DEMOCRATIC AND POPULAR REPUBLIC OF ALGERIA MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
UNIVERSITY ABBES LAGHROUR OF KHENCHELA
FACULTY OF SCIENCE AND TECHNOLOGY

Department of Mathematics and Computer Science

Dissertation

Presented to obtain the Master diploma in Computer Science

Specialty: Security and Web Technologies

Design of an Intelligent Information System: Application for the management of student orientation.

Presented by:

BOUKHIL Mounir
KELLIL Rami

Supervised by:

Dr. HEMAM Sofiane

2021/2022

Appreciation

We would like to thank all the people who contributed to the success of our studies and who helped us during the writing of this Dissertation.

First of all, We would like to thank our supervisor Dr: HEMAM Sofiane for his patience, his availability and above all his wise advice, We thank him for his guidance and assistance.

We would like to thank all the teachers of the Mathematics and Computer Science Department who helped us throughout our cycle of studies.

Abstract

Artificial Intelligence in general and Machine Learning in particular has become a major role in human life, especially in terms of services, where we can take advantage of the speed of computer information processing and the large amount of digital information available to facilitate people's lives in various fields.

In this context, and with the aim of helping the development of the service sector in Algeria, especially in universities, we propose this application, which is based on the decision tree an algorithm of supervised machine learning. The main objective of the application is to help the first year students of the department of mathematics and computer science to choose their specialization for the second year.

The application uses student information from previous years to help new students choose their orientation.

Key Word: Artificial Intelligence, Machine Learning, Decision Tree, Supervised Machine Learning

الملخص

أصبح الذكاء الاصطناعي بشكل عام والتعلم الآلي بشكل خاص يلعب دورًا رئيسيًا في حياة الإنسان ، خاصة من حيث الخدمات ، حيث يمكننا الاستفادة من سرعة معالجة معلومات الكمبيوتر والكمية الكبيرة من المعلومات الرقمية المتاحة لتسهيل حياة الناس في مختلف مجالات

في هذا السياق ، و بهدف المساعدة في تطوير قطاع الخدمات في الجزائر ، وخاصة في الجامعات ، نقترح هذا التطبيق ، الذي يعتمد على شجرة القرار وهي خوارزمية للتعلم الآلي الخاضع للإشراف. الهدف الرئيسي من التطبيق هو مساعدة طلاب السنة الأولى في قسم الرياضيات والاعلام الي على اختيار تخصصهم للسنة الثانية

يستخدم التطبيق معلومات الطلاب من السنوات السابقة لمساعدة الطلاب الجدد في اختيار توجهاتهم.

الكلمات المفتاحية: الذكاء الاصطناعي ، التعلم الآلي ، شجرة القرار ، التعلم الآلي الخاضع للإشراف

Résumé

L'intelligence artificielle en général et l'apprentissage automatique en particulier sont devenus un rôle majeur dans la vie humaine, notamment en termes de services, où nous pouvons tirer parti de la vitesse de traitement des informations informatiques et de la grande quantité d'informations numériques disponibles pour faciliter la vie des gens dans divers domaines.

Dans ce contexte, et dans le but d'aider le développement du secteur des services en Algérie, notamment dans les universités, nous proposons cette application, qui est basée sur l'arbre de décision un algorithme d'apprentissage automatique supervisé. L'objectif principal de l'application est d'aider les étudiants de première année du département de mathématiques et d'informatique à choisir leur spécialisation pour la deuxième année.

L'application utilise les informations sur les étudiants des années précédentes pour aider les nouveaux étudiants à choisir la meilleure orientation.

Mot clé: intelligence artificielle, apprentissage automatique, arbre de décision, apprentissage automatique supervisé

Contents

General Introduction

Since the beginning of humanity, humans have learned about the world around them, their intelligence has enabled them to acquire knowledge. The memory of humans is very limited, compared to that of a computer. To make better use of its knowledge, they must associate a class with this knowledge. Humans use their reasoning to associate knowledge with a class. This classification is the grouping of ideas that makes it possible to distinguish one object from another.

The human brain uses reasoning to better understand knowledge. To establish a reasoning, humans generally use their previous experiences, their instincts and their moral sense.

A computer's ability to learn is limited, it can't really tell the difference between right and wrong, and it has no instincts. It must be based on the experiments that it carries out during a classification. To do this, artificial intelligence researchers use various techniques to associate ideas with a class. To better classify a particular situation, artificial intelligence is often inspired by nature to better interpret knowledge. Several approaches can be used, for example: decision trees, genetic algorithms, clustering.

In this dissertation, we are interested in one of the approaches called decision trees, and we are creating an application based on this approach, the application is devoted to the student of the first-year mathematics and computer science to help him to select the best choice to their orientation towards the second year. This orientation is based on the previous information students who have passed their first year and the second year with success.

We have divided this work into four parts:

- The first chapter we talk about Machine learning and its Categories, Terminology and Notation, Advantages and Disadvantages.
- The second chapter is about decision tree and how it works and its algorithms.
- The third chapter we describe the conception part of the application, Its architecture function and its behavior with user and with its different component and we talk about creation of decision tree
- The last chapter is the implementation part we talk about interfaces and we describe some technical part, we explain programing languages and frame works we used in application.

Chapter I

1 Introduction to Machine Learning

Machine learning is one of the hottest topics nowadays. People talk about machine learning as if it is magic. Organizations are racing to integrate machine learning into their functions. Everyone talks about it, but not too many people know what it really is. It is just math and statistics plus data!

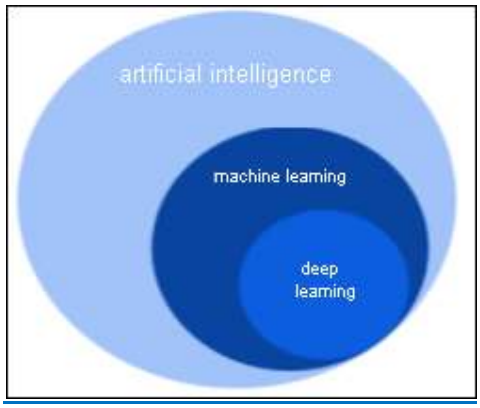

Figure 1: General concepts

Machine learning is an analytical method that automates model building by learning from the available data, identifying patterns and making decisions with minimal human intervention. (Pisani, 2020).

2 Definitions

Before getting into Machine Learning, it is necessary to understand certain concepts used in some articles and books. Important terms are listed below:

- **Training example**: Synonymous to observation, training record, training instance, training sample (in some contexts, sample refers to a collection of training examples).
- **Feature**: Synonymous to predictor, variable, independent variable, input, attribute, covariate.
- **Target**: Synonymous to outcome, ground truth, desired output, response variable, dependent variable, label, class label (in the context of classification).

- **Output**: Synonymous to model output or prediction; here, output means the return value of the model that is to be matched against the target.
- **Sample**: it is a segment of the population
- **Validation set**: set used to correct mistakes learned from the training set.
- **Training set** : set used for learning
- **Test set**: set used to check how well the algorithm has learned
- **Normalization**: set the same scale to all the variables.
- **Objective function**: function which you are trying to minimize/maximize.
- **Cost function**: represents the error for the entire training set
- **Loss function**: computes the error for a single training example. Sometimes cost and loss function are used interchangeably. Because in the end when you are trying to minimize the loss function in each iteration, the overall objective is to minimize the cost function
- **Times series data vs Cross-sectional**: while times series refers to data for which the observations are collected over intervals of times, the observations for cross-sectional data are for a single point in time
- **Machine learning model**: design of a mathematical model that makes predictions or finds patterns from data (Pisani, 2020)

3 Concepts

Concepts related to machine learning that are relevant to identify are deep learning and artificial intelligence. Sometimes people get confused with these terms and use them interchangeably. However, there are relevant differences (Pisani, 2020)

Artificial intelligence: Is a more general concept that consists of the use of machines to perform tasks based on algorithms.

Machine learning: Is a part of AI that focuses on the ability of machines to learn from past data and correct themselves.

Deep Learning: Is a subset of machine learning, where multi-layer neural networks are used to apply pattern matching.

Machine learning is a huge field, which has a variety of algorithms and methods to resolve different problems. Below, we present some of them.

4 Categories and the types of machine learning:

There are different types of machine learning: Supervised learning, Unsupervised learning, Semi-supervised learning and Reinforcement learning. The methods are different on the way the algorithm learns. While some methods need that the data is labeled, others discover patterns in the unlabeled data.(Pisani, 2020)

4.1 Supervised Learning

In supervised learning the main purpose is to predict or classify future data based on past data. Supervised learning is concerned with predicting a target value given input observations. In machine learning, we call the model inputs "features". The target values that supervised models are trained to predict are also often called *labels*. Supervised learning can be categorized into two major subcategories: regression analysis and classification. In regression analysis, the target values or labels are continuous variables (image A figure 2). In classification, the labels are so-called *class labels*, which can be understood as discrete class- or group-membership indicators(image B figure 2) (Raschka, 2020)

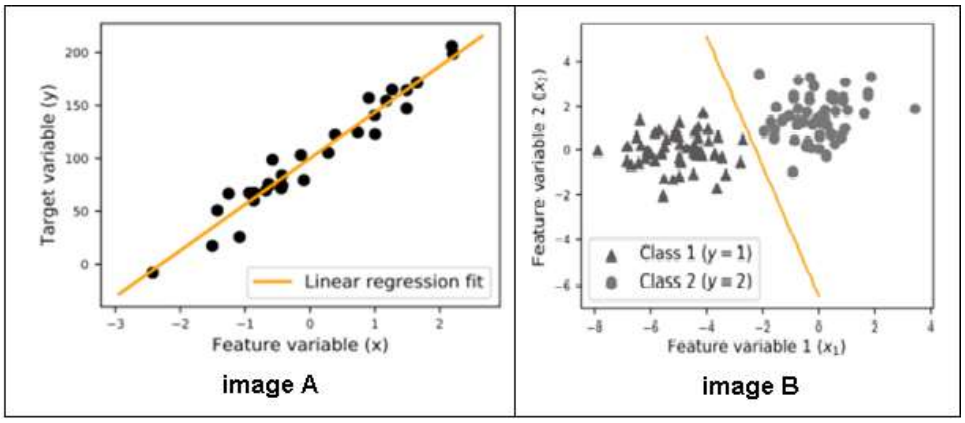

Figure 2: Illustrations of the two main categories of supervised learning, regression (A) and classification (B) (Raschka, 2020)

In machine learning, we often work with high-dimensional datasets; that is, datasets consisting of many input features. However, due to the limitations of the human imagination

and the written medium, conventional illustrations can only depict two (or at most three) spatial dimensions. The 2D scatterplots in Figure 3 show two simple datasets. Here, sub-panel A depicts a simple regression example for a dataset with only a single feature. The target variable, the values we want to predict, is depicted as the y-axis. Sub-panel B depicts a 2-dimensional classification dataset, where the target variable, the discrete class label information, is encoded as a symbol (triangle vs. circle). In both cases, there is a target variable that the model learns to predict. In the case of the linear regression example, the target variable is a continuous variable depicted on the y-axis in (Figure 2) A. For the classification example in (Figure 2) B, the target variable is comprised of class labels depicted as symbols (triangles and circles) (Raschka, 2020).

4.2 Unsupervised Learning

The previous section introduced supervised learning, which is the most prominent subcategory of machine learning. This section discusses the second major category of machine learning: unsupervised learning. In unsupervised learning, in contrast to supervised learning, no labeling information is given. The goal is to discover or model hidden structures in data rather than predicting continuous or discrete target labels.

The major subcategory of unsupervised learning is clustering, which assigns group membership information to data points(Figure 3). It can be thought of as a task similar to classification but without labeling information given in the training dataset. Hence, in the absence of class label information, the clustering approach is to group data records by similarity and define distinct groups based on similarity thresholds.

Clustering can be divided into three major groups: prototype-based, density-based, and hierarchical clustering. In prototype-based clustering algorithms, such as K-Meansthenumber of cluster centers is defined (the cluster centers are repositioned iteratively), and data points are assigned to the closest prototype based on a pair-wise distance measure (for example, Euclidean distance). In density-based clustering, unlike in prototype-based clustering, the number of cluster centers is not fixed but assigned by identifying regions of high density (the location of many data records close to each other measured by a user-defined distance metric. In hierarchical clustering, a distance metric is used to group examples in a tree like-fashion, where examples at the root are more related to each other. The depth of the tree defines the number of clusters (Raschka, 2020).

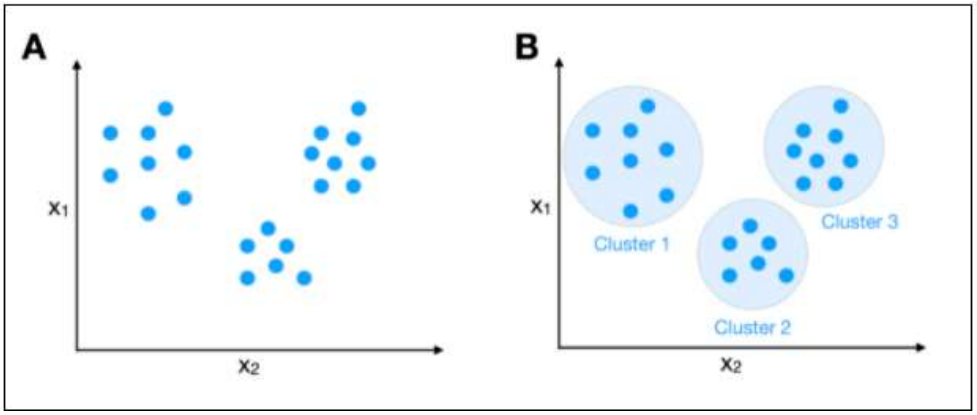

Figure 3: Illustration of clustering. (A) A two-dimensional, unlabeled dataset. (B) Clusters inferred by a clustering algorithm that groups similar points into the same cluster (Raschka, 2020).

4.3 Reinforcement Learning

The third subcategory of machine learning is reinforcement learning (figure4). In contrast to supervised learning, which focuses on predicting a specific outcome, reinforcement learning is concerned with learning a *series of actions* that lead to a particular outcome. To illustrate reinforcement learning in the context of (Figure4), (1) given a chess board state StStand some reward value RtRt at iteration tt, (2) the reinforcement learning agent selects an action AtAt that moves one of the pawns by two fields. (3) Next, the environment considers AtAt to produce the next state, St+1St+1 and the corresponding reward for performing the action, Rt+1Rt+1. This cycle repeats until the end of the episode (Raschka, 2020).

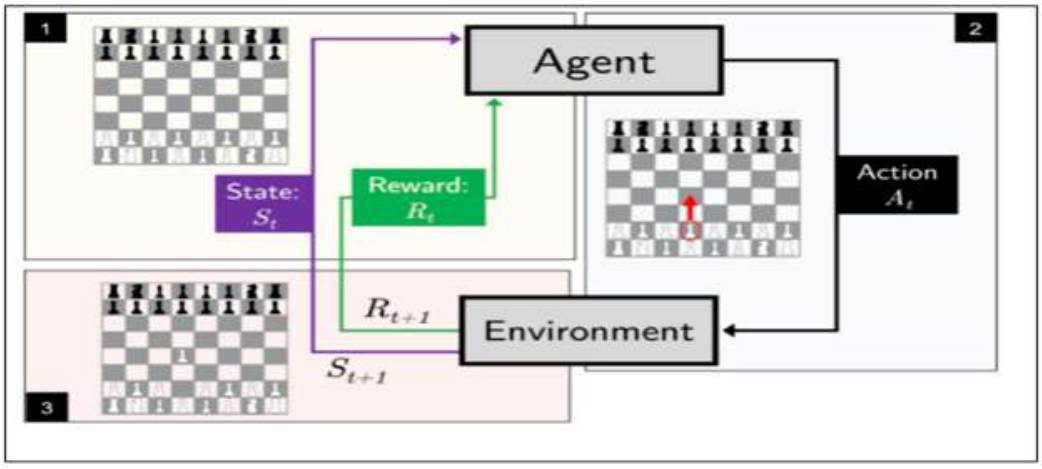

Figure 4: Illustration of reinforcement learning using a chess board as an example (Raschka, 2020)

4.4 Semi-supervised Learning

Semi-supervised[5] learning is a category mix between supervised and unsupervised learning. Semi-supervised learning refers to scenarios where some training examples are labeled while others are not. The main idea behind semi-supervised learning is to use the labeled portion of the dataset (via supervised learning) to label the unlabeled portion, which can then be used for supervised learning. Here, the use of unlabeled examples can improve generalization performance, as more data is available during training.(Raschka, 2020)

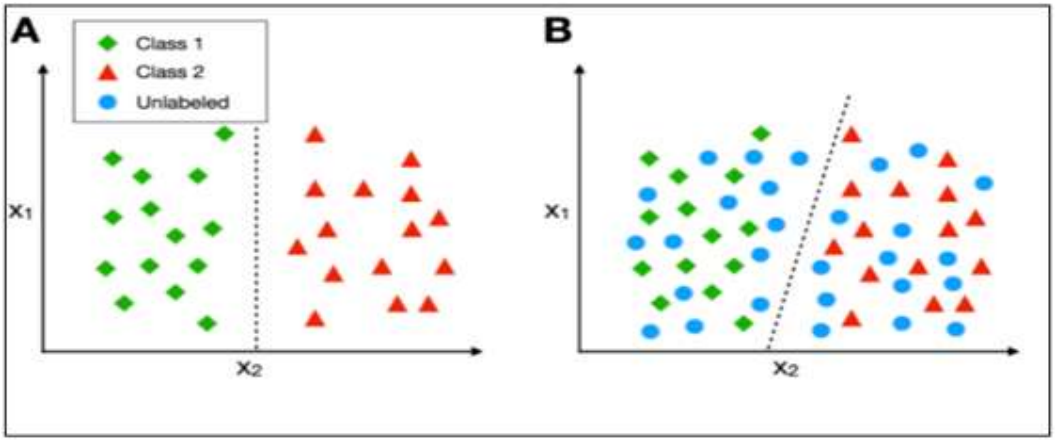

Figure 5: Illustration for semi-supervised learning incorporating unlabeled examples. (A) A decision boundary derived from the labeled training examples only. (B) A decision boundary based on both labeled and unlabeled examples (Raschka, 2020).

4.5 Self-supervised Learning

Collecting labels for large datasets can be prohibitively expensive. Self-supervised learning aims to leverage large amounts of unlabeled data for supervised learning. Due to its ability to use naturally available information as labels for supervised learning, self-supervised learning is sometimes also described as *autonomous* supervised learning. Both supervised learning and self-supervised learning are closely related since both learn mappings from inputs (for example, Figure6) to outputs (for example, class labels) from labeled inputs-output pairs. What distinguishes self-supervised learning from regular supervised learning is that in the former, the labels are automatically generated or naturally embedded in the data.

Using self-supervised learning, a model (for supervised learning on a target task) can be (pre-)trained on a related task first, before it is trained on the target task. This allows leveraging more data. The related task is also often known as a "pretext task." For image classification, common pretext tasks include predicting by how many degrees an image was rotated

colorizing grayscale versions of an image and reassembling an image that has been divided into subregions or predicting the context of a random image patch (figure 6).

In practice, self-supervised learning is often used when the labeled dataset corresponding target task is relatively small, or the model could benefit from additional data. In this scenario, one could leverage a more extensive, unlabeled dataset of a similar type as the data for the target task (for example, images with the same resolution) and create the labels for the pretext task, as described in the previous paragraph. After pre-training the model (typically a neural network) on the pretext labels, it can be trained on the target-task dataset. In practice, pre-training a model on the pretext task can result in better model performance than training the model on the target-task dataset alone.(Raschka, 2020)

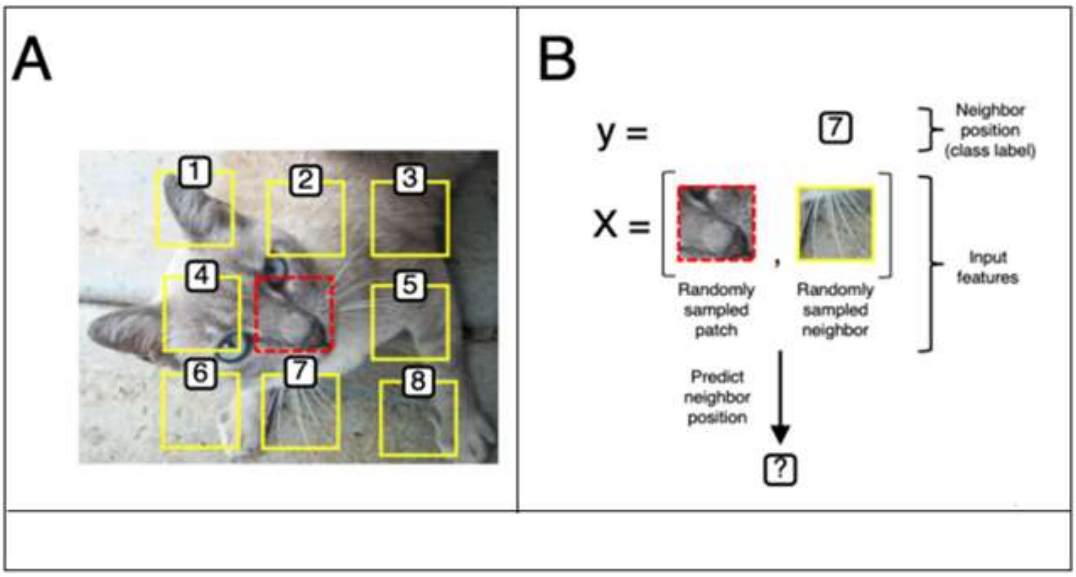

Figure 6:*Self-supervised learning via context prediction. (A) A random patch is sampled (red square) along with 9 neighboring patches. (B) Given the random patch and a random neighbor patch, the task is to predict the position of the neighboring patch relative to the center patch (red square).*

5 Machine Learning Terminology and Notation

5.1 Predictive Modeling

For the most part, supervised learning is focused on developing predictive models; that is, training classifiers or regression models to predict target information (for instance, class labels) of new observations. The flowchart in (Figure 7) summarizes a typical predictive modeling workflow)

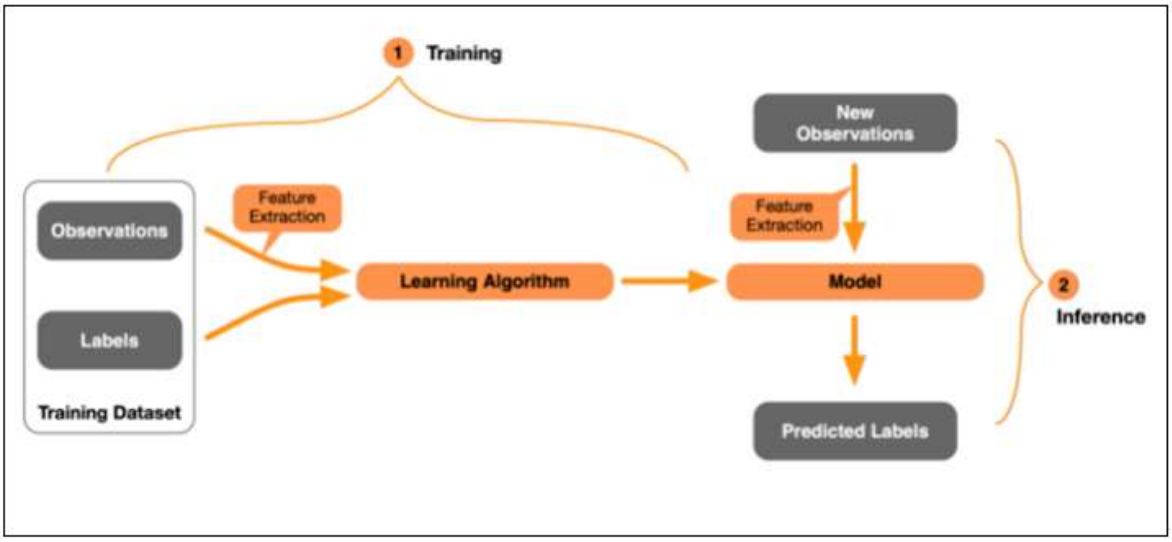

Figure 7: Illustration of a supervised learning and predictive modeling pipeline (Raschka, 2020)

In brief, the supervised learning workflow depicted in (Figure 7) can be structured into two main steps: (1) model training and (2) inference. Training involves a labeled training dataset and a machine or deep learning algorithm. The learning algorithm learns how to associate the observations (also called features) in the training dataset – for example, images of flowers – with label information, such as thenames of the flower species. In particular, the learning algorithm will use the training dataset to create or parameterize a predictive model that can then be used to make predictions on new observations.

Note that certain learning algorithms work on so-called “raw features” whereas other classes of learning algorithms operate on preprocessed or extracted features, which is discussed in more detail in the next subsection.

Using a model to predict the target information of new observation is nowadays also called “inference” among deep learning researchers and practitioners. In practice, before we apply the model in the real world, we typically involve a model evaluation step. This is similar to the “inference” stage shown in figure 8, but the new observations come from an independent test dataset for which we know the true labels that we want to predict. This basic model evaluation is illustrated on (Figure8): We apply the model to the new observations (data records from the test dataset) and then compare the predicted labels to the actual labels from the test dataset and compute an evaluation metric such as prediction error or accuracy. Note that model evaluation is a broad topic in itself that is out of the scope of this book. However, if you are interested in further details, I recommend the freely available article “Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning” (Raschka, 2020)

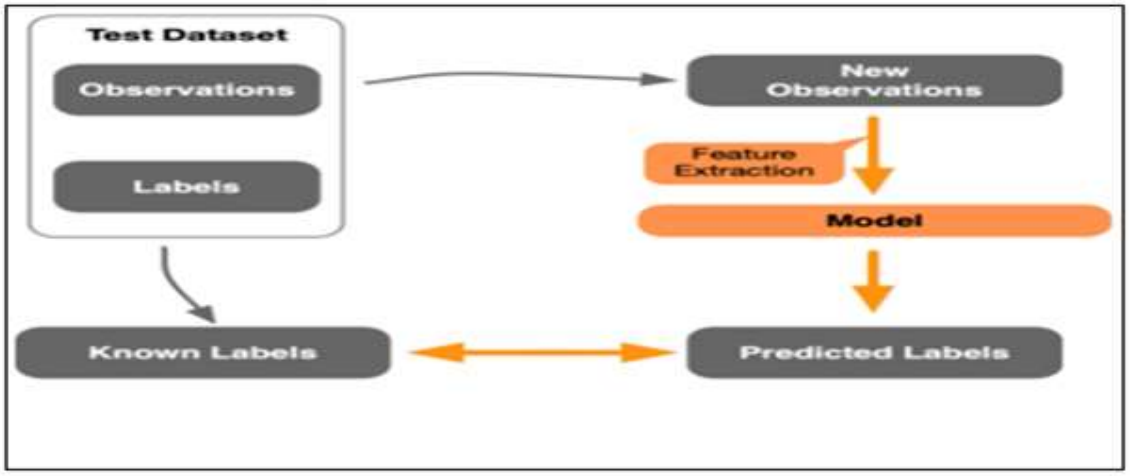

Figure 8: Using a test dataset to evaluate the performance of a predictive model (Raschka, 2020).

5.2 Data Representation

In the context of feature representation used in conventional machine learning versus deep learning, it is helpful to think about the concepts of *structured* and *unstructured* data (Figure 7).

In simple words, structured data can be understood as tabular data. A characteristic of structured data is that it usually has undergone preprocessing, including feature extraction. For example, structured data is typically stored in the form of database entries, spreadsheets, or CSV files. In machine learning, it is common to format structured data such that each row of the tabular dataset represents a data instance or record (for example, a training example). Each column represents an observed or extracted feature. In statistics, this type of data representation is also often referred to as *design matrix*. (Figure 7)illustrates an example of a structured dataset containing iris flower leaf measurements (sepal length, sepal width, petal length, and petal width). Here, the first column is simply a dataset index. Columns 2-5 are the flower measurements (features) and comprise the design matrix component of the dataset. The last column contains the class label (here, the flower species: Iris-setosa, Iris-versicolor, and Iris-virginica) associated with each flower (training example). In a supervised learning setting, we could train a classifier to predict the class label (last column) from the features (columns 2-5).

Unstructured data described data closer to its *raw* form – the form that it was collected. For example, in the context of the iris flower example described in the previous paragraph, an unstructured data format would be a collection of images corresponding to the flowers listed in the structured dataset table shown in (A in the image I). It is then easy to see how features

can be extracted from an unstructured data record (**B in the figure 7**) to create a structured data record (**a row in A in the Figure 7**).

While machine learning is traditionally designed to work with structured data, deep learning has been developed with unstructured data in mind. Most deep learning architectures, such as convolutional and recurrent neural networks covered later in this book, are designed to operate on image and text data. As part of the learning process, deep neural networks learn to extract and transform the raw (unstructured) data such that it is conducive for predictive modeling (in supervised learning settings). For this reason, deep learning is also often described as *representation learning*.(Raschka, 2020)

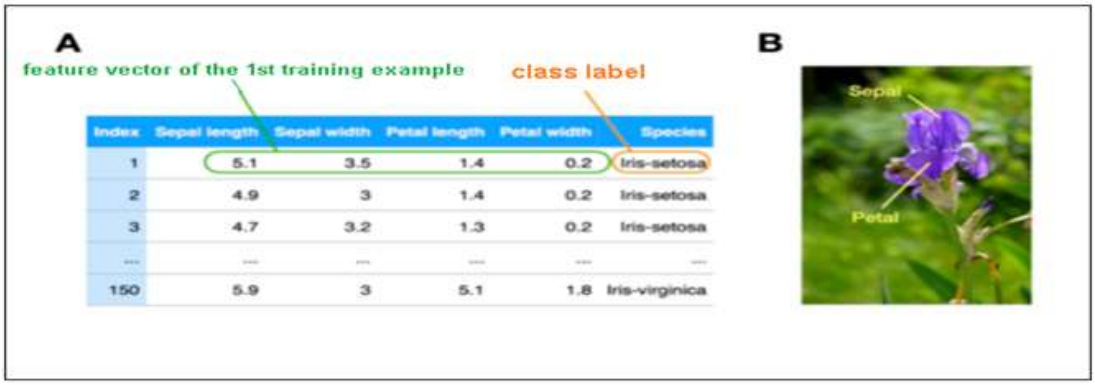

Figure 9: *Structured versus unstructured data. (A) A structured, tabular dataset for supervised learning where each row represents a training example (here: iris flower) associated with four measurements and a class label (iris species). (B) A photograph of an iris flower (unstructured data). Such photographs can be processed into a structured dataset shown in (A)* (Raschka, 2020).

6 Advantages and Disadvantages of Machine Learning Language

Every coin has two faces, each face has its own property and features. It's time to uncover the faces of ML. A very powerful tool that holds the potential to revolutionize the way things work.

6.1 Advantages of Machine learning

6.1.1 Easily Identifies Trends and Patterns

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them. Do you know the Applications of Machine Learning?

6.1.2 No Human Intervention Needed (Automation)

With ML, you don't need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

6.1.3 Continuous Improvement

As ML algorithms gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

6.1.4 Handling Multi-Dimensional and Multi-Variety Data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

6.2 Disadvantages of Machine Learning

With all those advantages to its powerfulness and popularity, Machine Learning isn't perfect. The following factors serve to limit it:

6.2.1 Data Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

6.2.2 Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

Also, see the future of Machine Learning

6.2.3 Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

6.2.4 High Error-Susceptibility

Machine Learning is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it. (data-flair.training/blogs/advantages-and-disadvantages-of-machine-learning/).

7 Conclusion

In this chapter, we have introduce the machine learning and its category such as supervised, unsupervised and reinforcement learning and where we could use them. Through this chapter, we could understand how the predictive modeling work and how the data be represented in the machine learning and finally the Advantages and Disadvantages of Machine Learning Language are presented.

Chapter II

1 Introduction

A decision tree is a type of supervised machine learning used to categorize or make predictions based on how a previous set of questions were answered(dataset). A Decision tree is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

2 Decision Tree Behavior

To illustrate how classification with a decision tree works we consider a simple version of the vertebrate classification problem which is detect if the animal is mammals or not:

So we Suppose a new species is discovered by scientists. How can we tell whether it is a mammal or a non-mammal? One approach is to pose a series of questions about the characteristics of the species. The first question we may ask is whether the species has cold or warm blood. If it has cold blood, then it is definitely not a mammal. Otherwise, it is either a bird or a mammal. In the latter case, we need to ask a follow-up question: Do the females of the species give birth to their young? Those that do give birth are definitely mammals, while those that do not are likely to be non-mammals.

The previous example illustrates how we can solve a classification problem by asking a series of carefully crafted questions about the attributes of the test record. Each time we receive an answer, a follow-up question is asked until we reach a conclusion about the class label of the record. The series of questions and their possible answers can be organized in the form of a decision tree, which is a hierarchical structure consisting of nodes and directed edges.

The (Figure10) shows the decision tree for the mammal classification problem. The tree has three types of nodes:

-A root node: that has no incoming edges and zero or more outgoing edges.

-Internal nodes: each of which has exactly one incoming edge and two or more outgoing edges.

-Leaf or terminal nodes: each of which has exactly one incoming edge and no outgoing edges.

In a decision tree, each leaf node is assigned a class label. The nonterminal nodes, which include the root and other internal nodes, contain attribute test conditions to separate records that have different characteristics. For example, the root node shown in image1 uses the attribute Body

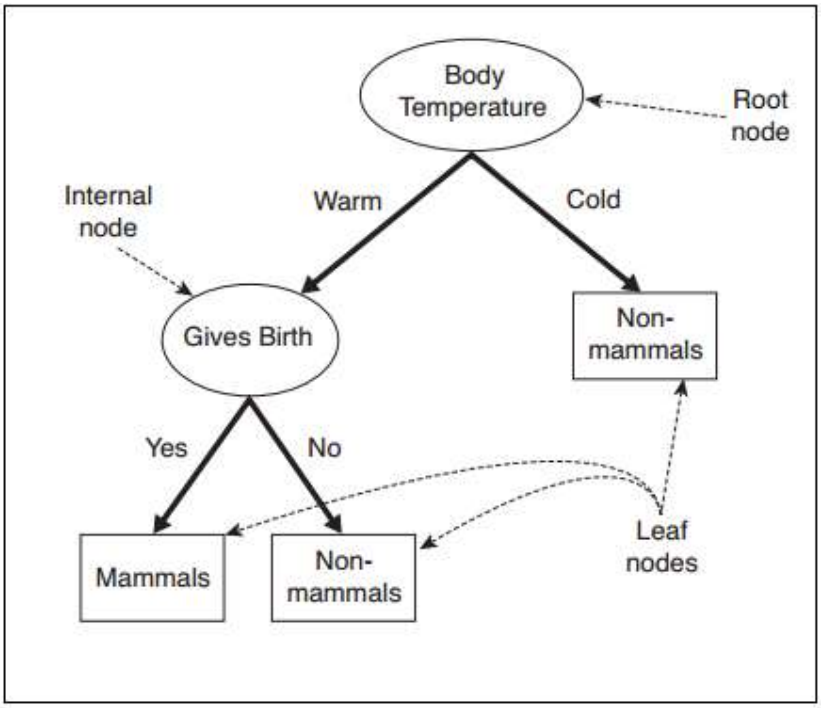

Figure 10: Decision tree for the mammal classification problem (L. Breiman, 1984)

Temperature to separate warm-blooded from cold-blooded vertebrates. Since all cold-blooded vertebrates are non-mammals, a leaf node labeled Non-mammals is created as the right child of the root node. If the vertebrate has a warm blood, a subsequent attribute, Gives Birth, is used to distinguish mammals from other warm-blooded creatures, which are mostly birds. Classifying a test record is straightforward once a decision tree has been constructed. Starting from the root node, we apply the test condition to the record and follow the appropriate branch based on the outcome of the test. This will lead us either to another internal node, for which a new test condition is applied, or to a leaf node. The class label associated with the leaf node is then assigned to the record. As an illustration, (Figure 11) traces the path in the decision tree that is used to predict the class label of a flamingo. The path terminates at a leaf node labeled Non-mammals.

(L. Breiman, 1984)

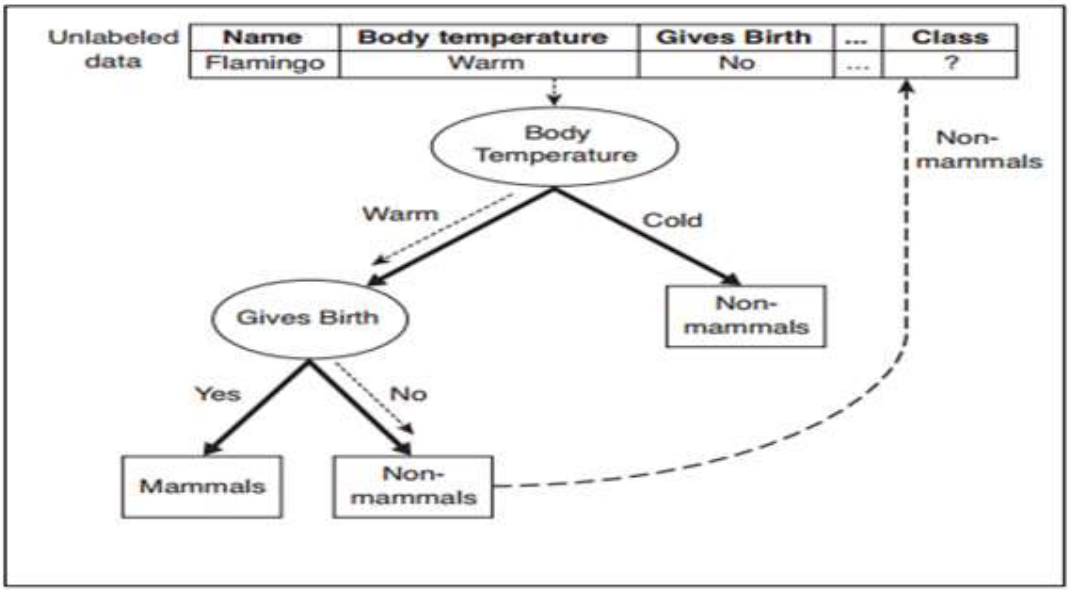

Figure 11: Classifying an unlabeled vertebrate. The dashed lines represent the outcomes of applyingvarious attribute test conditions on the unlabeled vertebrate. The vertebrate is eventually assigned totheNon-mammalclass (L. Breiman, 1984)

3 How to Build a Decision Tree

In principle, there are exponentially many decision trees that can be constructed from a given set of attributes. While some of the trees are more accurate than others, finding the optimal tree is computationally infeasible because of the exponential size of the search space. Nevertheless, efficient algorithms have been developed to induce a reasonably accurate, albeit suboptimal, decision tree in a reasonable amount of time. These algorithms usually employ a greedy strategy that grows a decision tree by making a series of locally optimum decisions about which attribute to use for partitioning the data. One such algorithm is Hunt's algorithm, which is the basis of many existing decision tree induction algorithms, including ID3, C4.5, and CART (Pang-Ning Tan, 2021)

3.1 Hunt's Algorithm

In Hunt's algorithm, a decision tree is grown in a recursive fashion by partitioning the training records into successively purer subsets. Let Dt be the set of training records that are associated with node t and y = {y1, y2,...,yc} be the class labels. The following is a recursive definition of Hunt's algorithm.

Step 1: If all the records in Dt belong to the same class yt, then t is a leaf node labeled as yt.

Step 2: If Dt contains records that belong to more than one class, an attribute test condition is selected to partition the records into smaller subsets. A child node is created for each outcome of the test condition and the records in Dt are distributed to the children based on the outcomes. The algorithm is then recursively applied to each child node.(Pang-Ning Tan, 2021)

	binary	categorical	continuous	class
Tid	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Figure 12: Training set for predicting borrowers who will default on loan payments (L. Breiman, 1984).

To illustrate how the algorithm works, consider the problem of predicting whether a loan applicant will repay her loan obligations or become delinquent, subsequently defaulting on her loan. A training set for this problem can be constructed by examining the records of previous borrowers. In the example shown in (Figure 12), each record contains the personal information of a borrower along with a class label indicating whether the borrower has defaulted on loan payments.

The initial tree for the classification problem contains a single node with class label Defaulted = No (see Figure 13(a)), which means that most of the borrowers successfully repaid their loans. The tree, however, needs to be refined since the root node contains records from both classes. The records are subsequently divided into smaller subsets based on the outcomes of the Home Owner test condition, as shown in (Figure13(b)). The justification for choosing this attribute test condition will be discussed later. For now, we will assume that this is the best criterion for splitting the data at this point. Hunt's algorithm is then applied recursively to each child of the root node. From the training set given in (Figure 12), notice that all

borrowers who are home owners successfully repaid their loans. The left child of the root is therefore a leaf node labeled Defaulted = No (see Figure13 (b)). For the right child, we need to continue applying the recursive step of Hunt's algorithm until all the records belong to the same class. The trees resulting from each recursive step are shown in (Figure 13) (c) and (d).

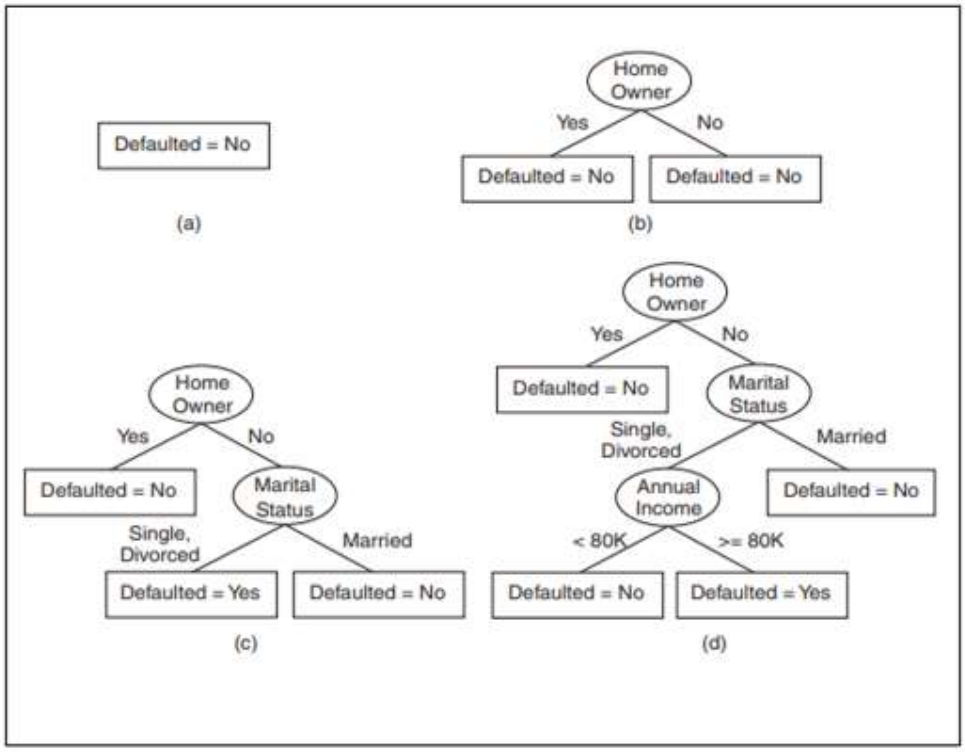

Figure 13:Hunt's algorithm for inducing decision trees (L. Breiman, 1984)

Hunt's algorithm will work if every combination of attribute values is present in the training data and each combination has a unique class label. These assumptions are too stringent for use in most practical situations. Additional conditions are needed to handle the following cases:

1. It is possible for some of the child nodes created in Step 2 to be empty; i.e., there are no records associated with these nodes. This can happen if none of the training records have the combination of attribute values associated with such nodes. In this case the node is declared a leaf node with the same class label as the majority class of training records associated with its parent node.

2. In Step 2, if all the records associated with Dt have identical attribute values (except for the class label), then it is not possible to split these records any further. In this case, the node is declared a leaf node with the same class label as the majority class of training records associated with this node (Pang-Ning Tan, 2021)

4 Design Issues of Decision Tree

A learning algorithm for inducing decision trees must address the following two issues.

1.How should the training records be spliteach recursive step of the tree-growing process must select an attribute test condition to divide the records into smaller subsets. To implement this step, the algorithm must provide a method for specifying the test condition for different attribute types as well as an objective measure for evaluatingthe goodness of each test condition.

2.How should the splitting procedure stop.A stopping condition is needed to terminate the tree-growing process. A possible strategy is to continue expanding a node until either all the records belong to the same class or all the records have identical attribute values. Although both conditions are sufficient to stop any decision tree induction algorithm.(Pang-Ning Tan, 2021)

5 Methods to Express Attribute Test Conditions

Decision tree induction algorithms must provide a method for expressing an attribute test condition and its corresponding outcomes for different attribute types.

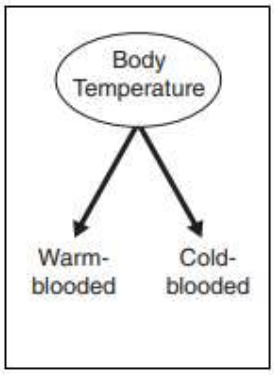

Figure 14: Test condition for binary attributes (L. Breiman, 1984)

Binary Attributes The test condition for a binary attribute generates two potential outcomes, as shown in (Figure 14)

Nominal Attributes Since a nominal attribute can have many values, its test condition can be expressed in two ways, as shown in (Figure 15). For a multiway split (Figure15) (a), the number of outcomes depends on the number of distinct values for the corresponding attribute. For example, if an attribute such as marital status has three distinct values -single, married, or divorced-its test condition will produce a three-way split. On the other hand, some decision tree algorithms, such as CART, produce only binary splits by considering all $2^{k-1} - 1$ ways of creating a binary partition of k attribute values. (Figure15) (b) illustrates three different ways of grouping the attribute values for marital status into two subsets.

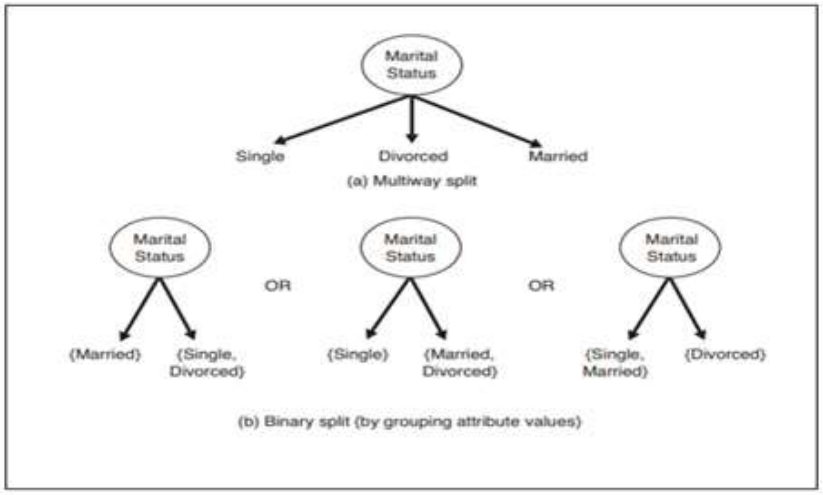

Figure 15: Test conditions for nominal attributes (L. Breiman, 1984)

Ordinal Attributes Ordinal attributes can also produce binary or multiway splits. Ordinal attribute values can be grouped as long as the grouping does not violate the order property of the attribute values. Figure 16 illustrates various ways of splitting training records based on the Shirt Size attribute. The groupings shown in (Figure 16) (a) and (b) preserve the order among the attribute values, whereas the grouping shown in (Figure 16) (c) violates this property because it combines the attribute values Small and Large into the same partition while Medium and Extra Large are combined into another partition.

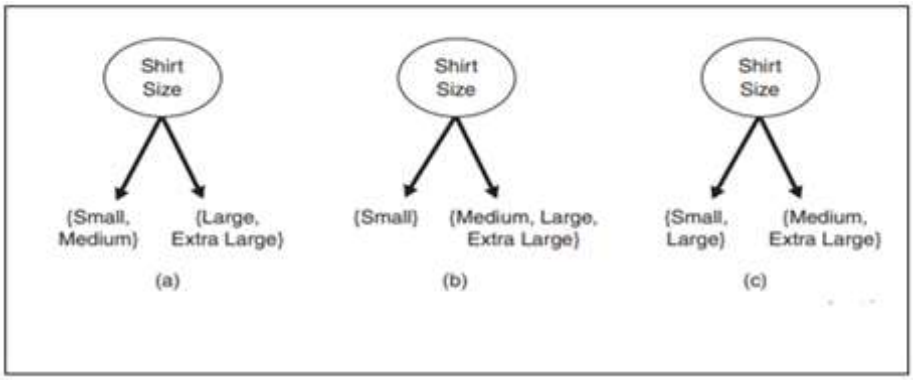

Figure 16: Different ways of grouping ordinal attribute values (L. Breiman, 1984)

Continuous Attributes For continuous attributes, the test condition can be expressed as a comparison test **(A<v)**or**(A≥v)** with binary outcomes, or a range query with outcomes of the form **vi ≤A<vi+1, for i=1,...,k**. The difference between these approaches is shown in (Figure17). For the binary case, the decision tree algorithm must consider all possible split positions, and it selects the one that produces the best partition.(Pang-Ning Tan, 2021)

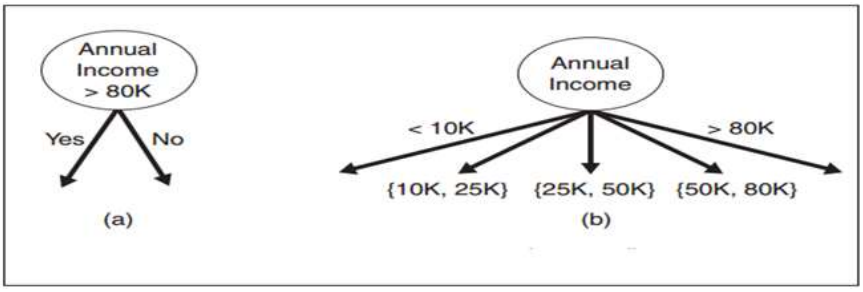

Figure 17: Test condition for continuous attributes (L. Breiman, 1984)

6 Measures for Selecting the Best Split

There are many measures that can be used to determine the best way to split the records. These measures are defined in terms of the class distribution of the records before and after splitting. Let **p(i|t)** denote the fraction of records belonging to class **i** at a given node **t**. We sometimes omit the reference to node **t** and express the fraction as **pi**. In a two-class problem, the class distribution at any node can be written as **(p0, p1)**, where **p1 = 1 − p0**. To illustrate, consider the test conditions shown in (Figure 18). The class distribution before splitting is **(0.5, 0.5)** because there are an equal number of records from each class. If we split the data using the Gender attribute, then the class distributions of the child nodes are **(0.6, 0.4)** and

(0.4, 0.6), respectively. Although the classes are no longer evenly distributed, the child nodes still contain records from both classes. Splitting on the second attribute, Car Type, will result in purer partitions. The measures developed for selecting the best split are often based on the degree of impurity of the child nodes. The smaller the degree of impurity, the more skewed the class distribution. For example, a node with class distribution **(0, 1)** has zero impurity, whereas a node with uniform class distribution **(0.5, 0.5)** has the highest impurity.

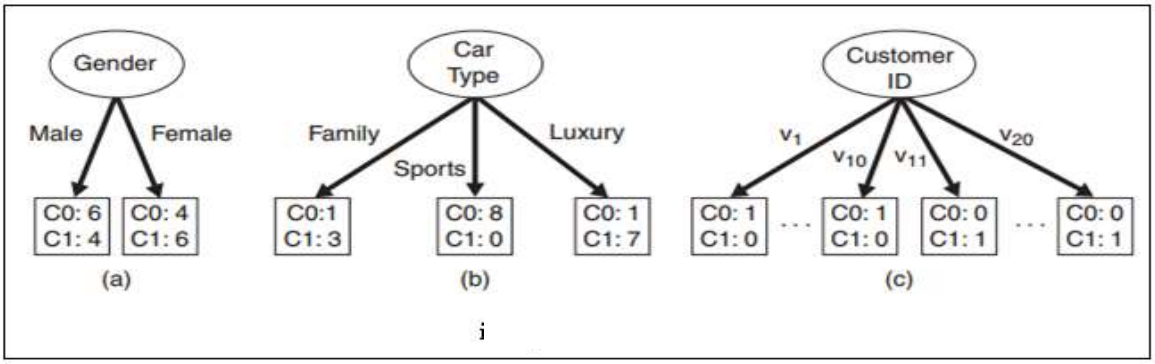

Figure 18: Multiway versus binary splits (L. Breiman, 1984)

Examples of impurity measures include:

$$Entropy(t) = -\sum_{i=0}^{c-1} p(i|t) \log_2 p(i|t)$$

$$Gini(t) = 1 - \sum_{i=0}^{c-1} [p(i|t)]^2$$

$$Classigication\ error(t) = 1 - {}^{max}_{i}[p(i|t)]$$

where c is the number of classes and 0 log2 0 = 0 in entropy calculations.

On the (Figure19) we have comparison for the values of the impurity measures for binary classification problems. **p** refers to the fraction of records that belong to one of the two classes. Observe that all three measures attain their maximum value when the class distribution is uniform **(i.e., when p = 0.5).** The minimum values for the measures are attained when all the records belong to the same class **(i.e., when p equals 0 or 1)**.

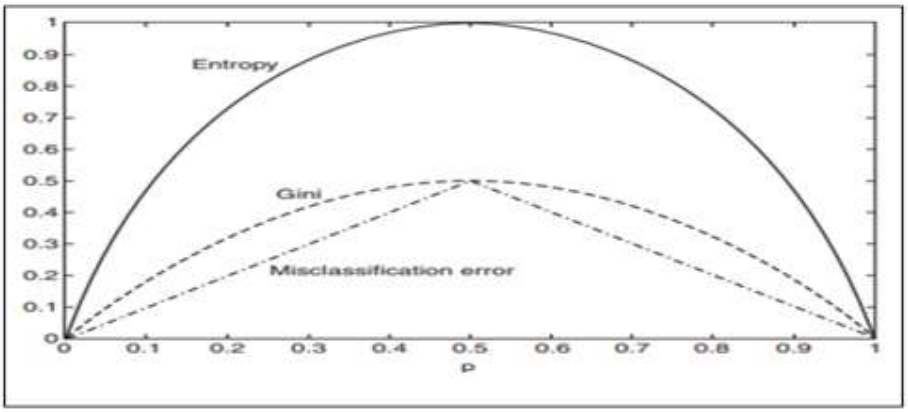

Figure 19: Comparison among the impurity (L. Breiman, 1984)

We next provide several examples of computing the different impurity measures.

Node N_1	Count
Class=0	0
Class=1	6

$$Gini = 1 - \left(\frac{0}{6}\right)^2 - \left(\frac{6}{6}\right)^2 = 0$$

$$Entropy = -\left(\frac{0}{6}\right) log_2 \left(\frac{0}{6}\right) - \left(\frac{6}{6}\right) log_2 \left(\frac{6}{6}\right) = 0$$

$$Error = 1 - \max[\frac{0}{6}, \frac{6}{6}] = 0$$

Node N_1	Count
Class=0	1
Class=1	5

$$Gini = 1 - \left(\frac{1}{6}\right)^2 - \left(\frac{5}{6}\right)^2 = 0.278$$

$$Entropy = -\left(\frac{1}{6}\right) log_2 \left(\frac{1}{6}\right) - \left(\frac{5}{6}\right) log_2 \left(\frac{5}{6}\right) = 0.650$$

$$Error = 1 - \max[\frac{1}{6}, \frac{5}{6}] = 0.167$$

Node N_1	Count
Class=0	3
Class=1	3

$$Gini = 1 - \left(\frac{3}{6}\right)^2 - \left(\frac{3}{6}\right)^2 = 0.5$$

$$Entropy = -\left(\frac{3}{6}\right) log_2 \left(\frac{3}{6}\right) - \left(\frac{3}{6}\right) log_2 \left(\frac{3}{6}\right) = 1$$

$$Error = 1 - \max[\frac{3}{6}, \frac{3}{6}] = 0.5$$

The preceding examples, along with image10, illustrate the consistency among different impurity measures. Based on these calculations, node N1 has the lowest impurity value, followed by N2 and N3. Despite their consistency, the attribute chosen as the test condition may vary depending on the choice of impurity measure

To determine how well a test condition performs, we need to compare the degree of impurity of the parent node (before splitting) with the degree of impurity of the child nodes (after splitting). The larger their difference, the better the test condition. The gain, Δ, is a criterion that can be used to determine the goodness of a split:

$$\Delta = \text{I(parent)} - \sum_{j=1}^{k} \frac{N(u_j)}{N} I(u_j)$$

where I(·) is the impurity measure of a given node, N is the total number of records at the parent node, k is the number of attribute values, and N(vj) is the number of records associated with the child node, vj . Decision tree induction algorithms often choose a test condition that maximizes the gain Δ. Since I(parent) is the same for all test conditions, maximizing the gain is equivalent to minimizing the weighted average impurity measures of the child nodes. Finally, when entropy is used as the impurity measure in the Equation, the difference in entropy is known as the information gain, Δinfo.(Pang-Ning Tan, 2021)

7 Gain Ratio

Impurity measures such as entropy and Gini index tend to favor attributes that have a large number of distinct values(Figure 18). Comparing the first test condition, Gender, with the second, Car Type, it is easy to see that Car Type seems to provide a better way of splitting the data since it produces purer descendent nodes. However, if we compare both conditions with Customer ID, the latter appears to produce purer partitions. Yet Customer ID is not a predictive attribute because its value is unique for each record. Even in a less extreme situation, a test condition that results in a large number of outcomes may not be desirable because the number of records associated with each partition is too small to enable us to make any reliable predictions.

There are two strategies for overcoming this problem. The first strategy is to restrict the test conditions to binary splits only. This strategy is employed by decision tree algorithms such as CART. Another strategy is to modify the splitting criterion to take into account the number of outcomes produced by the attribute test condition. For example, in the C4.5 decision tree algorithm, a splitting criterion known as gain ratio is used to determine the goodness of a split. This criterion is defined as follows:

$$Gain\ ratio = \frac{\Delta_{info}}{Split\ Info}$$

Here $-\sum_{i=1}^{k} p(vi)log2p(vi)$ and k is the total number of splits. For example, if each attribute value has the same number of records, then **∀i : P(vi)=1/k** and the split information would be equal to log2 k. This example suggests that if an attribute produces a large number of splits, its split information will also be large, which in turn reduces its gain ratio.(Pang-Ning Tan, 2021)

8 Characteristics of Decision Tree

The following is a summary of the important characteristics of decision tree induction algorithms:

1. Decision tree induction is a nonparametric approach for building classification models. In other words, it does not require any prior assumptions regarding the type of probability distributions satisfied by the class and other attributes

2. Finding an optimal decision tree is an NP-complete problem. Many decision tree algorithms employ a heuristic-based approach to guide their search in the vast hypothesis space.

3. Techniques developed for constructing decision trees are computationally inexpensive, making it possible to quickly construct models even when the training set size is very large. Furthermore, once a decision tree has been built, classifying a test record is extremely fast, with a worst-case complexity of O(w), where w is the maximum depth of the tree.

4. Decision trees, especially smaller-sized trees, are relatively easy to interpret. The accuracies of the trees are also comparable to other classification techniques for many simple data sets.

5. Decision trees provide an expressive representation for learning discretevalued functions. However, they do not generalize well to certain types of Boolean problems. One notable example is the parity function, whose value is 0 (1) when there is an odd (even) number of Boolean attributes with the value T rue. Accurate modeling of such a function requires a full decision tree with 2d nodes, where d is the number of Boolean attributes .

6. Decision tree algorithms are quite robust to the presence of noise, especially when methods for avoiding overfitting

7. The presence of redundant attributes does not adversely affect the accuracy of decision trees. An attribute is redundant if it is strongly correlated with another attribute in the data. One of the two redundant attributes will not be used for splitting once the other attribute has been chosen. However, if the data set contains many irrelevant attributes, i.e., attributes that are not useful for the classification task, then some of the irrelevant attributes may be accidently chosen during the tree-growing process, which results in a decision tree that is larger than necessary. Feature selection techniques can help to improve the accuracy of decision trees by eliminating the irrelevant attributes during preprocessing.(Pang-Ning Tan, 2021)

9 Conclusion

On this chapter we have introduce the decision tree, how its work and how we could build one with the best accuracy using the gain ratio and finally we understand the Characteristics of Decision Tree and why we should use it.

Chapter III

1 Introduction

Before we dive into the details let's define predictive analytics first. Popular definitions define predictive analytics as a set of advanced analytics to make predictions about the future. This can be achieved when using high-quality historical data, combined with statistical modeling, data mining, and machine learning techniques and algorithm. Data mining, text analytics, and statistics combined help to create patterns and relationships from structured and unstructured datasets.

2 Some Concepts that need to be clarified

2.1 Why predictive analytics matter

Organizations collect big chunks of data and this data is now put to action. Valuable data becomes increasingly important as historical data. What happened in the past, what is happening now, and what will happen in the future are the main stages for predictive analytics. Companies drive actions based on (the predicted) outcome of future trends. When these actions are linked to your strategy, it is clear that this plays a significant role in your work.

The end result is to identify **risks and opportunities** to which you need to react to. Being on the negative side (risks) help to predict when things might go wrong in a certain situation. For example: math and computer science student who should chose one speciality math or computer science if he have a wrong chose he may not succeed in his studies and can't pass so he need a good prediction to help him avoid that risk

2.2 The main phases

Simply speaking predictive analytics is based on three sequential phases. Start with valuable and reliable quality data in mind and progress through these phases:

- Reporting/analytics: answers the questions like what happened in the past and why did that happen at all. This phase acts as the foundation for historical data.

- Monitoring: collect real-time information about what is happening right now. Seen from a future point in time, this constantly feeds the historical data.
- Predictive Analytics: what will happen in the future and what will be the impact on your work or objective

2.3 The Common processes

Typical processes to execute within the concept of predictive analytics are the following:

Collection of data: mine the data which you need.

Analyze the data: process and clean up the data, aggregate, inspect where needed. Conduct these activities to extract useful information out of the immense collection of raw data. This information is a core asset to draw conclusions later on in the process.

Predictive models: one of the most important steps to predict the future using the collected data.

2.4 The project and the predective analytics

Our project is a web application that offer help for student to have the best speciality choice for them based on their marks using the predictive analytics

We passed by three predictive analytics process to make our application:

Collection the data we need: we collect the data from the past student(their mark and if they manage to pass or not)

Treating and analyzing the data: we divided the marks on two parts a part related to math and part related to computer science to divide later the students on wich good speciality for them based on what part they are better in

Predictive models: we finally create the predictive model that could make good prediction using a supervised learning algorithm from the machine learning called the decision tree.

3 UML Diagrams

3.1 Use Case Diagram

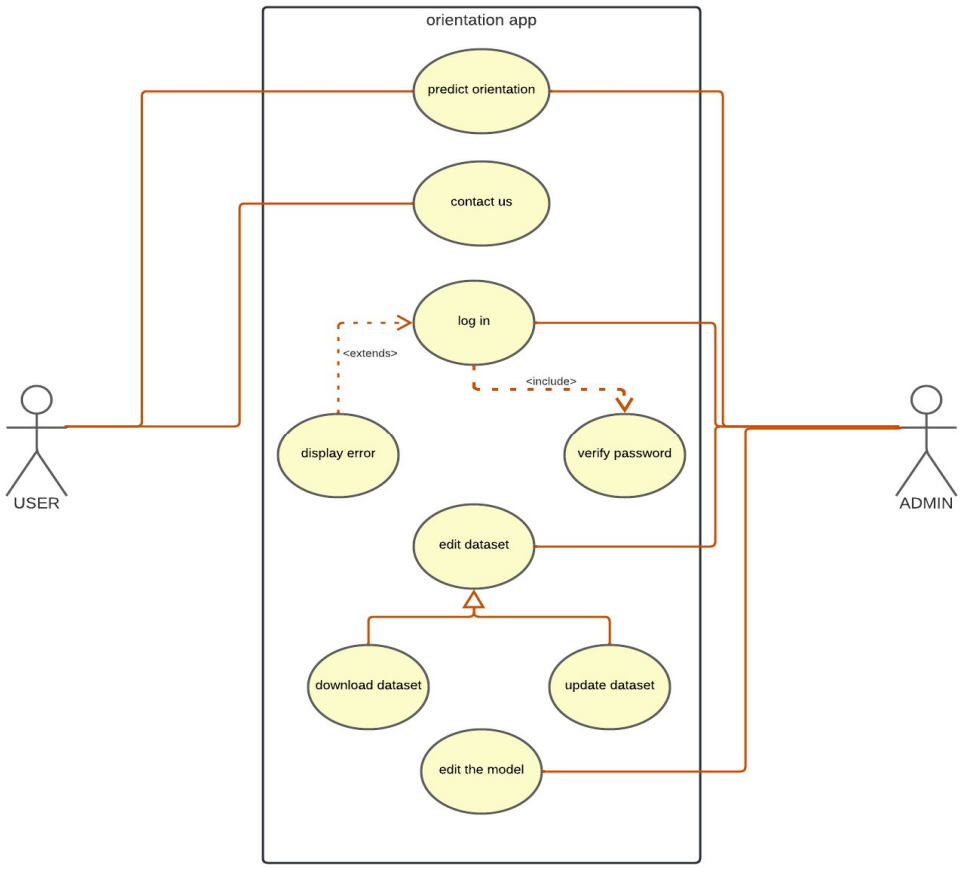

Figure 20: Use Case Diagram

3.1.1 USER

When the user enter the website he can go to predict orientation page by clicking predict orientation link in index page, then he can enter his grades and see his best orientation mathematics or computer science (Figure 20).

Also the second functionality the user can do is send a message to the admin of the website.

3.1.2 ADMIN

When the admin enter the website (Figure 20), he can:

1- Enter to login page for entering his information "username" and "password" the application will process his information if it is correct the user will enter to the admin state and he will be redirected to the index page, if it is wrong the app will print an error message
2- Enter predict information page like the user
3- Enter edit dataset page after that, he can download the dataset file and upload a new dataset file that will replace the old one
4- Enter edit the model page, after that he can create and train and test the model by entering the percentage of test and train of dataset

3.2 Activity diagram

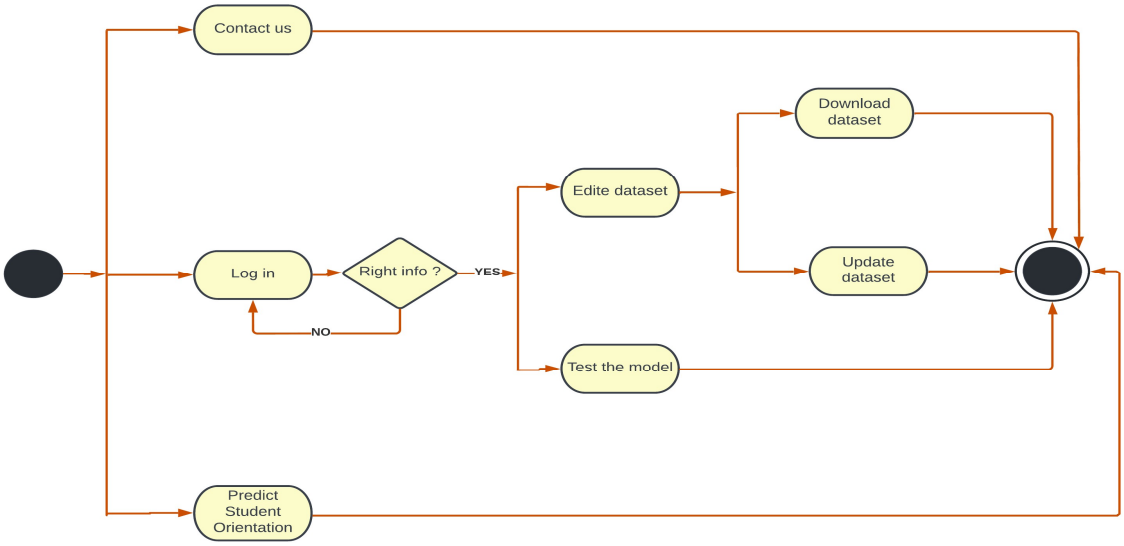

Figure 21: Activity Diagram

When a user enters the website, he can enter 3 pages (Figure 21).

The first one is predict orientation page, in this page he will enter his grades and see the best orientation for him.

The second one is Contact us page in this page you can send a message to the admin of website if you have any question.

The third one is login page when he can enter his information username and password and see if it is correct, He will enter the admin state. This state allows you to enter edit dataset page and edit the model page.

- Edit Dataset pageallows you to edit the dataset by updating a new dataset file, the file must be in csv formator download current file.
- Edit Model pageallows you to create and manipulate the model by entering the percentage of training and testing.

3.3 Sequence Diagram

Each page has its own Sequence diagram

3.3.1 Index Page Sequence Diagram

It is the first page (Figure 22) that the user can see when entering the website

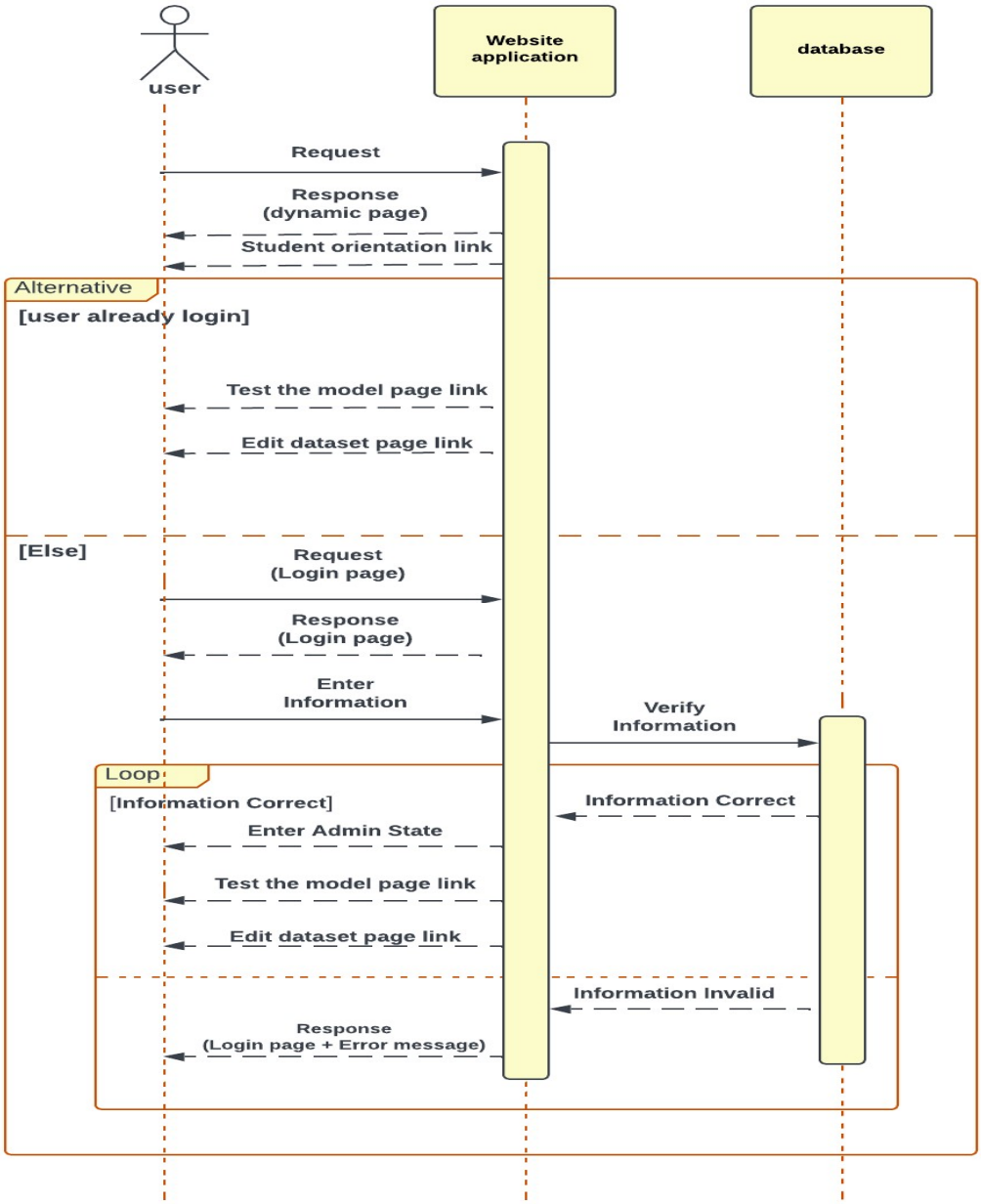

Figure 22: Sequence Diagram “index page”

When user receive the page he can go to student orientation page by clicking on "Computer science or Mathematics" link, Also he can login to enter into the administrator status

3.3.1.1 Login Operation

First the user has to click on login link to go to "login page"

When user receive the page, he will enter his information and click submit button.

The information will be sent to the website application, after that the website application will verify user information by comparing them with the information saved in the database.

If the information was correct the user will change its status the admin and will receive "edit the model" and "edit dataset" pages links and get access to them

3.3.2 Students Orientation Page Sequence Diagram

All users of the application can get access to this page (Figure 23).

The whole purpose of the page is to give best orientation for students based on their grades.

First when the server of our whole application runs, he create the model before everything

3.3.2.1 Creating the model

Student Orientation Application gets the dataset from its folder in form of CSV file, after that the application create and train the model from Dataset file.

When create model operation finished the user can get “Student orientation” page by sending request to the website application and the app will respond with the page.

After the user gets the page he can enter his grades (BAC + First year) and click submit to send the information.

The websites application receives the information, verifies it and sends it to Student Orientation Application, the app will predict the best orientation for the user based on their grades and resend it to the websites application.

The website application take information and send it to the user in html page.

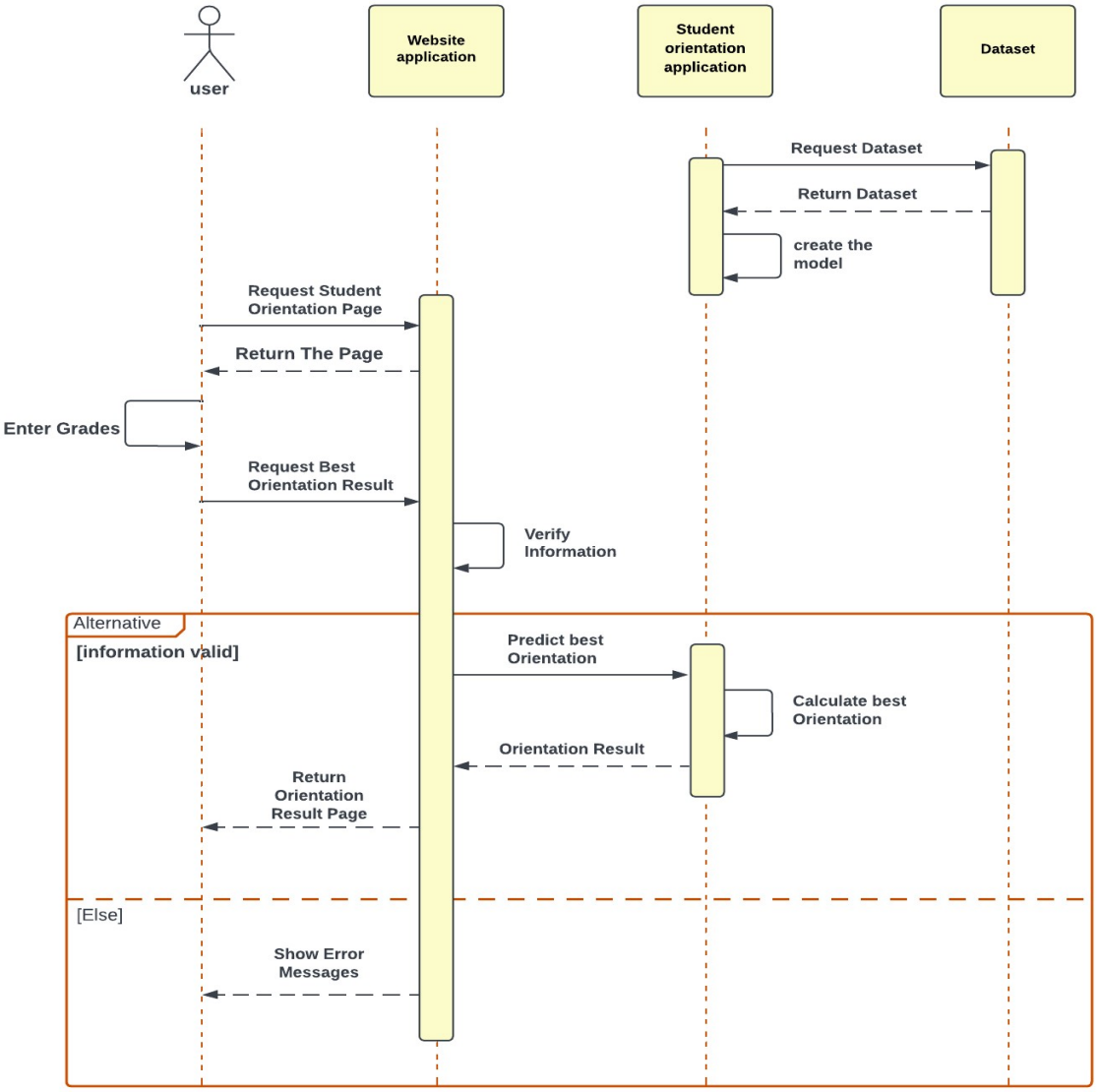

Figure 23: Sequence Diagram "Student Orientation page"

3.3.3 Edit the Model Page Sequence Diagram

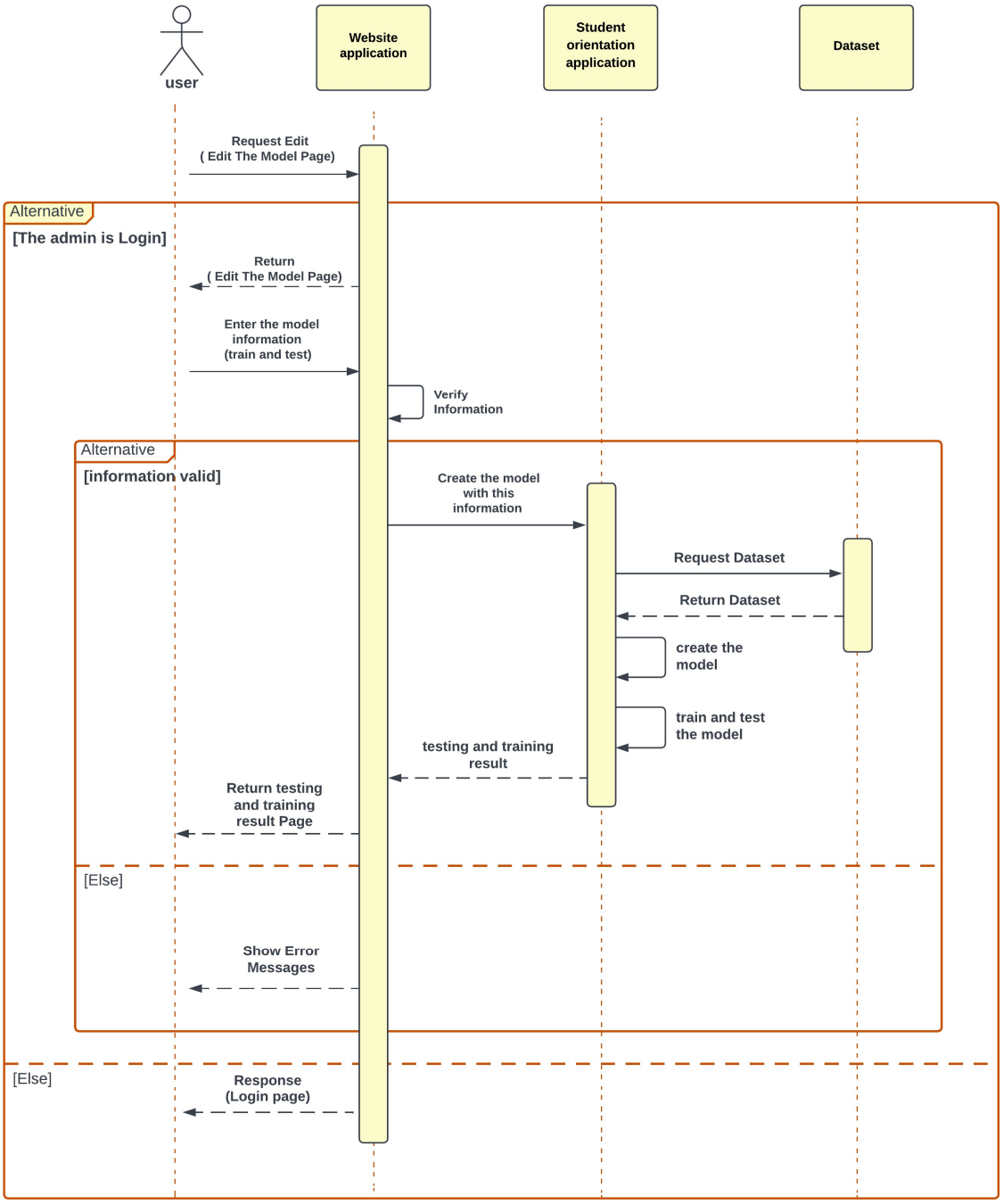

Figure 24: Sequence Diagram "model editingpage"

Edit the Model page (Figure 24), allows the admin to create and train the model.

The user send a request for Edit the Model page to the website application

When the request received to the website application, the app will checks if the user is in the administrator state, if it is the app will send the page to the user if not the user will be redirected to login page.

When the user receives the page, he enters the training and testing percentages for the model and send the information to website application

After the website application receive the information, it will be verified and send it to Student Orientation Application.

When Student Orientation application received User information, he gets Dataset file, after that it create the Model from the Dataset and train and test the Model from User information.

After the process is finished, Student Orientation application will send test result to the website application and the Website application will show the result to the user in form of html page.

3.3.4 Edit DatasetPage Sequence Diagram

Edit Dataset page (Figure 25) allows the admin to download and update the Dataset.

The user send a request for Edit the Model page to the website application.

When the request received to the website application, the app will checks if the user is in the administrator state, if it is, the app will send the page to the user if not user will be redirected to login page.

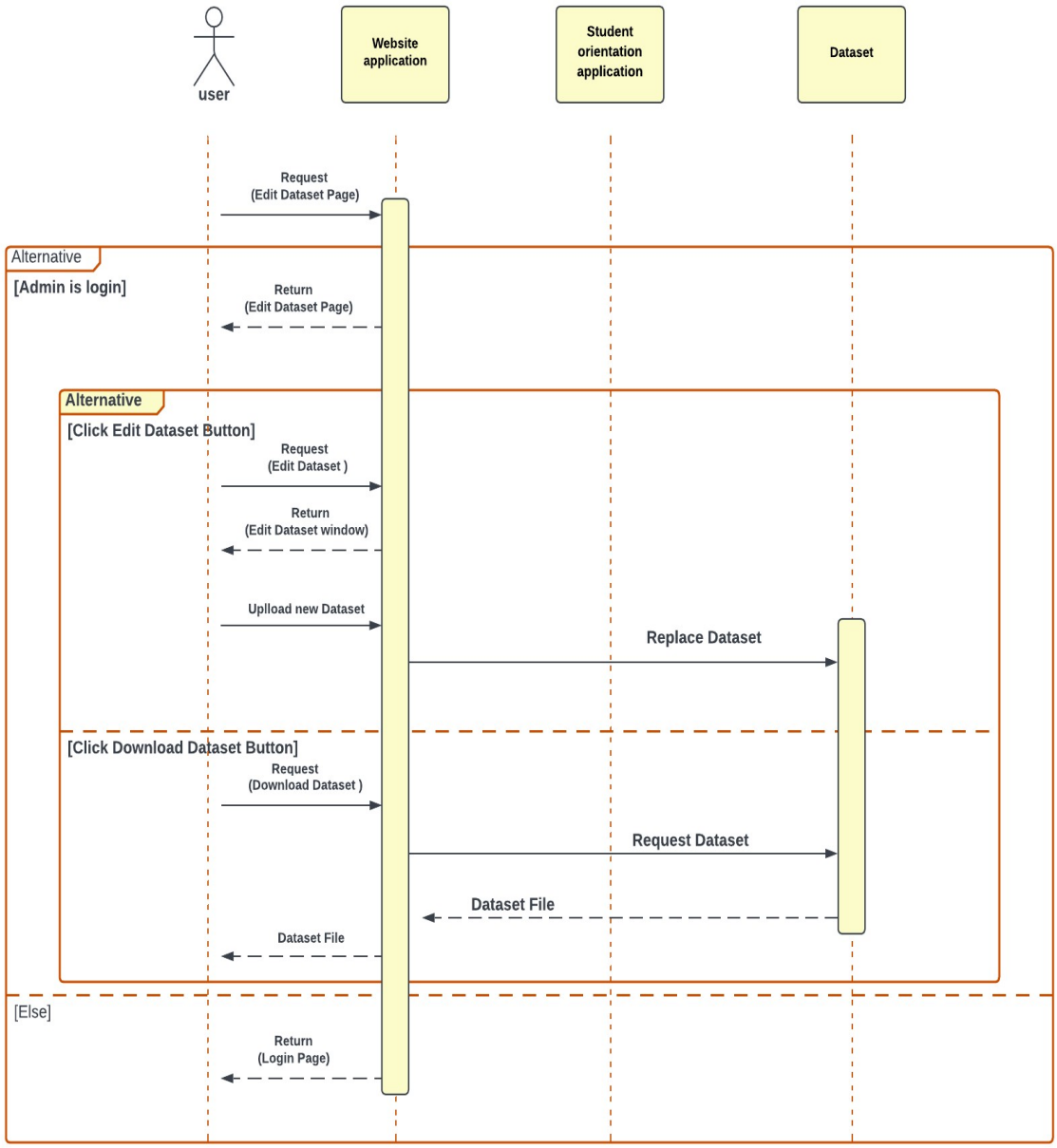

Figure 25: Sequence Diagram "dataset Editing Page"

When the user receives the page, he can download or update The Dataset file.

If User click the Download link a request message for the dataset file it will be sent to the website application, after that the website application will get the data set file and send it to the user.

If User click the Update link a request will be sent to the website application, after that the website application will send upload window to the user the user upload the new dataset file and sent it to the website application.

After the website application receives the new file, it will replace the old file.

4 Decision Tree

4.1 How we create our decision tree?

4.1.1 Predefined Function and Classes used

We use **sklearn.tree.DecisionTreeClassifier** is a class capable of performing multi-class classification on a dataset in python.

DecisionTreeClassifier takes as input two arrays: an array X, sparse or dense, of shape (n_samples, n_features) holding the training samples, and an array Y of integer values, shape (n_samples,), holding the class labels for the training samples.

We also use **sklearn.model_selection.train_test_split** to split arrays into random train and test subsets.

4.1.2 Actions

1- import dataset file

2- split dataset 80 for training and 20 for testing by using the **train_test_split** method

3- Use **DecisionTreeClassifier** class to create and train the model

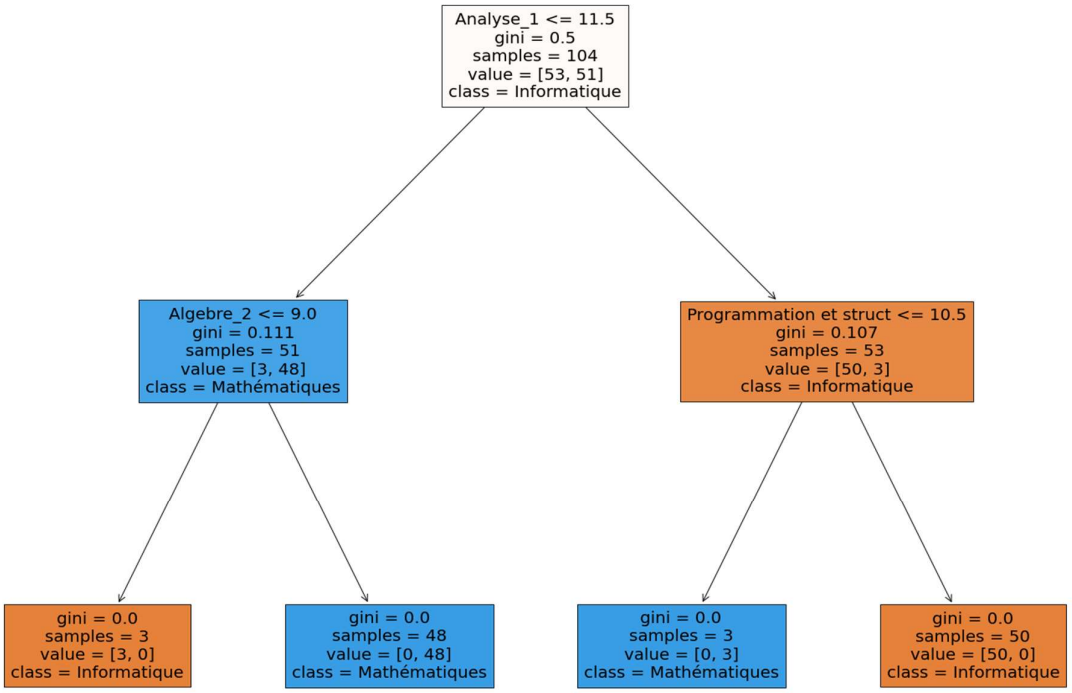

Figure 26: Decision Tree

4.2 Explain Decision Tree

Decision tree predicts the best direction by use multiple condition on user grades.

The first is tested if the **Analyse_1** score is less than or equal **11.5** or not

If yes it will test if **Algeber_2**scores is less than or equal**10.5** or not

If yes the best orientation is **Computer Science**.

If not the best orientation is **Mathematics.**

If not it will test if **Programmation_et_struct** scores is less than or equal**10.5** or not

If yes the best orientation is **Mathematics.**

If not the best orientation is **Computer Science.**

```
if (Analyse_1<=11.5) {
        if(Programmation_et_struct<=10.5){
Return(“ Mathematics “)
}
        else(){
        Return(“ Computer Science ”)
}
}
else{
        if(Algeber_2<=9.00){
        Return(“ Computer Science ”)
        }
        else{
        Return(“ Mathematics “)
        }
}
```

5 Conclusion

In this chapter we describe the architecture and behavioral of the application. We use three behavioral diagrams (use case diagram, sequence diagram and activity diagram) to explain the different function of the application and details of the interaction between the user and different parts of the application and the interaction between the parts of the application with each other.

Also we explain the method used to create the decision tree and we explain how it work to orientate the students.

Chapter IV

1 Introduction

Implementation is one the most necessary part for system conception on this chapter we will introduce our site content that we can found on the website parts(main bar, the footer, Index page, Edit dataset page, Edit the model page, Contact us, and Login page) and how to use it and the language, framework we used to build it with a simple definition for every one of them.

2 Code Editors used

2.1 Visual Studio

Visual Studio is an **Integrated Development Environment(IDE)** developed by Microsoft to develop GUI(Graphical User Interface), console, Web applications, web apps, mobile apps, cloud, and web services, etc. With the help of this IDE, you can create managed code as well as native code. It uses the various platforms of Microsoft software development software like Windows store, Microsoft Silverlight, and Windows API, etc. It is not a language-specific IDE as you can use this to write code in C#, C++, VB(Visual Basic), Python, JavaScript, and many more languages. It provides support for 36 different programming languages. It is available for Windows as well as for macOS.

2.2 PyCharm

PyCharm is the most popular IDE used for Python scripting language. This chapter will give you an introduction to PyCharm and explains its features.

PyCharm offers some of the best features to its users and developers in the following aspects

- Code completion and inspection
- Advanced debugging
- Support for web programming and frameworks such as Django and Flask

3 Languages and Frameworks used

3.1 Python

“Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.” ** www.python.org **

3.2 Django

“Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source

3.2.1 How does Django Work?

Django follows the MVT design pattern (Model View Template)

Model: The model provides data from the database,In Django, the data is delivered as an Object Relational Mapping (ORM), which is a technique designed to make it easier to work with databases.

The most common way to extract data from a database is SQL. One problem with SQL is that you have to have a pretty good understanding of the database structure to be able to work with it.

Django, with ORM, makes it easier to communicate with the database, without having to write complex SQL statements.

The models are usually located in a file called models.py.

View: A view is a function or method that takes http requests as arguments, imports the relevant model(s), and finds out what data to send to the template, and returns the final result.

The views are usually located in a file called views.py.

Template: A template is a file where you describe how the result should be represented. Templates are often .html files, with HTML code describing the layout of a web page, but it can also be in other file formats to present other results, but we will concentrate on html files.

3.3 JavaScript

JavaScript is a cross-platform, object-oriented scripting language used to make webpages interactive (e.g., having complex animations, clickable buttons, popup menus, etc.). There are also more advanced server side versions of JavaScript such as Node.js, which allow you to add more functionality to a website than downloading files (such as realtime collaboration between multiple computers). Inside a host environment (for example, a web browser), JavaScript can be connected to the objects of its environment to provide programmatic control over them.

JavaScript contains a standard library of objects, such as Array, Date, and Math, and a core set of language elements such as operators, control structures, and statements. Core JavaScript can be extended for a variety of purposes by supplementing it with additional objects; for example:

- *Client-side JavaScript* extends the core language by supplying objects to control a browser and its *Document Object Model* (DOM). For example, client-side extensions allow an application to place elements on an HTML form and respond to user events such as mouse clicks, form input, and page navigation.
- *Server-side JavaScript* extends the core language by supplying objects relevant to running JavaScript on a server. For example, server-side extensions allow an application to communicate with a database, provide continuity of information from one invocation to another of the application, or perform file manipulations on a server.

This means that in the browser, JavaScript can change the way the webpage (DOM) looks. And likewise, Node.js JavaScript on the server can respond to custom requests from code written in the browser.(developer.mozilla, 2022)

3.4 HTML

HTML stands for HyperText Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document within tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g. HTML) are human-readable. The language uses tags to define what manipulation has to be done on the text.
HTML is a markup language used by the browser to manipulate text, images, and other content, in order to display it in the required format. HTML was created by Tim Berners-Lee in 1991. The first-ever version of HTML was HTML 1.0, but the first standard version was HTML 2.0, published in 1995.

(geeksforgeeks, 2022)

3.5 CSS

“Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

CSS is among the core languages of the open web and is standardized across Web browsers according to W3C specifications. Previously, development of various parts of CSS specification was done synchronously, which allowed versioning of the latest

recommendations. You might have heard about CSS1, CSS2.1, CSS3. However, CSS4 has never become an official version(developer.mozilla, 2022)

4 Website Pages

4.1 Header and footer

4.1.1 Main Navigation Bar

Figure 27: Navigation Bar

In the navigation bar (Figure 27), we have four links

1.The first is the logo link that takes you to the index page

2.The second is the "About Us" link that takes you to the footer part where you can see information about the website.

3.The third is Contact Us link that takes you to a contact uspage.

4.The fourth is login link that takes you to a login page

4.1.2 Footer of the page

Figure 28: Footer of the page

The Footer (Figure 28) show you general information about the website and a link of Abbes Laghrour University website.

4.2 Index Page

This is the first page you see when you enter the website (Figure 29).

If enter to the website as user you will just get a link for student orientation page.

4.2.1 User will see

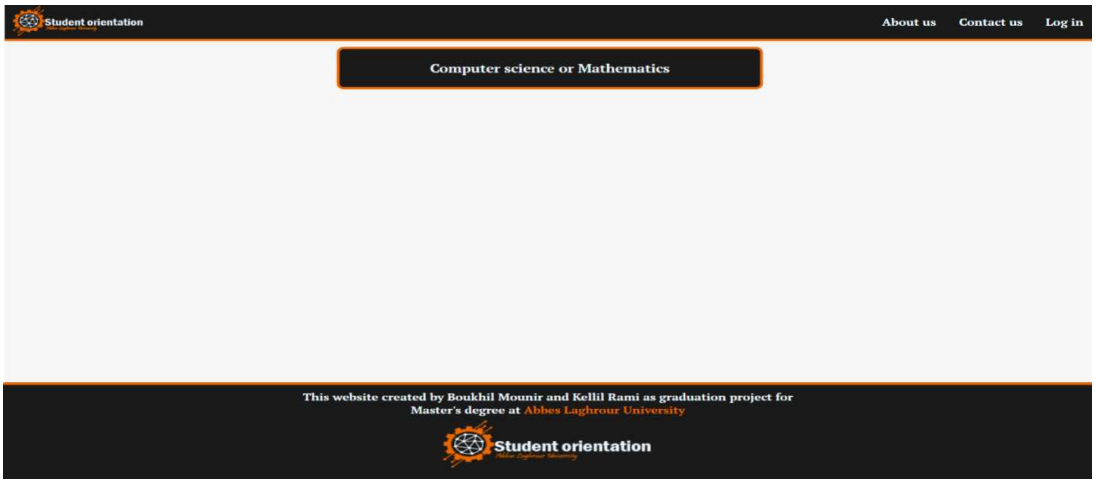

Figure 29 : Index page for User

4.2.2 Admin will see

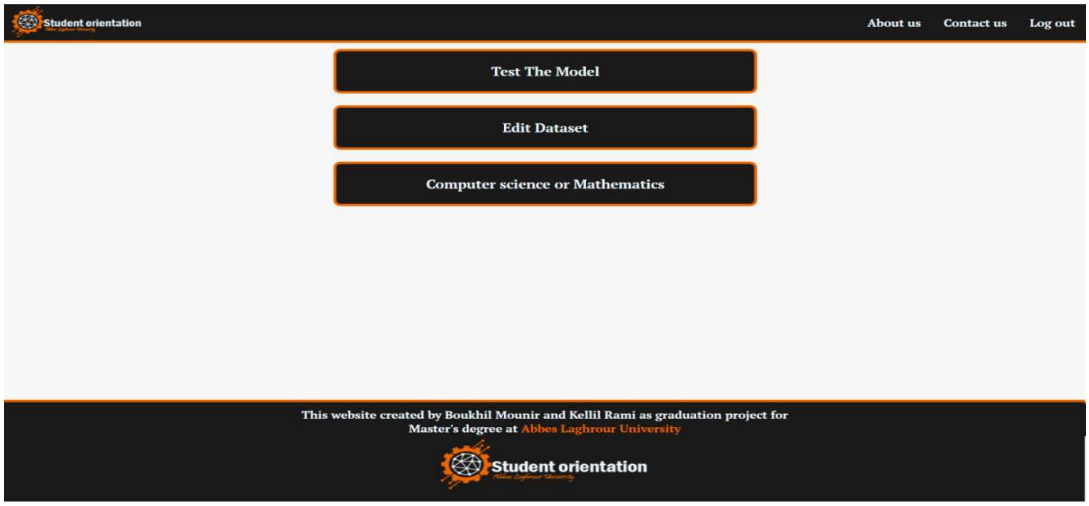

Figure 30: Index page for admin

If you enter to the website as an administrator, you will see links to the Student Orientation page, the Edit Dataset page, and the Edit Model page (Figure 30).

4.3 Login Page

You can access this page (Figure 31) by clicking on the login link in the navigation bar.Here you can enter the user name and password to login into the administrator account to access

Edit Dataset page and Edit Model page.

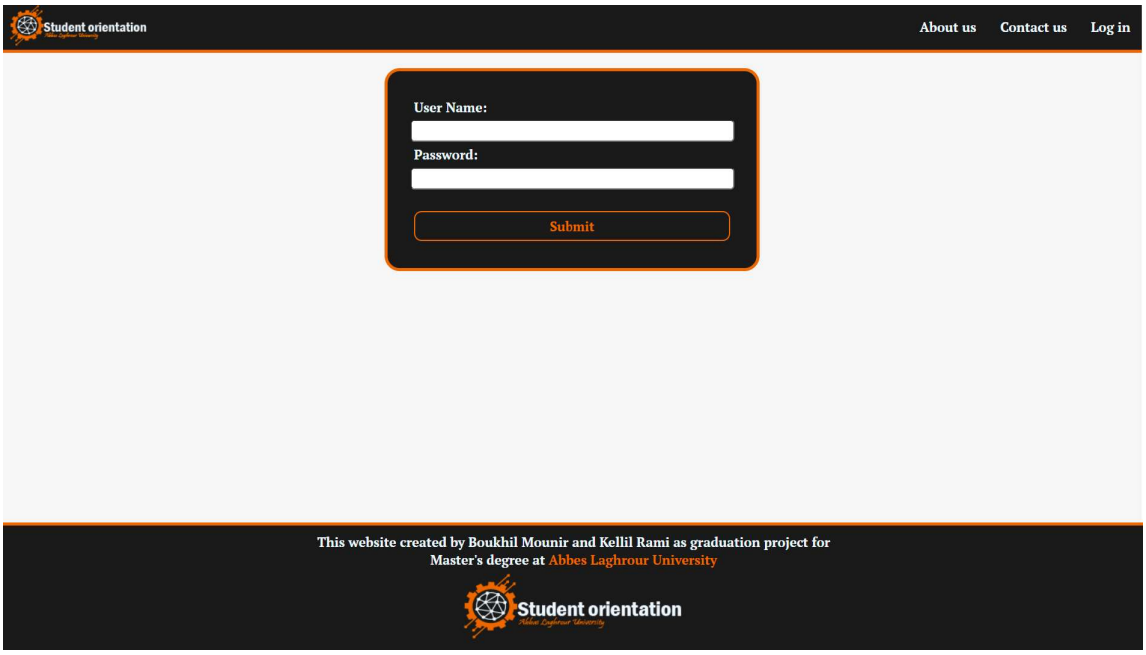

Figure 31: Login Page

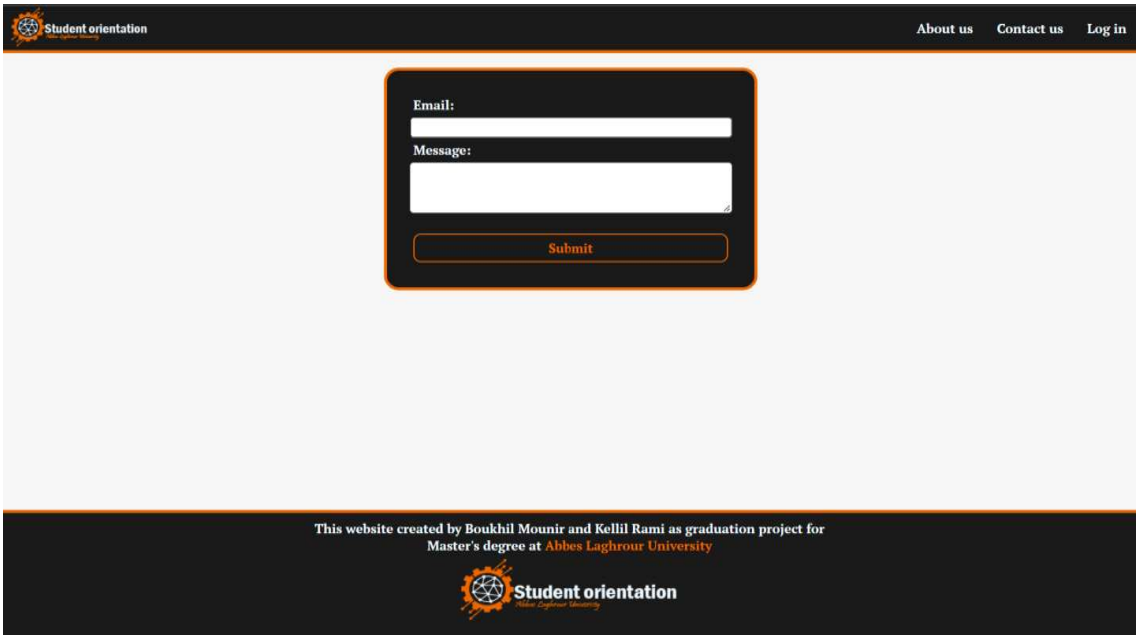

Figure 32: Contact us Page

4.4 Contact us Page

You can access this page (Figure 32) by clicking on the Contact us link in the navigation bar. Here you can enter your email and the message you want to send and after that click Submit button to send the message.

4.5 Student Orientation Page

You can access this page (Figure 33) by clicking on the Computer Science or Mathematics link in index page.

The page divided into three parts.

The first part is when you can enter your BAC grades.

The Second part is when you can enter your first year grades

The third part is to submit the information by clicking the submit button

After clicking the submit button, a result window will appear (Figure 34)in which you can see your best orientation.

4.5.1 Student Orientation page (before entering information)

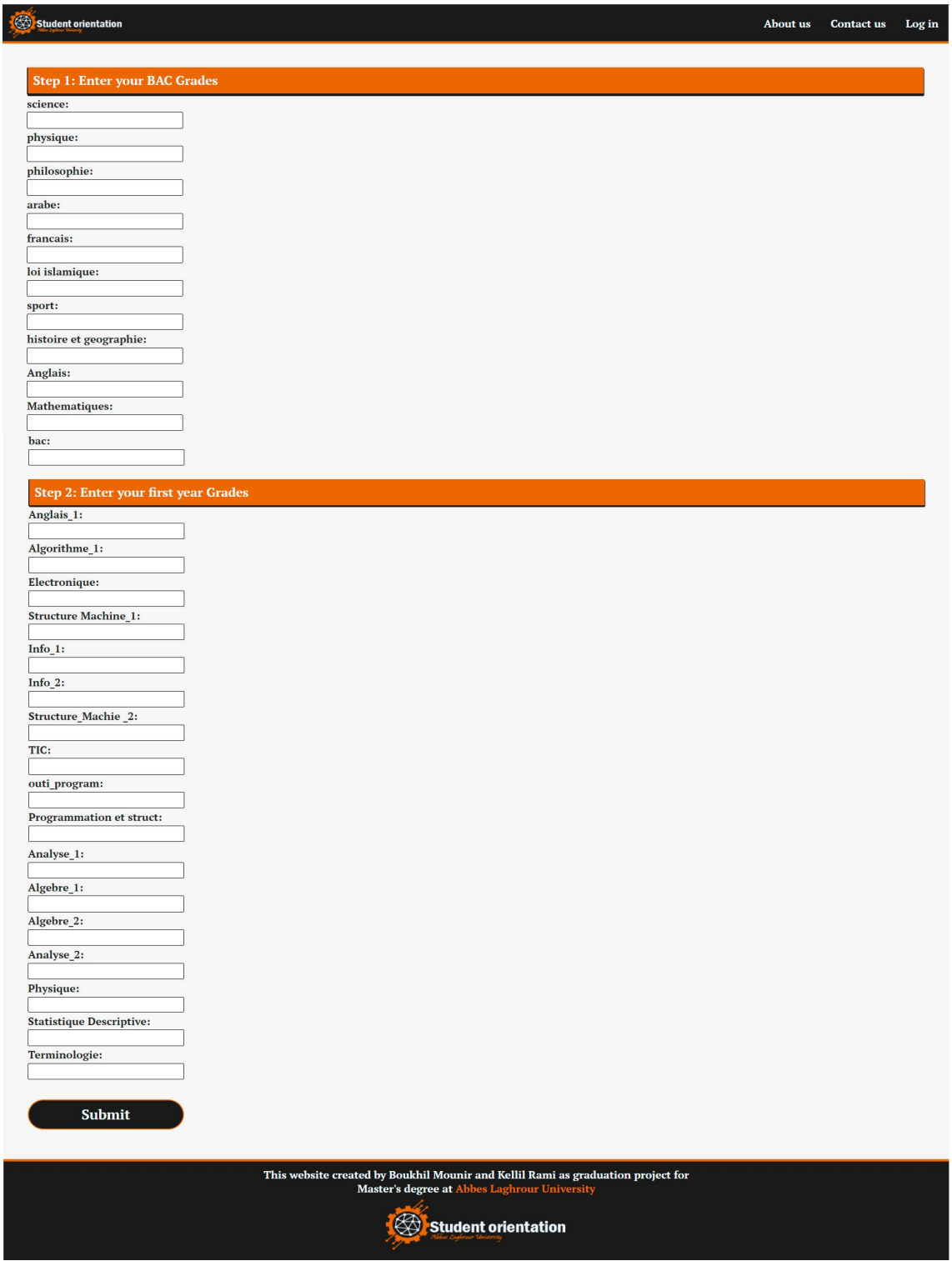

Figure 33: Student Orientation page (before entering information)

4.5.2 Student Orientation page (after entering information)

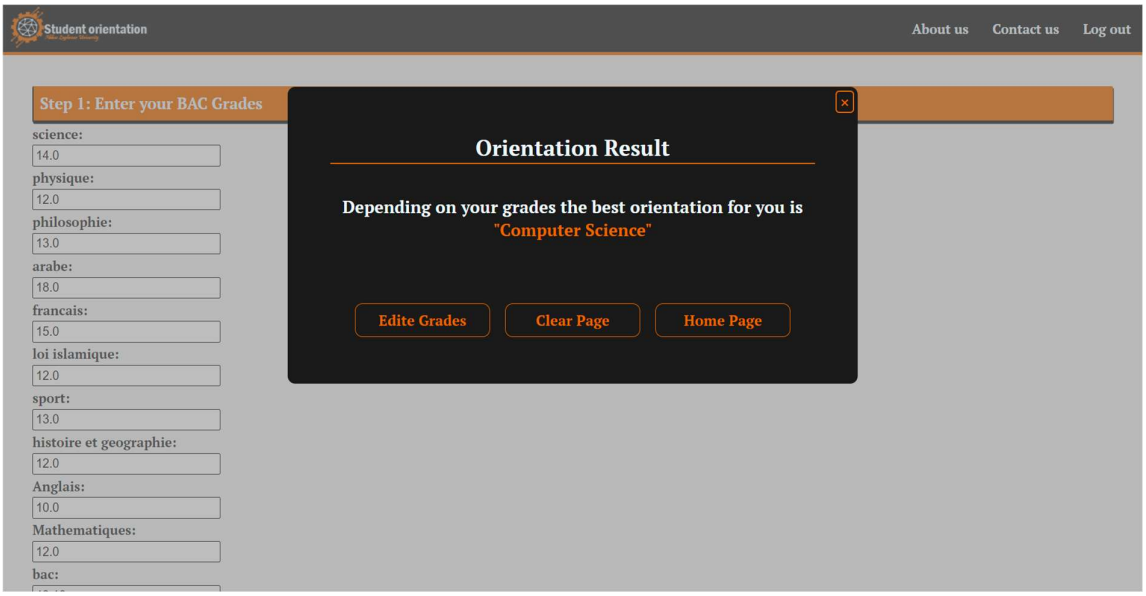

Figure 34: Student Orientation page (after entering information and clicking submit button)

4.6 Test Model Page

You can get this page (Figure 35) by clicking on the Test Model link on the Index page but you must be in Administrator status first.

On the left, you can see the range of the train model where you can enter the percentages of the dataset that will be used for training

On the right, you can see the range of the test model where you can enter the percentages of the dataset that will be usedfor testing.

On the bottom, you can see test the Model button where you will submit your information.

After submitting your information, a window(Figure 36) will open showing you the test result.

4.6.1 Test Model Page (before submitting your information)

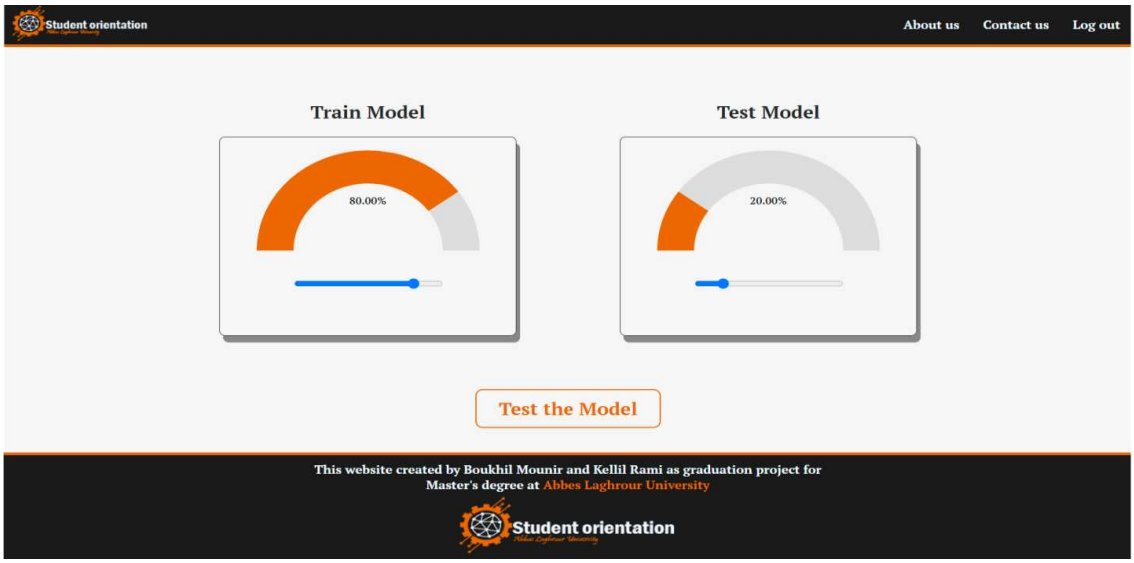

Figure 35: Test Model Page (before submitting the information)

4.6.2 Test Model Page (after submitting your information)

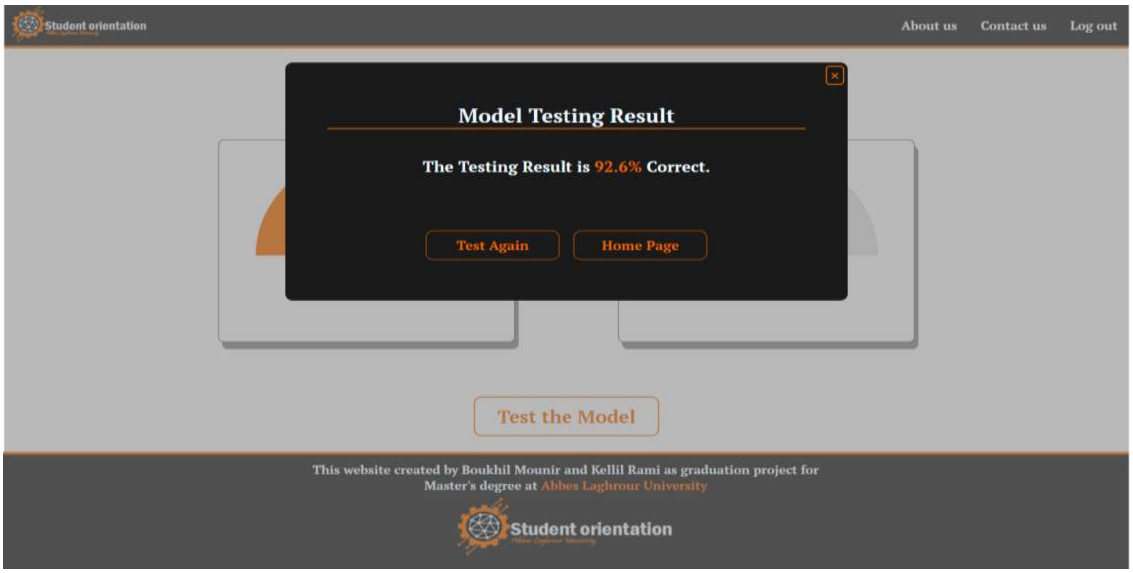

Figure 36: Test Model Page (after submitting your information)

4.7 Edit Dataset page

You can get this page (Figure 37) by clicking on Edit Model link on the Index page but you must be in Administrator status first.

On this page you will find two buttons, the first is Update Dataset at the top and the second is Download Dataset at the bottom.

If you click on update Dataset button, a window will appear to select the new dataset file.

If you click the Download Dataset button, the Dataset file will be downloaded into your device (Figure 38)

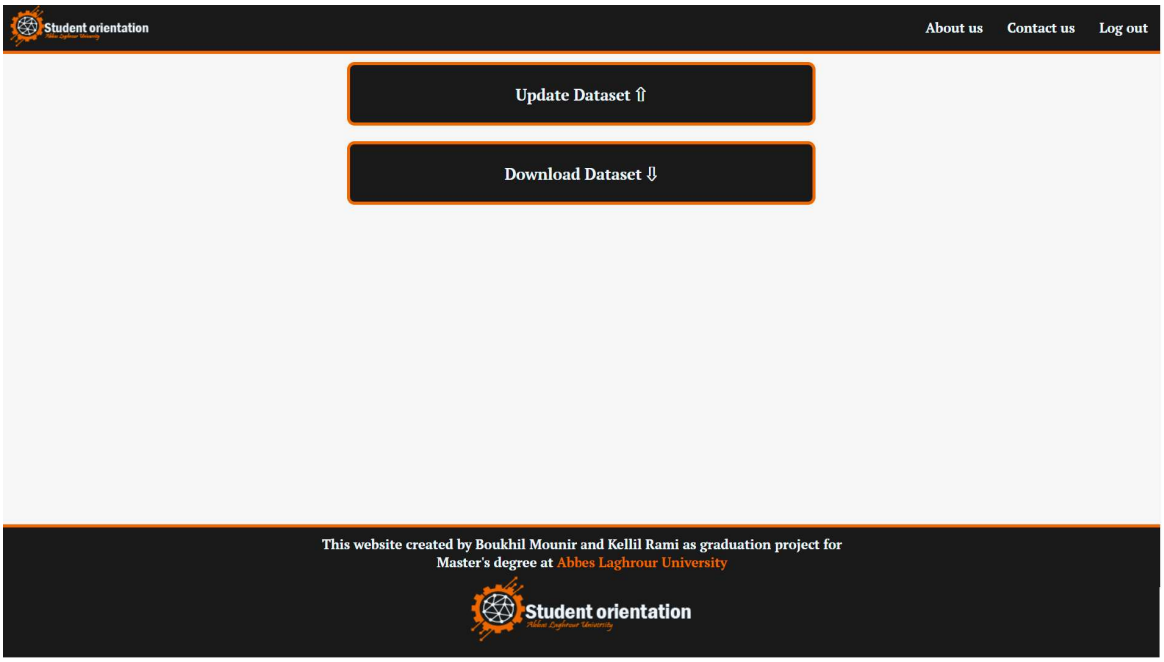

Figure 37: Edit dataset page

4.7.1 Edit dataset Page (after clicking on Download Dataset button)

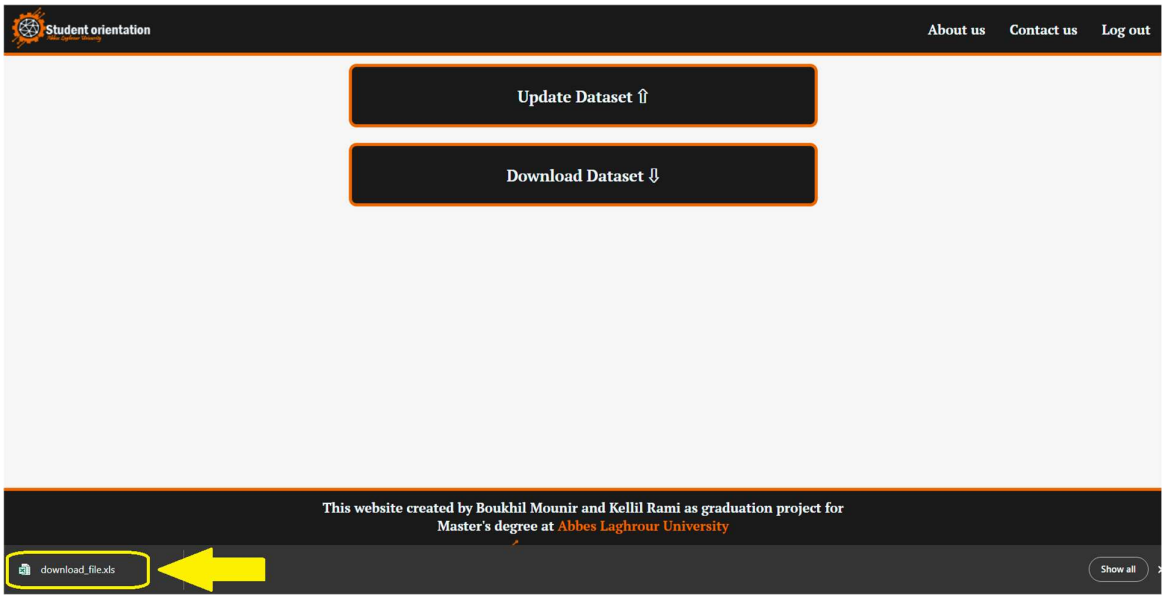

Figure 38: Edit dataset Page (after clicking on Download Dataset button)

4.7.2 Edit dataset Page (after clicking on Update Dataset button)

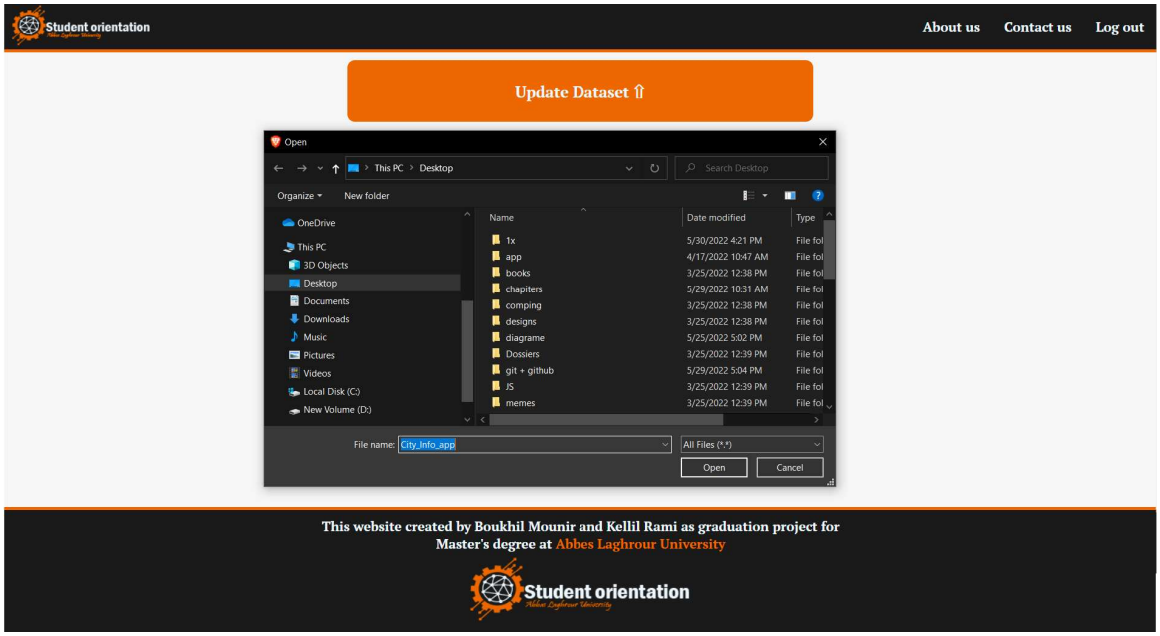

Figure 39: Edit dataset Page (Download file)

5 Conclusion

In this part we described the technical architecture which will receive our solution as well as the environment of developments and we also presented all the tools that we used and which are Visual Studio and PyCharm code editor used.

We also Explain the pages (index page, login page, contact us page, Edit dataset page, Edit the model page, header and footer) the user will see when he interact with the application and give some instruction to get better experience in the website

General Conclusion

In conclusion, on the previous chapters we have introduce the machine learning her category like supervised learning unsupervised learning and Reinforcement Learning and their advantages that make it easy to see why people think that machine learning is a solution for a lot of hard world problems than we understand that we can solve the problem of the bad decisions from the students in choosing specialty (which could cause that they could not pass the year) by the predictive analytics using a supervised learning machine learning algorithm called the decision tree that could help for a good decision using previous data of the past students and pycharm help us to use python that could help us to use that algorithm(as back end for the project)and cause the users need a platform or a front end to use that algorithm and get the help that given by it we create a web site using html, css, javascript in visual code(editor code) with some diagram like Use Case Diagram, Activity diagram and Sequence Diagram to make a plan and organize the web site content and django to make a link between the python program and the web site.

Bibliography

[(L. Breiman, 1984)] Classification and Regression tree 1984 by L. Breiman, J. H. Friedman, R. Olshen, and C. J. Stone

[(Pang-Ning Tan, 2021)] to Data Mining(second edition) 2021 by Pang-Ning Tan, Michael Steinbach,Anuj Karpatne,Vipin Kumar (Pang-Ning Tan, 2021)

[(Pisani, 2020)] MACHINE LEARNING by Pisani, Mikaela (Pisani, 2020)

[(Raschka, 2020)]Introduction to Machine Learning and Deep Learning 2020 by Raschka, Sebastian

[(data-flair.training/blogs)]data-flair.training Dataflair is the rapidly growing online learning platform that provides quality education and offers live, instructor led courses.

[(developer.mozilla, 2022) developer.mozilla.org(developer.mozilla, 2022)]TheMDN Web Docs site provides information about Open Web technologies including HTML, CSS, and APIs for both Web sites and progressive web apps.

[geeksforgeeks.org (geeksforgeeks)]A Computer Science portal for geeks. It contains well written, well thought and well explained computer science and programming articles

[tutorialspoint.com (tutorialspoint)] Tutorials Point is a leading Ed Tech company striving to provide the best learning material on technical and non-technical subjects

[w3schools.com (w3schools, 2005)] W3Schools is a freemium educational website for learning coding online Developed in 1998 Owner is Refsnes Data.