three subsets — training, test, and validation sets.

1. **Training set**. A data scientist uses a training set to train a model and define its optimal parameters — parameters it must learn from data.
2. **Test set**. A test set is needed for an evaluation of the trained model and its capability for generalization. The latter means a model's ability to identify

patterns in new unseen data after having been trained over a training data. It is crucial to use different subsets for training and testing to avoid model overfitting, which is the incapacity for generalization we mentioned above.

3. **Validation set**. The purpose of a validation set is to tweak a model's hyperparameters — higher-level structural settings that cannot be directly learned from data. These settings can express, for instance, how complex a model is and how fast it finds patterns in data.

The proportion of a training and a test set is usually 80 to 20 percent, respectively. A training set is then split again, and its 20 percent will be used to form a validation set. At the same time, machine learning practitioner Jason Brownlee suggests using 66 percent of data for training and 33 percent for testing. A size of each subset depends on the total dataset size

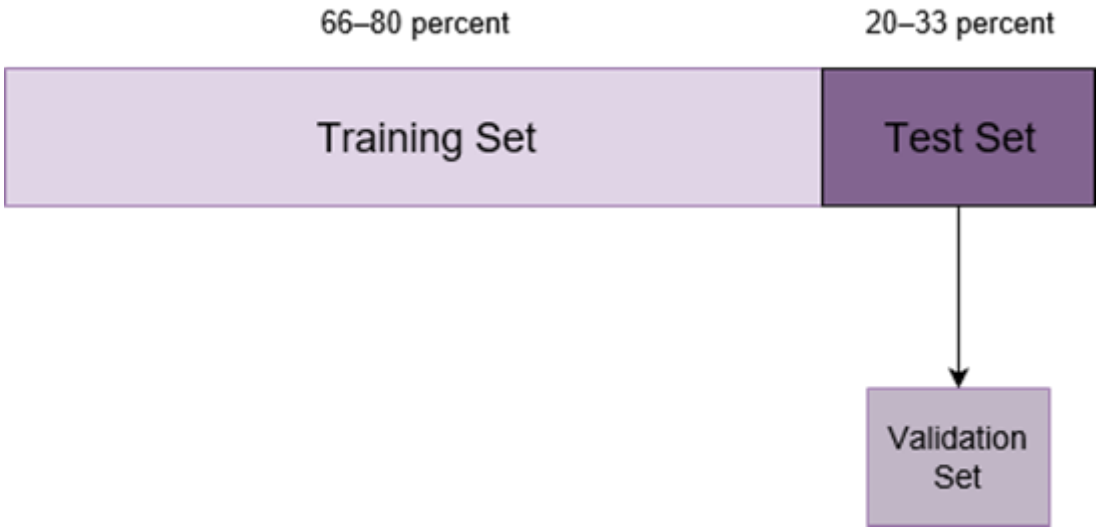

Fig 3.2 – Data Splitting

The more training data a data scientist uses, the better the potential model will perform. Consequently, more results of model testing data lead to better model performance and generalization capability.

3.1.8 Modelling

After pre-processing the collected data and splitting it into three subsets, we can proceed with a model training. This process entails “feeding” the algorithm with training data. An algorithm will process data and output a model that is able to find a target value (attribute) in new data — an answer you want to get with predictive analysis. The purpose of model training is to develop a model.

Two model training styles are most common — **supervised and unsupervised learning.** The choice of each style depends on whether you must forecast specific attributes or group data objects by similarities.

Model evaluation and testing: The goal of this step is to develop the simplest model able to formulate a target value fast and well enough. A data scientist can achieve this goal through model tuning. That is the optimization of model parameters to achieve an algorithm’s best performance. One of the more efficient methods for model evaluation and tuning is cross-validation.

Cross-validation: Cross-validation is the most used tuning method. It entails splitting a training dataset into ten equal parts (folds). A given model is trained on only nine folds and then tested on the tenth one (the one previously left out). Training continues until every fold is left aside and used for testing. As a result of model performance measure, a specialist calculates a cross-validated score for each set of hyperparameters. A data scientist trains models with different sets of hyperparameters to define which model has the highest prediction accuracy. The cross-validated score indicates average model performance across ten hold-out folds.

3.1.9 Model Deployment

Deployment is the method by which you integrate a machine learning model into an existing production environment to make practical business decisions based on data. It is one of the last stages in the machine learning life cycle and can be one of the most cumbersome. Often, an organization's IT systems are incompatible with traditional model-building languages, forcing data scientists and programmers to spend valuable time and brainpower rewriting them.

3.2 Non-Functional Requirements

3.2.1 Usability

The system should be easy to use. The system also should be user friendly for users because anyone can use it instead of programmers.

3.2.2 Reliability

This software will be developed with machine learning, feature engineering and deep learning techniques. So, in this step there is no certain reliable percentage that is measurable. Also, user provided data will be used to compare with results and measure reliability. With recent machine learning techniques, user gained data should be enough for reliability if enough data is obtained.

3.2.3 Performance

Processing time and response time should be as little as possible providing the result at a faster rate when compared to other methods.

3.2.4 Supportability

The system should require Python knowledge for maintenance. If any problem is acquired in user side and deep learning methods, it requires code knowledge and deep learning background to solve.

3.3 Hardware Requirements

- OS: Windows 10
- RAM: Minimum of 4GB

3.4 Software Requirements

- Basic Text-Editor: Visual Studio Code, Pycharm.
- Jupyter Notebook: Development of Python Scripts.
- Flask Framework: API Development.
- Angular Framework: UI Development.
-

3.4.1 Why Python?

- General purpose programming language
- Increasing popularity for use in data science
- Easy to build end-to-end products like web applications

Since the goal of this project is to build a web application, Python is a better choice. Though frameworks like Shiny can be used with R to create web applications, it is extremely slow.

There are two major libraries for machine learning in python: TensorFlow, ScikitLearn. The main differences are discussed below.

TensorFlow	**Scikit-Learn**
Low-Level, algorithms should be implemented manually	Provides off-the-shelf algorithms
Better choice for deep learning	Less efficient for deep learning
Steep learning curve	Easier to learn

Table 3.1 – Machine Learning Libraries

Scikit-learn was chosen for the project for the following reasons:

- Off-the-shelf algorithm
- Shallow learning curve compared to TensorFlow