Loan Prediction System

Aishwarya Patil
Dept. of Computer Science
Drexel University
Philadelphia, PA, United States of America
asp344@drexel.edu

Sravya Atluri
Dept. of Computer Science
Drexel University
Philadelphia, PA, United States of America
sa3638@drexel.edu

Atharva Deshpande
Dept. of Computer Science
Drexel University
Philadelphia, PA, United States of America
ad3756@drexel.edu

Nishit De
Dept. of Computer Science
Drexel University
Philadelphia, PA, United States of America
npd57@drexel.edu

***Abstract*—The only way to find out if you'll be approved for a loan is when you apply. Yet that leaves a mark on your credit file that other lenders can see, potentially affecting your ability to get future credit. In this project, we take user information on various elements such as loan amount, credit history, marital status, etc in a form and use that data to predict whether the user will get a loan or not without affecting your credit in any way. While the decision may not always coincide with the decision of the bank but it will still give the user a baseline on what to expect. The machine was trained using previous cases on whether a bank grants a loan depending on the inputs in the data set. In the backend, we also used various algorithms and compared them against one another to check which algorithm best fits the dataset. The models used: Logistic Regression, Naive Bayes, Support Vector Machine, Decision Tree and Random Forest.**

***Index Terms*—loan, prediction, accuracy, algorithms, logistic regression**

I. Introduction

Almost in every students' life, there comes a point where they wonder how to pay for higher studies. This financial implication may appear to be a significant challenge, and there are many ways to get this resolved - from scholarships to applying for bank loans. That would mean a massive pressure on the banks to validate everyone's eligibility for the loan, which would involve validating a lot of factors like loan history, the riskiness of the borrower, credit score, financial background, etc. But the pressure is not just restricted to the lenders; the borrowers are under equal pressure of repaying the debt. The possibility of losing your assets kept as a mortgage for the loan could be terrifying. This has caused a tremendous amount of anxiety in people. The American Psychological Association's annual "Stress in America" survey consistently cites money as the most significant source of stress by about two-thirds of respondents. As our attempt to reduce the stress and anxiety towards dealing with the debt, we propose a Loan Predicting K Nearest Learning algorithm for the borrowers to predict their eligibility for the loan and determine the approval for the same. This way the borrowers can have an idea of whether or not they will be able to repay the loan.

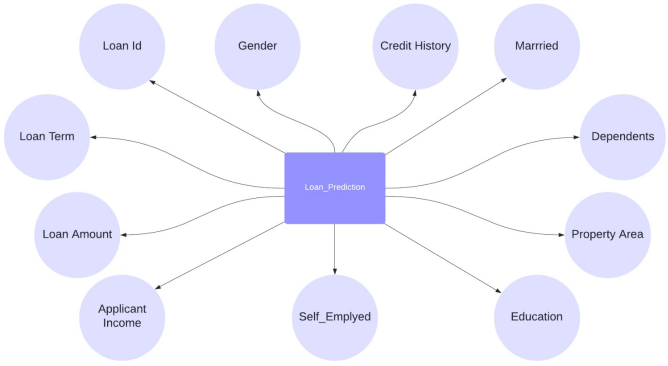

Fig. 1. Features of the Dataset

II. Related Work

The factors that the bank consider when the borrowers apply for loans majorly consist of a good credit score. But a credit score is not the only thing that they look for while deciding the eligibility of the borrower. The lender can go a lot pickier about the eligibility because if the borrower fails to repay the loan, they are at a loss. Thus, the banks need to carefully scrutinize all the factors about an individual that would determine the risk involved. Ira Makrygianni and Angelos Markopoulos have proposed a model that calculates the probability or the uncertainty associated with the non-repayment of the loan by the borrower using Artificial Neural Network, which has been trained on the real-time historical client data [4]. To have the trust between the lender and the borrower, traditionally, there was a need for a third party mediator intervening in the transactions and ensuring that either of the parties is not breaching the contracts. An attempt to replace this need of a third-party mediator was made by Qi Yang, Xiao Zeng, Yu Zhang, and Wei Hu. They digitalized the documented contracts and with the help of blockchain as storage for a decentralized database that took

care of transactions and contract breaches [6]. To Majorly, the lender wants to ensure that the loan can be repaid within the given time limit. Thus, the eligibility could variedly depend on the borrower's income, employment history, any previous investments, repayment history, etc. Some personal factors also affect the process and decision making such as marital status, dependents, education, etc.

On the flip side, whenever a customer has to apply for a loan, the first thing that they consider, which is also of prime importance, is the interest rate. Along with that, there are several other permutations and combinations that a borrower has to go through before making a decision to apply for a loan. Thus, in their decision-making process, they have to take into account various factors such as the leverage for the amount of loan they are going to request for - mortgage, the tenure for the repayment, the amount of the loan, etc.

Another element related to loan prediction is the network of guarantors that are given while filing for a loan. A guarantor, as the name suggests, is a guarantee that the loan must be repaid to the bank. If a borrower is unable to pay the loan, then the guarantor is the one who has to repay it in his place. But when taking into account the extensive network of guarantor borrower relations, one loophole can lead to a chain reaction of inconsistencies. Dawei Cheng, Yiyi Zhang, Fangzhou Yang, Yi Tu, Zhibin Niu and Liqing Zhang proposed a method to better equip the bank with such inconsistencies. They attempted to predict bank guarantee loan defaults by adding an objectively temporal network structure using an end-to-end learning frame. They designed and implemented a Dynamic Default Prediction Framework (DDPF), which enables the model to represent temporal network structures. Their research was evaluated in a real-world loan risk management system and it was deemed to be successful [3].

Credit history is a crucial factor in determining whether a borrower can get a loan or not. But credit history can change substantially due to insufficient credit information. Many countries don't even have a well-established credit system hence get an accurate credit history is easier said than done. Beibei Niu, Jinzheng Ren, and Xiaotao Li suggested using a credit scoring system based on social network information. They also used logistic regression to find a relation between social network information and loan default. Since we are also using logistic regression in loan prediction it increases the compatibility of the two types of research. The collected a total of 21,036 P2P loan samples were collected from mobile network operators and a Chinese P2P lending platform. This makes the research pretty reliable as the paper concluded there was a significant relationship between social network information and loan default prediction [5]. Thus, the correlation between the two types of research can definitely improve the performance of our research.

Another paper that was an inspiration for this project was the " Exploratory analysis on prediction of loan privilege for customers using Random Forest" [2] this is a similar project with a more focused scope. In this project, they only use the Random Forest algorithm in order to determine whether a borrower gets a loan or not. In our project, we have broadened the scope by using multiple algorithms and then using the one with the best accuracy in terms of the predictions. Having a wider variety of algorithms makes the project more robust due to trial and error.

Another research that would go hand in hand with ours is the " Developing Prediction Model of Loan Risks in Banks using Data Mining", in this project they use data mining to predicts loan risks from a large dataset. This project gives the banks a better evaluation if a customer is eligible for a loan or not. If we can use the same ideology in the future implementation of this project, this will give us a better idea of how the banks evaluate the customers which in turn helps us predicts whether a customer will get a loan or not [1]. This gives us a look into the bank's perspective of how they evaluate the customer which will help us improve our model by improving the accuracy of the predictions.

III. Methodology

In this project, we work towards automating the eligibility process of the customer to receive a loan or not, depending on the details provided by the customer himself. This process involves identifying various factors such as the marital status of the customer, how many dependents does he/she have, is he/she a college graduate or not, annual income, etc. We do so by taking into consideration a dataset consisting of all such features, and training our models with these data in order for the system to help predict it.

A. Dataset Processing

The dataset that we have used for our purpose has been sourced from 'Kaggle' and goes by the name of 'Loan Prediction Problem Dataset.' The dataset consists of a total of 13 feature columns and a sum of 981 records. These 981 records are then split into train and test sets, with the train set having 614 records and the test set having 367 records.

Transforming raw data into a readable format can be said as the definition of data preprocessing. Often, datasets in the real-world are inconsistent i.e. contain missing values, outliers and are likely to contain many errors.

We can deal with missing values in various ways. We always go ahead and delete the rows but that can cause information loss and might result in poor working of the model especially when the dataset is small. Keep in mind that our dataset is small, we have eliminated this option. The second option was to assign a unique category. This method negates the loss of data by adding an unique category but would add less variance which would affect the performance. We have decided to replace missing values with mean/median/mode values because it is considered a better approach when the dataset is small and would also prevent data loss.

The features can be divided into categorical and numerical. Missing values in categorical features can be replaced with the mode of the values in the column. And, in the case of numerical features we can use mean or median. Using mean isn't generally suggested because there can always be outliers present in the data.

B. Feature Engineering

In this paper, we present an application that predicts whether the users are eligible for the loan or not. The application allows the users to enter their details, which is then taken as a test data by the model that has been trained using the Loan Prediction Problem Dataset. The model works really well with the data provided by the user, the accuracy of the model using Logistic Regression is above 80%. Amongst all the algorithms used Logistic Regression has the highest accuracy, followed by Naive Bayes, Random Forest, Decision Tree and SVM.

a) EMI: Idea behind making this variable is that people who have high EMI's might find it difficult to pay back the loan. We can calculate the EMI by taking the ratio of loan amount with respect to loan amount term.

b) Total Income: Instead of considering your coapplicant's income as a different feature we can combime them. This indicates, if the total income is high, chances of loan approval might also be high.

c) Balanced Income: Idea behind creating this variable is that if this value is high, the chances are high that a person will repay the loan and hence increasing the chances of loan approval.

C. Building Model

To check how robust our model is to unseen data, we use validation. It is a technique that involves reserving a particular sample of a dataset on which you do not train the model. In our project, we apply cross-validation to five different algorithms to see which one works the best in terms of accuracy. The algorithms implemented in our project are as follows:

- Logistic Regression
- Decision Tree
- Random Forest Classifier
- Naive Bayes
- Support Vector Machine

a) Logistic Regression: Logistic regression is a statistical analysis method used to predict a data value based on prior observations of a data set. A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables [7]. The purpose of logistic regression is to estimate the probabilities of events, including determining a relationship between features and the probabilities of particular outcomes.

b) Decision Tree: A tree can be "learned" by splitting the source set into subsets based on an attribute value test [8]. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions. The construction of a decision tree classifier does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery, and can handle high dimensional data.

c) Random Forest Classifier: Random Forest is a tree based bootstrapping algorithm wherein a certain no. of weak learners (decision trees) are combined to make a powerful prediction model [9]. For every individual learner, a random sample of rows and a few randomly chosen variables are used to build a decision tree model. Final prediction can be a function of all the predictions made by the individual learners. In case of a regression problem, the final prediction can be mean of all the predictions.

d) Naive Bayes: 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 of each other. The dataset is divided into two parts, namely, the feature matrix and the response vector. Feature matrix contains all the vectors of the dataset in which each vector consists of the value of dependent features. In the above dataset, features are 'Married', 'Dependants', 'Education' and 'Credit history', etc. Response vector contains the value of the class variable(prediction or output) for each row of the feature matrix. In the above dataset, the class variable name is 'Loan Approval'.

e) Support Vector Machine: A support-vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class, since in general the larger the margin, the lower the generalization error of the classifier [11].

Additionally, we have adopted stratified K fold cross validation as we have a small dataset for prediction. K-fold Cross-validation is a re-sampling procedure used to evaluate machine learning models on a limited data sample, where K means the number of groups to split the data into. By implementing we use all the data for building the model which achieves better accuracy. In the table below we can see the accuracies that various algorithms achieve with and without K fold cross validation.

Algorithm	With K-fold	Without K-fold
Logistic Regression	80.61	77.29
Decision Tree	73.30	72.97
Random Forest Classifier	77.35	73.51
Naive Bayes	79.62	78.91
Support Vector Machine	68.73	64.41

Table 1: Comparing With or Without K-fold

D. Observation

In our paper, we have used Logistic Regression over Decision Trees, Random Forest and Naive Bayes for one most primary reason - Accuracy. Logistic Regression has the highest accuracy (80.45%), followed by Naive Bayes (79.79%), Random Forest (77%), Decision Tree (70.36%) and SVM (68.73%).

1) Advantages of Logistic Regression over Decision Trees Algorithm:

- It is slightly faster.
- Not prone to overfitting.
- Robust to noise.

2) Advantages of Logistic Regression over Random Forest Classifier:

- Simple interpretation of the explanatory variables.
- Not prone to overfitting.
- Need to choose the number of trees.
- Comparatively less complex.
- Less time consuming.

3) Advantages of Logistic Regression over Naive Bayes:

- It works decently when the features are correlated.
- The model assumes all features as conditionally independent. Thus if the features are dependent on one another, the prediction might be poor.

4) Advantages of Logistic Regression over Support Vector Machine:

- It gives calculated probabilities due to which interpretation is better.
- Comparatively better prediction.

IV. Approach

Loan Predictor focuses on predicting the eligibility of the users for getting a loan based on the inputs provided by them. The way the application has been designed is shown as follows.

The website has 3 screens : Home page, Prediction page and About Us page. Users are first navigated to the Home page of the website.

Here the users can click on the Begin button to proceed with the prediction of the eligibility. “prediction.html” page has been designed to serve the purpose. This page is a form that asks for users input for various fields that are used in calculating the eligibility based on the Logistic Regression.

All the fields on this page are mandatory for the users to enter. The fields are, First Name, Last Name, Gender, Married, Dependents, Education, Self-Employed, Applicant Income, Co-Applicant Income, Loan Amount, Loan Amount Term, Credit Score and Property. Restrictions have been put on these fields to be as mandatory for the users to input. Credit Score has to be put between 0-840.

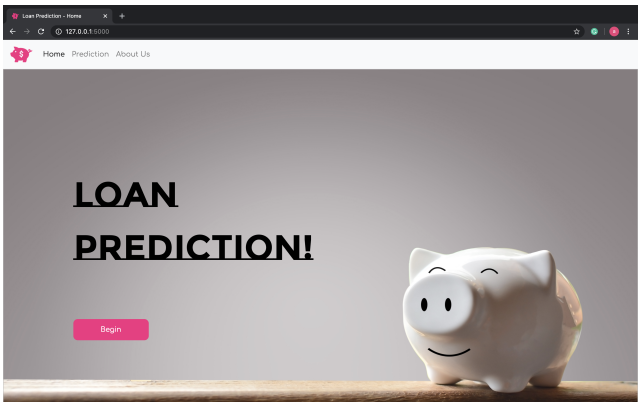

Fig. 2. Home Page

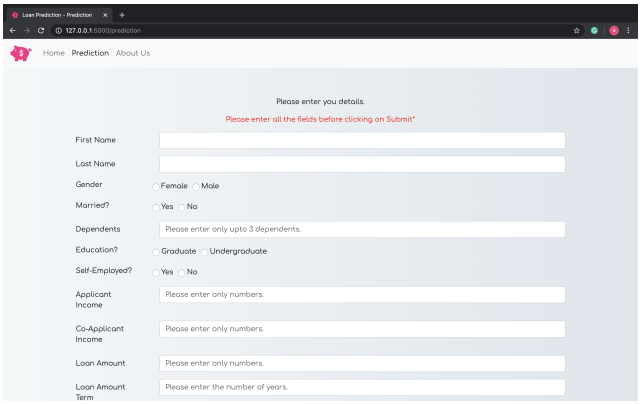

Fig. 3. Customer Detail Form Pt. 1

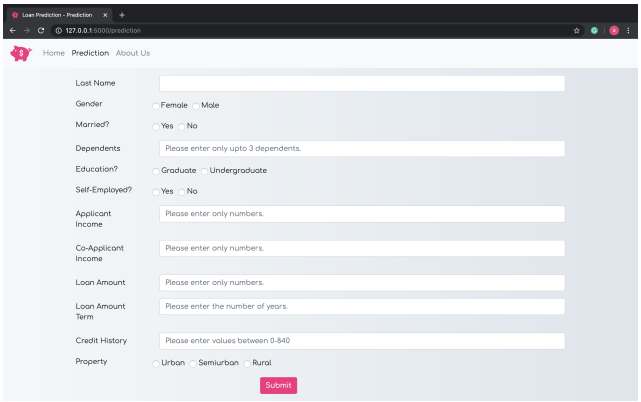

Fig. 4. Customer Detail Form Pt. 2

When the users click on the Submit button, after successfully filling the form, they are navigated to “thankyou.html” or “noThankyou.html” depending on whether they are eligible for the loan or not, respectively.

“about.html” is designed for the users to understand what the application is all about. the About Us tab in the menu.

A. Implementation

Table in Database - loanPrediction.db: ”userInfo” stores all the user input fetched from “prediction.html”

The Loan Prediction has been covered in the website under three menu items; namely, Home, Prediction and About Us. All the screens have one single base skeleton - “base.html” which contains the navigation bar with the menu items as shown in Figure (1) along with the Loan Prediction logo. This template remains constant throughout the website.

The prediction begins on the Home page under the Home menu. “index.html” has been designed for the home page, which consists of an image for the website followed by a button “Begin” where users can begin their validation of the eligibility towards getting the loan.

The Prediction page selection.html consists of a form with 'method="POST" ' and an 'action=/ create_userDetails' which sends the user input to a function create_userDetails()' in run.py file. In this function, the model is trained using Logistic Regression, imported from the Sklearn library. The input boxes on the page have been given an attribute "name" which stores the users' input. Eventually, the function 'create_selectedIngr()' stores the input into a variable and calls another function 'create_profile()' in the db.py file. This function calculates EMI based on the Loan Amount and Loan Amount Term entered by the users,

$$EMI = \frac{LoanAmount}{LoanAmountTerm}$$

As and when the user hits the "Submit" button on the Prediction page, this function inserts the data as entered by the users into the database table - "userInfo". This user information is then again saved into a variable in the create_userDetails() function in the run.py file. Using this data, the users' Total Income and Balance income are calculated,

$$TotalIncome = ApplicantIncome + Co-ApplicantIncome$$

$$BalanceIncome = TotalIncome - (EMI \times 1000)$$

The updated data is used to create a successful test data in the same format as data has been trained using X_train.csv and Y_train.csv. This test data is then passed as a parameter to the model to predict the eligibility. If the model predicts the output as 1, that means the users are eligible for the loan. In that case, users are navigated to "thankyou.html" page; otherwise they are navigated to "noThankyou.html".

"thankyou.html" fetches the first name of the user from the database to be displayed on the page along with a message, "Thank you XYZ, for using our application. You are eligible for the loan. This page has an additional button "Begin Again" that navigates the users back to the Prediction page. Similarly, "noThankyou.html" fetches the first name of the user from the database to be displayed on the page along with a message, "Thank you XYZ,for using our application. Sorry, but you are not eligible for the loan." This page also has an additional button "Try Again" that navigates the users back to the Prediction page.

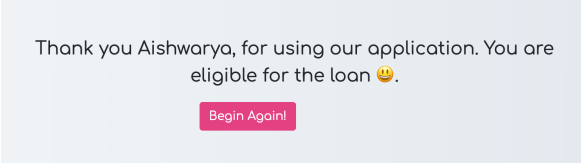

Fig. 5. Loan Approved Message

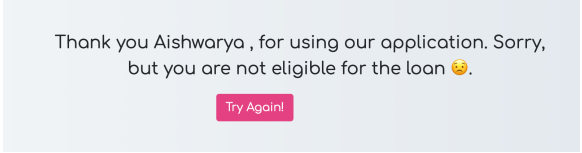

Fig. 6. Loan Not Approved Message

All the menu items, "Begin Again" and "Try Again" buttons have an onclick function assigned which clears the userInfo table in the database.

V. Testing

A. Experiment

Initially, we had planned on experimenting with a broader set of audiences, but due to the current situation, we restricted ourselves to our roommates. For the purpose of this project, we interviewed three people as below:

a) Pratik Joglekar: A Senior Product Designer at MassMutual, liked the interface that we had built. He also liked the idea of borrowers getting an opportunity to check their eligibility on their own.

Fig. 7. Subject 1: Pratik Joglekar

b) Sarika Joglekar: A UX Designer at The Dice Group, suggested a few changes that she felt were confusing for her as she was entering the details for the prediction.

Fig. 8. Subject 2: Sarika Joglekar

c) Niranjan Patil: A student at Jefferson University, liked the purpose of our application and since he has had a recent interaction with the bank for his student loan, he suggested a few changes in terms of the data to be entered by the user.

Fig. 9. Subject 3: Niranjan Patil

B. Result

It was interesting to know the feedback that the above-mentioned candidates had for our application.

a) Pratik Joglekar: Pratik suggested that the dataset could also include the country the prediction is being done for, that is, the dataset could be country-specific. For eg, Credit Score exists only in the US.

b) Sarika Joglekar: Sarika concluded that she would love it if she knew the currency the dataset is currently related to.

c) Niranjan Patil: Niranjan suggested that the inputs from the users could be more detailed than what it is currently. He inferred that the co-applicants relationship with the applicant should also affect the percentage of the prediction. Furthermore, he also suggested that even if the income of the applicant is less than what it is expected for the eligibility, having an asset to keep as a mortgage could also increase the chances of getting a loan.

Thus, from all the feedback that we received, we can infer that the dataset could be more comprehensive in range, and various features could be included more than currently being used.

VI. Future Scope

Four directions could be further enhanced in this application. The first one would be trying to implement a wider variety of algorithms. For the scope of this paper, we have compared within four algorithms, viz. Logistic Regression, Decision Trees, Random Forest, and Naive Bayes and evaluated that the Logistic Regression gives out the maximum accuracy over the rest of them. As a future scope, we will try out other algorithms apart from the mentioned four above, to validate whether the accuracy can be achieved more than what we are getting with Logistic Regression. The second one would be to use an extensive dataset for training the model, with features much larger in number than the dataset currently being used (taking care of the overfitting). Adding more features would eventually increase the diversity of the data, thereby helping us improve the accuracy of predicting eligibility. The third one being no restriction on the number of the dependents being entered by the users. For this project, we have limited the users to enter dependents only up to three. This will facilitate the users with a more user-friendly application. The fourth one would be a bank-specific database. This will allow the users to select the bank of their choice in order to find out their eligibility of the loan.

VII. Conclusion

In this paper, we present an application that predicts whether the users are eligible for the loan or not. The application allows the users to enter their details, which is then taken as a test data by the model that has been trained using the Loan Prediction Problem Dataset. The model works really well with the data provided by the user, the accuracy of the model using Logistic Regression is above 80.61%. Amongst all the algorithms used Logistic Regression has the highest accuracy (80.61%), followed by Naive Bayes (79.62%), Random Forest (77.35%), Decision Tree (73.30%) and SVM (68.73%).

References

[1] Aboobyda Jafar Hamid and Tarig Mohammed Ahmed-" Developing prediction model of loan risks in banks using data mining", 2016.

[2] K. Ulaga Priya, S. Pushpa, K. Kalaivani, A. Sartiha" Exploratory analysis on prediction of loan privilege for customers using Random Forest", 2018.

[3] Dawei Cheng, Yiyi Zhang, Fangzhou Yang, Yi Tu, Zhibin Niu,Liqing Zhang "A Dynamic Default Prediction Framework for Networked-guarantee Loans."

[4] Ira Makrygianni and Angelos Markopoulos "Loan Evaluation Applying Artificial Neural Networks"

[5] Beibei Niu, Jinzheng Ren and Xiaotao Li "Credit Scoring Using Machine Learning by Combing Social Network Information: Evidence from Peer-to-Peer Lending", 2019.

[6] Qi Yang, Xiao Zeng, Yu Zhang, Wei HuNew "Loan System Based on Smart Contract"

[7] Rouse, Margaret. *Search Business Analytics*. Definition of Logistic Regression, www.searchbusinessanalytics.techtarget.com/definition/logistic-regression, Accessed March 23, 2020.

[8] Gupta, Saloni. *Geeks for Geeks*. Decision Tree, www.geeksforgeeks.org/decision-tree, Accessed March 23, 2020.

[9] *Wikipedia*. Random Forest Classifier, en.wikipedia.org/wiki/Random_forest, Accessed March 23, 2020.

[10] *Geeks for Geeks*. Naive Bayes, www.geeksforgeeks.org/naive-bayes-classifiers, Accessed March 23, 2020.

[11] *Wikipedia*. Support Vector Machine, en.wikipedia.org/wiki/Support-vector_machine, Accessed March 23, 2020.