Curriculum Vitae (CVs) Evaluation Using Machine Learning Approach

Rabih Haddad, Eunika Mercier-Laurent

Curriculum Vitae (CVs) Evaluation Using Machine Learning Approach

Rabih Haddad 0000-0001-6508-1057 and Eunika Mercier-Laurent

University of Reims Champagne-Ardenne,CRESTIC Laboratory EPITA Engineering School of Computer Science

http://www.univ-reims.eu/
http://www.epita.fr/en

Abstract. Resumes or Curriculum Vitae (CVs) are still an important standard document and a decision element in evaluating the life journeys and human personalities of candidates. Its main role is to detect the eligibility of people who are applying to job vacancies or higher education programs. This research work ambitions in elaborating a system that automates the preselection of eligibility and assessment of candidates in the higher education students' recruitment process. This system will replace the tedious tasks of manual processing of CVs and will provide accurate and effective evaluation results. To achieve this requirement, the system will be implemented using a machine learning approach using different classification algorithms. The evaluation is conducted on the four main knowledge categories that build the CV: personal information, academic background, professional experience, and soft and technical skills. The output of the system will be an indicator to shortlist, discard or request more information to evaluate the candidates' eligibility. Moreover, the scores obtained for each part of the CV will be used to calibrate the indicator in each information category. Consequently, this system boosts the recruitment process of candidates and provide a reasonable decision

Keywords: Curriculum Vitae, Data and Text Mining, Natural Language Processing, Classification, Machine Learning, Naïve Bayes Classifier, Support Vector Machine, and Random Forest

1 Introduction

CVs are still an important standard document and a decision element in evaluating life journeys and human personalities of candidates who are applying to a specific job or pursue an academic program. This work is motivated by real need of automated processing of students applications.

International students who wish to continue their higher studies at the bachelor, or masters, or doctorate level in France should directly obtain their admission from the corresponding institution or by choosing a French institution on Campus France [4]. To apply, candidates should fill out an application form

on the institution website. The characteristics of each country and institution shape the admission system, however, there is a big percentage of commonality between these systems as they require the same traditional documentation, evaluation, information: admission and languages proficiency exams, interviews, CVs, transcripts, motivation letters, and recommendations letters. [This is the procedure that is applicable today at EPITA [2].]

This is a important challenge for automated processing. In this article, a system that automates the eligibility screening and assessment of applicants in the process of recruiting higher education students will be discussed. This system might replace the tedious tasks of manually processing Curriculum Vitae and provide accurate and efficient assessment results. Considering the various forms of the CV and the needs of extracting information various classification algorithms might be used to handle this exercise. The goal of CV Analysis is to give an outcome at the current stage whether the CV fulfills the criteria to be eligible for a certain academic program. For the current experimentation, the analysis scope is applied to the candidates who are applying to pursue a Masters degree at EPITA [3].

The assessment will focus on the four main categories of knowledge that make up the CV: personal information, academic background, work experience, and personal and technical skills. Spoken languages and extracurricular activities are considered as well. The result of the system would be indicative for a shortlist, ignore or request more information to assess the eligibility of some candidates.

Based on what has been elaborated earlier and the availability of an important volume of CVs to be tested in the frame of this research, a Machine Learning approach has been selected. After research and study of the nature of information that are available and the existing algorithm, the below evaluation process will be studied. The evaluation process will be divided into several steps as follows and as in figure 1:

1. System Parameterization
2. Training Data Collection
3. Model Dataset Creation
4. Data Extraction and Cleansing
5. Information Extraction using Natural Language Processing
6. Classification and final Output Prediction

Fig. 1. Evaluation Process

2 Literature Review

The evaluation process that is presented in this paper, is mainly based on data mining and text mining. Data mining is the process of finding discrepancies, patterns, and correlations in large data sets to predict outcomes. With a wide array of technologies, this information can be used to apply in many life sectors. Methods of data mining can be decision trees, associations rules, segmentation algorithms, nearest neighbors' algorithm, and neural networks [8]

In data mining data available is used in databases to identify unknown, important and useful combinations and structures and patterns. The concepts and methods of data mining can be applied in various fields like marketing, medicine, engineering, web mining, etc. Educational data is a new emerging data mining technique that can be applied to many fields including education. This new emerging field, called educational data mining, concerns the development of methods for discovering knowledge from data from educational settings.

One of the researches, which has been already done on CV evaluation using text mining and the related techniques, is mentioned here [9]. Many works exist that mention information extraction application on CV, but the evaluation of such CVs is completely absent. For example, in one of the papers that has been recently published in ResearchGate, the study included on how to extract information from conceptual pattern from CV. The aim was to perform only the below tasks out of it:

- Identifying information to be extracted
- Preparation of linguistic and ontological resources for applying IE tools in Polish language
- Preparing tools to extract relevant information from texts
- Applying selected tools on the test document collection
- Validation of the results.

This might facilitate the task of preparing the CV text prior to being processed and evaluated, however, it will not address the main task of conducting the evaluation. Another work has been done in this regard and presented in the Journal of Critical Reviews [5]. It establishes clear and efficient registration procedures by integrating strategies of web dragging, content extraction and regular language management. This makes the registration process basic and successful by offering the possibility of transferring the course material for the required skill during the work. Additionally, it provides a scheduled suggestion for the activity seeker when individual abilities are coordinated with the activity station. This is done using the cooperative separation account, in addition to improving the additional space for specialist collaboration. On the business side, it uses the web crawler to generate a set of business responsibilities and basic requirements. On the activity seeker side, when the CV is published, word separation and letter splitting are stopped. After content segmentation, enrollment is based on training, an understanding of the job, abilities, personality traits and grade frequency. Finally, a framework is proposed that the next age at which the level of education meets the prerequisites of the profession. This approach is a bit

general and its output is abstract, making it difficult to give a valid or coherent evaluation indicator.

Hence, in this work, further experimentation aligned with the methodologies is presented in order to get more adjourned results.

3 Proposed System

As mentioned earlier, a Machine Learning approach is used in this paper to conduct the evaluations of CVs. This approach is be based on the steps that were mentioned in the first section of this article. A detailed elaboration of each step is presented below and the CV evaluation process architecture is presented in figure 2.

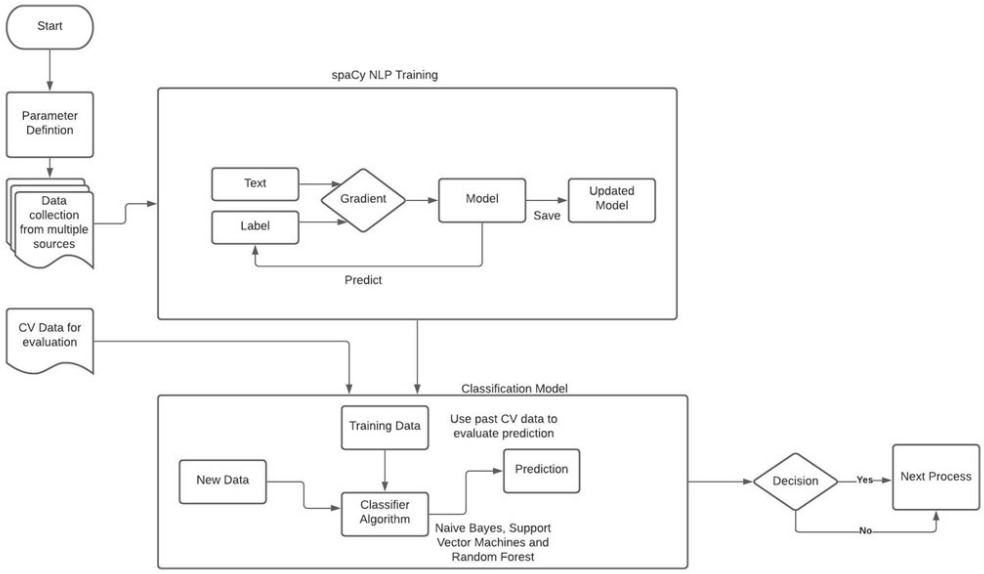

Fig. 2. CV Evaluation Process Architecture

3.1 System Parameterization

The first step that was conducted is to train the evaluation algorithm using historical CVs and using some existing datasets. It is an essential exercise to guarantee the accuracy of results. This can be done by determining the parameters to evaluate the CV quantitatively. The parameters are considered based on their usability and their helpfulness in providing the quality of the CV. For

instance, technologies practiced by the applicant is an important attribute to evaluate in the CV and has a weighted value of 5 points for this type. This step helps to identify the parameters against which the CV is evaluated. The weightings for each of its parameters were given based on its impact on the desirable qualities of the candidate. It also helps to extract only the relevant information from the CV. A decision table as shown in the figure 3 has been established to help in providing the overall score of the CV which can then be processed in the classification algorithms to determine the outcome of the candidate.

Technologies (Java , Python, Php etc..)	>5	>=3 and <=5	>=1 and <=3
Points	3	2	1

Fig. 3. Decision Table

3.2 Sample Data Collection for training

The process of data collection for training the algorithm is very essential as the trained algorithm can effectively determine the accuracy of the results. Therefore, a relevant data source has been used. Below is the list of data source used for various criteria:

- Technologies known: data source of Stack Overflow [11] questions and answers has been used. There are over a million records of questions and answers covering a wide range of technologies in various text combinations.
- Program specific criteria: various data sources are available in Kaggle [6] for each specific program type. Also, a manual insertion of data from historical CVs that were processed earlier at EPITA International Programs.
- Extracurricular activity: There is a Kaggle dataset available with over 666 hobbies and manually updating hobbies have not been covered in the above list.
- Languages: Only French and English are considered in our case making the related dataset simple.
- Type of degrees: The exhaustive list of degree types was obtained from Kaggle and Data World. The abbreviations for the degree types have been manually inserted.[7]

To train the Natural Language Processing NLP models, relevant data is needed to identify and quantify the parameters. By having comprehensive Datasets like stack-overflow questions and answers, the gaps in the matching patterns are easily identified. This data collection process is repeated for each of the parameters defined previously. The more comprehensive the dataset is the more efficient is the named entity recognition model obtained through NLP training. This step was done by keeping in mind the objective of obtaining accurate models.

These steps are also summarized in figure 4.

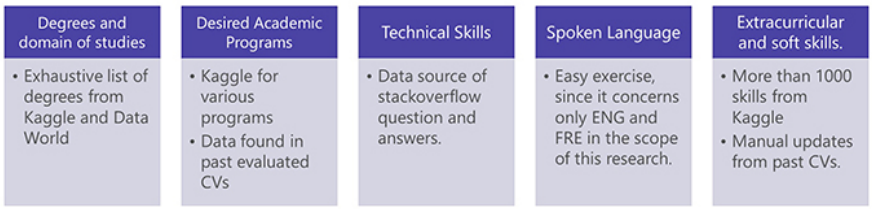

Fig. 4. Data Collection Samples

3.3 Dataset Creation for Testing the Trained Model

This step is to create a CV Dataset. Around 1000 CVs were randomly selected from the database of CVs available at EPITA from candidates applications. The CVs selected were based on the following criteria:

- The candidate whose CV has been selected should have applied for any of the international programs that are available.
- The admission process of the file should have been completed; therefore, the final status of the application is already known.
- Have a good combination of Accepted and Rejected CV's along with the combination of CVs of different programs that the candidates applied for.

In the previous step, specific parameter dataset has been used to be able to train each model independently. Then the collected CVs are trained in the desired context. The model is trained to learn words in the context of a CV, for example, Tesla is a company and not a physician. For this reason, the training data should always be representative of the data to process. A model trained in romantic novels will likely perform badly on legal text. This also means that in order to know how the model is performing, and whether it's learning the right things, not only training data is needed but we also evaluate data. If the model is tested only on data used in the training, chances are that it might not be generalizing. To train a model from scratch, are at least needed a few hundred examples for both training and evaluation. This step contributes to the evaluation data of the training data provided in step 2.

3.4 CV extraction and cleaning data

The main objective of this step is to parse information from a CV using Natural Language Processing and find the keywords, cluster them onto parameters based on their keywords. This information extraction process involves extracting structured specific information from unstructured or semi-structured natural language (in the form of text). The data extracted is then fed to the NLP tools like spaCy [15] for example for text categorization. PDF miner [12] has been used to parse the text from PDF files and the parsed result is cleaned by various methods using RegEx patterns to remove unwanted text and special characters and split the text in different lines.

Most text data contains a lot of words that are not actually useful. These words, called stopwords, are useful in human speech, but they do not have much to contribute to data analysis. Hence stop words and lemmatizing should be performed on the selected text. For example, words like connect, connection, connecting, connected, etc. aren't exactly the same, they all have the same essential meaning: connect. The differences in spelling have grammatical functions in spoken language, but for machine processing, those differences can be confusing, so we need a way to change all the words that are forms of the word connect into the word connect itself. Spacy has a built-in way to break a word down to its lemma by calling the method *lemma* to produce the lemma for each word after analyzing.

Because in CVs data formats that are used are not completely unstructured, it is still quite challenging to take them into the structured format as there is no set in stone rule for writing a CV. As a result, many possible ways of representing qualifications in a CV has been established so far such as chronological CV and functional CV. Beyond these two types, there are many other formats and many people follow their own unique style to make their CV stand out from other ones. Additionally, there is a tendency of adding visual elements to a CV to make it more interesting to visualize. Opposed to many of the visual elements just being there for aesthetic purpose, there are exceptional cases when someone uses visual elements like graphs or charts to represent important information such as their skills because creating and interpreting graphs or charts encourages critical thinking.

As in most of the cases, these graphs are included in image formats, there is no definitive way to process them without using image processing techniques. Hence, these CVs will be kept out of consideration in the frame of this work as it is beyond the scope, therefore PDF Miner [12] is used as a tool for extracting information from PDF documents. Unlike other PDF-related tools, it focuses entirely on getting and analyzing text data. It includes a PDF converter that can transform PDF files into text format.

3.5 Information Extraction using Natural Language Processing

The cleaned text is passed to Natural Language Processing Algorithm to tokenize the text and identify the phrases and entities in the text. There are various of library options to use and spaCy is chosen as it is designed specifically for this type of applications and it understands a large volume of text. It can be used to build Information Extraction or Natural Language understanding systems or to pre-process text for Deep Learning. The objective is to train the model against reference annotation. Training is an iterative process in which the model's predictions are compared against the reference annotations in order to estimate the gradient of the loss. The gradient of the loss is then used to calculate the gradient of the weights through back-propagation. The gradients indicate how the weight values should be changed so that the model's predictions become more similar to the reference labels over time as shown in figure 5.

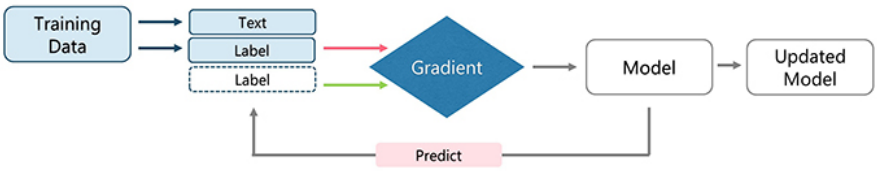

Fig. 5. Model Training

The main idea is to train and obtain the model for each of the parameters that form a certain CV. For example, it could be applied on the Programming Language, Extra-Curricular Activities, Qualification or Degree, Spoken Language, English Language Test, Program Specific Criteria, etc...The result obtained by processing the CV data of each of the trained model is then passed on to the trained classification model to predict the outcome.

For elaboration, an example of applying the model to programming language is presented. To obtain a consistent Named Entity Recognizer model for the programming language, the model is trained with a comprehensive list of language including the various patterns it could appear in (eg: php3.4, php7, Go lang, GoLang). In order to do that a large Dataset is needed based on which the model is trained. Through this research, it was found that Stack Overflow provides a dataset for its question and answers which can be downloaded from Kaggle. Now the dataset is ready spaCy Matcher pattern is used to identify and train the programming languages in each question.

The matching algorithm should follow the steps mentioned in the below figure. It starts by writing exhaustive matcher pattern rules to cover all the criteria for the programming languages, hobbies, languages, degree types in order to obtain the model to be used in the CV evaluation. This algorithm is presented in figure 6.

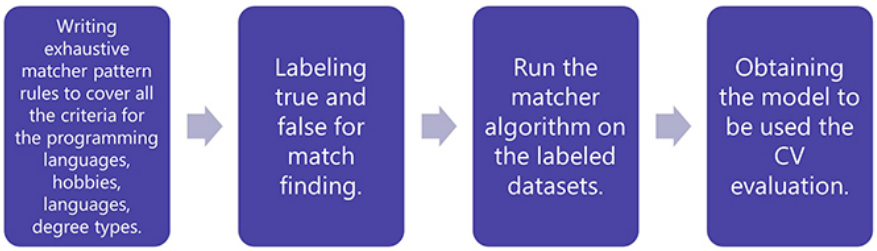

Fig. 6. Matcher Algorithm

Based on various samples testing with Stack Overflow questions dataset over multiple iterations, an exhaustive list of programming languages (like php, java,

python, C), was after obtained as well as patterns across programming language (like php3.4,php7, python3). An interesting scenario for the programming language obtained here was for "GO Lang" since "go" can be a verb in a lot of sentences. The part of speech feature of NLP library is used to help us distinguish the programming language GO from the verb GO. This observation also shows that using NLP along with Regex (Regular Expression) helps efficiently train the algorithm. Once the matcher patterns are defined, labelling is used to evaluate the effectiveness of the pattern. In order to do this, a copy of the Questions Dataset is manually prepared and added in additional column Label. Then this column is manually marked to 1 if a programming language is found for that question and 0 if not. This was done for over 650 records and tested. On running the labelled csv (comma separated values) against the patterns defined, identification of any mistake made in pattern definition was done as well as the pattern to accommodate the changes. The statistical analysis of this model can be determined through the generation of the confusion matrix and classification report. A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. The confusion matrix itself is relatively simple to understand, but the related terminology can be confusing. Below is an example of a confusion matrix that has been created for this example.

A classification report is used to measure the quality of predictions from a classification algorithm. How many predictions are True and how many are False. More specifically, True Positives TP, False Positives FP, True Negatives TN and False Negatives FNare used to predict the metrics of a classification report.

Figure 7 shows an example of a confusion matrix that has been created for this example.

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Fig. 7. Confusion Matrix

Figure 8 shows a classification report for this example.

Precision can be seen as a measure of a classifier's exactness. For each class, it is defined as the ratio of true positives to the sum of true and false positives. Said in another way, "for all instances classified positive, what percent was correct?"

	precision	recall	f1-score	support
0	0.77	0.86	0.81	37584
1	0.84	0.75	0.79	37577
accuracy			0.80	75161
macro avg	0.81	0.80	0.80	75161
weighted avg	0.81	0.80	0.80	75161

Fig. 8. Classification Report

A recall is a measure of the classifier's completeness; the ability of a classifier to correctly find all positive instances. For each class, it is defined as the ratio of true positives to the sum of true positives and false negatives. Said another way, "for all instances that were actually positive, what percent was classified correctly?" The F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. Generally speaking, F1 scores are lower than accuracy measures as they embed precision and recall into their computation. As a rule of thumb, the weighted average of F1 should be used to compare classifier models, not global accuracy.

Support is the number of actual occurrences of the class in the specified Dataset. Imbalanced support in the training data may indicate structural weaknesses in the reported scores of the classifier and could indicate the need for stratified sampling or re-balancing. Support doesn't change between models but instead diagnoses the evaluation process. Moving further after obtaining satisfactory results from the above steps, the programming model is ready to get trained by constructing the model as per the NLP requirement. The obtained trained data is shown in figure 9.

This exercise has been repeated for all parts of the CV (Extra-Curricular activities, English Language Test, Program Specific Criteria...)

This step will provide us with a custom Named Entity Recognition model using spaCy statistical modelling for each parameter that will later be used for information extraction (IE) that seeks out and categorizes specified entities in a body or bodies of texts of the CV. The generated model can recognize a wide range of named or numerical entities, which include qualification, extracurricular activities, language, program specific criteria and English language test. The result obtained by processing CV data to each of the trained models is fed in the trained classification model to predict the outcome.

3.6 Classification and final Output Prediction

The role of the classification model is to draw some conclusion from observed values. Given one or more inputs a classification model will try to predict the

Fig. 9. Trained Data

value of one or more outcomes. Outcomes are labels that can be applied to a dataset. For example, when filtering emails "spam" or "not spam", when looking at transaction data, "fraudulent", or "authorized". Similarly, in our case, for a given result set of Programming Languages, Extra-Curricular Activities, Qualification, Spoken Language, English Language Test and Program Specific criteria the classification model should be able to predict the final outcome. It aims to provide pattern recognition based on previously obtained results. Data is segregated in 2 main Datasets: the training test that is used to train the model to recognize the patterns in the given data for suitable predictions and the test set that contains the previously predicted value. The model is trained using several Machine Learning algorithms and the one with the least error will be chosen as a classifier. In order to classify the data obtained after the CV evaluation under the accepted and rejected categories, a sample of over 500 CVs has been manually evaluated based on the parameters mentioned above. An example of the manual attribution on an accepted and refused CV is mentioned in the figures 10, 11 and 12.

Parameter	Points	Reason
Type of Degree	1	Bachelor in Information Management
Technologies	2	C, C++, Java
Languages Known	1	English
Competitive Entrance	1	TOEFL
Extra-Curricular	1	Football, Cricket, games, music
Program Related criteria	1	IT Entrepreneurship, Supply chain management

Fig. 10. Shortlisted CV

Parameter	Points	Reason
Type of Degree	1	Bachelor of Technology
Technologies	0	None mentioned
Languages Known	1	English
Competitive Entrance	0	None mentioned
Extra-Curricular	0	None mentioned
Program Related criteria	0	None mentioned

Fig. 11. Refused CV

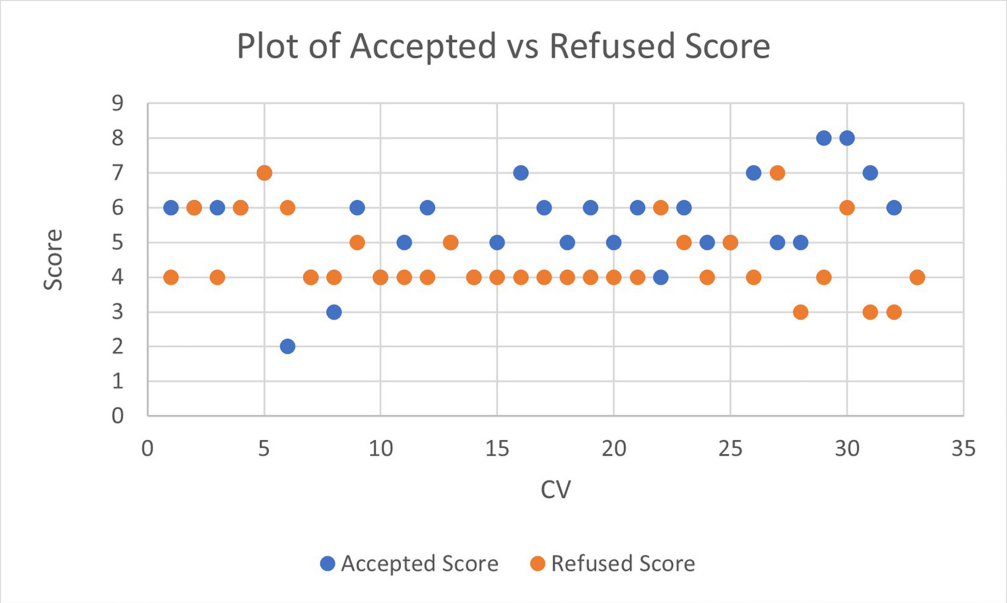

Fig. 12. Accepted/Refused CV

This step provides three classification models for predictive modeling. In machine learning selecting the best hypothesis for a given data is the most interesting part. The accuracy obtained in general is most for Random Forest classifier. Therefore its result is used to determine the final outcome for the CV. Then it is only needed to provide the parameters weighting to the model to predict the final outcome. In a typical supervised learning workflow, evaluation of various combinations of feature sub-spaces, learning algorithms, and hyper-parameters is done before selecting the model that has a satisfactory performance. As mentioned above, cross-validation is a good way for such an assessment in order to avoid over-fitting to the training data.

4 Experimentation

This part of the article is dedicated for the experimentation of the CV evaluation. The experimentation will be dedicated to the CV Information Extraction, the training process, and the classification process using 3 different algorithms as follows:

1. Naïve Bayes Classifier
2. Support Vector Machine SVM
3. Random Forest

The classification is done after finalizing the 5 steps that are mentioned in the previous sections as all of them serve to conduct the classification experiment to obtain the desired outcomes.

4.1 Naïve Bayes Classifier

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem [10]. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e., every pair of features being classified is independent each other. The Dataset is required in two parts namely feature matrix and the response vector as explained below:

- Feature matrix: it contains all the vectors(rows) of Dataset in which each vector consists of the value of dependent features. In the above Dataset, features are 'Programming Language', 'Extra-Curricular Activities', 'Language', 'Qualification', 'Work Experience' and 'Test'.
- Response Vector: it contains the value of class variable (prediction or output) for each row of the feature matrix. In the above Dataset, the class variable name is 'Result'.

The fundamental Naive Bayes assumption is that each feature makes an *independent* and *equal* contribution to the outcome. Bayes' Theorem predicts the probability of an event occurring given the probability of another event that has already occurred. Bayes' theorem is stated mathematically as the following equation:

where A and B are events and P(B) is different from 0.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- Basically, the probability of event A is to be predicted, given the event B is true. Event B is also termed as evidence.
- P(A) is the priori of A (the prior probability, i.e., Probability of event before evidence is seen). The evidence is an attribute value of an unknown instance (here, it is event B).
- P(A—B) is a posteriori probability of B, i.e., probability of event after evidence is seen.

With regards to the used Dataset, Bayes' theorem is applied in the following way:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

where, y is class variable and X is a dependent feature vector (of size n). where: To add classification, an example of a feature vector and corresponding class variable can be: (refer 1st row of dataset)

X = (0(Programming Language), 0(Extra Curricular), 1(Languages) , 1(Program Specific Criteria) ,1(Qualification) ,0(Test) ,1(Work Experience)) y = Yes

$$X = (x_1, x_2, x_3, ..., x_n)$$

For simplicity data from each label is assigned to be drawn from a simple Gaussian distribution. The likelihood of the features is assumed to be Gaussian, hence, conditional probability is given by:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right)$$

The used dataset involves predicting the CV result given individual parameters like Technologies, Degree, Language, Extra-Curricular activities and Program Specific criteria. It is therefore a multi-class classification problem. There are around 40 observations with 6 input variables and 1 output variable. A sample of the input variables is listed below:

Using sci-kit learn, it is now possible to train and test the model prediction for Gaussian Naïve Bayes as shown in the figure 13.

['1', '1', '1', '0', '1', '0', 'Accepted'],

['1', '2', '2', '0', '1', '0', 'Accepted'],

['1', '3', '1', '0', '1', '1', 'Accepted'],

['1', '2', '1', '0', '1', '0', 'Accepted'],

['2', '1', '1', '0', '1', '1', 'Accepted']

['1', '1', '1', '0', '1', '1', 'Accepted'],

```
In [25]: from sklearn.model_selection import train_test_split
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20)

In [26]: from sklearn.naive_bayes import GaussianNB
         gnb = GaussianNB()
         gnb.fit(X_train, y_train)

Out[26]: GaussianNB()

In [27]: y_pred = gnb.predict(X_test)

In [28]: from sklearn.metrics import classification_report,confusion_matrix
         print(confusion_matrix(y_test,y_pred))

         [[ 9 12]
          [12 34]]

In [29]: from sklearn.metrics import classification_report
         print(classification_report(y_test, y_pred))

                       precision    recall  f1-score   support

                    0       0.43      0.43      0.43        21
                    1       0.74      0.74      0.74        46

             accuracy                           0.64        67
            macro avg       0.58      0.58      0.58        67
         weighted avg       0.64      0.64      0.64        67
```

Fig. 13. Model Training using Gaussian Naïve Bayes

It was observed that the algorithm generates a Reject response if it matches a similar combination of already existing rejected candidates and Accepted otherwise.

4.2 Support Vector Machine(SVM)

The objective of the support vector machine algorithm [13] is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points as shown in the figure 14.

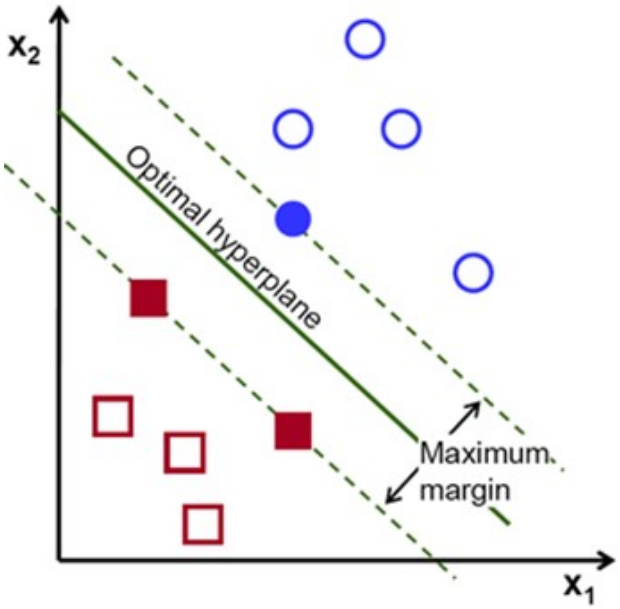

Fig. 14. Support Vector Machine (SVM)

To separate the two classes of data points, there are many possible hyperplanes that could be chosen. The objective of this paper is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features. If the number of input features is 2, then the hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane. Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support

vectors, the margin of the classifier is maximized. Deleting the support vectors changes the position of the hyperplane. These are the points that help build the SVM.

In the SVM algorithm, maximizing the margin between the data points and the hyperplane is looked for. The loss function that helps maximize the margin is hinge loss.

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else} \end{cases}$$

The cost is 0 if the predicted value and the actual value are of the same sign. If they are not, then calculate the loss value is then calculated. A regularization parameter is added to the cost function. The objective of the regularization parameter is to balance the margin maximization and loss. After adding the regularization parameter, the cost functions looks as below.

$$min_w \lambda \parallel w \parallel^2 + \sum_{i=1}^{n} (1 - y_i \langle x_i, w \rangle)_+$$

Below, the implementation of SVM in python is depicted using the scikit library using Sigmoid Kernel as it was more accurate compared to linear, polynomial linear and the Gaussian kernels:

It was observed that the algorithm generates an Accept response in almost all test cases.

4.3 Random Forest Classifier

Random Forest is a supervised learning algorithm [1]. The "forest" builds, is an ensemble of decision trees, usually trained with the "bagging" method. The general idea of the bagging method is that a combination of learning models increases the overall result.

Random forests also offers a good feature selection indicator. Scikit-learn provides an extra variable with the model, which shows the relative importance or contribution of each feature in the prediction. It automatically computes the relevance score of each feature in the training phase. Then, it scales the relevance down so that the sum of all scores is 1.

This score choose the most important features and drop the least important ones for model building.

```
In [11]: from sklearn.model_selection import train_test_split
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20)

In [12]: from sklearn.svm import SVC
         svclassifier = SVC(kernel='sigmoid')
         svclassifier.fit(X_train, y_train)

Out[12]: SVC(kernel='sigmoid')

In [13]: y_pred = svclassifier.predict(X_test)

In [14]: from sklearn.metrics import classification_report,confusion_matrix
         print(confusion_matrix(y_test,y_pred))

         [[ 3 12]
          [ 9 43]]

In [15]: from sklearn.metrics import classification_report
         print(classification_report(y_test, y_pred))

                       precision    recall  f1-score   support

                    0       0.25      0.20      0.22        15
                    1       0.78      0.83      0.80        52

             accuracy                           0.69        67
            macro avg       0.52      0.51      0.51        67
         weighted avg       0.66      0.69      0.67        67
```

Fig. 15. Model Training using SVM

Random Forest uses GINI importance or mean decrease in impurity (MDI) to calculate the importance of each feature. Gini importance is also known as the total decrease in node impurity. This is how much the model fit or accuracy decreases when you drop a variable. The larger the decrease, the more significant the variable is. Here, the mean decrease is a significant parameter for variable selection. The Gini index can describe the overall explanatory power of the variables. Figure 16 shows the implementation of Random Forest classifier using scikit learn [14]:

5 Conclusion and Future Work

As the CV evaluation is a part of a decision support system that we applied to the admission of international students in the French Higher Education systems, the results obtained in the actual classifier is cross-validated with results from the online video interviews results.

```
In [67]: from sklearn.ensemble import RandomForestClassifier
         rfc = RandomForestClassifier(random_state=0)
         rfc.fit(X_train, y_train)

Out[67]: RandomForestClassifier(random_state=0)

In [68]: y_pred = rfc.predict(X_test)

In [69]: from sklearn.metrics import classification_report,confusion_matrix
         print(confusion_matrix(y_test,y_pred))

         [[ 5  7]
          [ 8 47]]

In [70]: from sklearn.metrics import classification_report
         print(classification_report(y_test, y_pred))

                       precision    recall  f1-score   support

                    0       0.38      0.42      0.40        12
                    1       0.87      0.85      0.86        55

             accuracy                           0.78        67
            macro avg       0.63      0.64      0.63        67
         weighted avg       0.78      0.78      0.78        67
```

Fig. 16. Model Training using Random Forest

References

1. Data Camp. Understanding random forests classifiers in python: , https://www.datacamp.com/community/tutorials/random-forests-classifier-python.
2. EPITA. Epita international: https://www.epita.fr/en/apply-online/.
3. EPITA. Epita masters: https://https://www.epita.fr/en/degree-programs-english/.
4. CAMPUS FRANCE. Institutional: https://www.campusfrance.org/en/institutions.
5. Ashish S.. Ganesh K., Ritvik J. A novel job portal with resume evaluation system based on text mining and nlp techniques. *Journal of Critical Reviews*, pages 1234–1236, 2020.
6. Google. Kaggle:, https://www.kaggle.com/.
7. Google. stackoverflow: https://www.kaggle.com/stackoverflow/stackoverflow.
8. DUFOUR J.C. Aix-marseille university, https://sesstim.univ-amu.fr/sites.
9. Piskorski J. Kaczmarek T., Kowalkiewicz M. Ainformation extraction from cv. 2015.
10. Machine Learning Mastery. Classification: , https://machinelearningmastery.com/classification-as-conditional-probability-and-the-naive-bayes-algorithm.
11. Stack Overflow. Stack overflow nlp:, https://stackoverflow.com/.
12. Python. Pdfminer: https://pypi.org/project/pdfminer.
13. Towards Data Science. Support vector machine — introduction to machine learning algorithms: , https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47.

14. scikit learn. Machine learning in python: , https://scikit-learn.org/.
15. SpaCy. Industrial-strength nlp:, https://spacy.io/.