Support Vector Machine for Classification

Jeshlin Donna J
Department of Metallurgical and Materials Engineering
Indian Institute of Technology Madras
Chennai, India
mm20b029@smail.iitm.ac.in

***Abstract*—The key task of this paper is to evaluate whether a candidate is a Pulsar Star or not. The aim is to use and apply Support Vector Machine Classifiers and perform Exploratory data analysis to uncover previously unknown or hidden facts in the data set available, visualize them using Data Visualization and make predictions on the validation data. This is followed by model evaluation on the Support Vector Machine model. The task is to conclude the study of which types of candidates are more likely to be a Pulsar star.**

***Index Terms*—Support Vector Machines, Exploratory data analysis, Data Visualization, Model Evaluation**

I. INTRODUCTION

The key task is to evaluate whether a candidate is a Pulsar Star or not. The aim is to use and apply Support Vector Machine and perform Exploratory data analysis to uncover previously unknown or hidden facts in the data set available, visualize them using Data Visualization and make predictions on the validation data. The task is to conclude the study of which types of candidates are more likely to be a Pulsar star.

In machine learning, support-vector machines are supervised learning models that analyze data for classification and regression analysis. SVMs are one of the most robust prediction methods, being based on statistical learning frameworks. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. SVM maps training examples to points in space so as to maximise the width of the gap between the two categories. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

The goal of this research paper is to correctly predict which type of candidates are pulsar stars given a set of information. A predictive classification model based on a Support Vector Machine pipeline was built using the given data so that the 'Mean of the integrated profile', 'Standard deviation of the integrated profile', 'Excess kurtosis of the integrated profile', and other features contributed to the candidates chances of being a pulsar star.

In this paper, we conducted exploratory data analysis to excavate different knowledge existing in the given data set and to perceive the impact of every field with respect to the possibility of the candidate being a pulsar star by the use of 'target_class' field analysis in between each field of the data set. Data cleaning and pre-processing was done, data analysis was performed and, a SVM Classifier model was built and trained on the training data, and the accuracy was tested on the validation dataset. After analyzing the dataset, we discovered insights and information on what the pulsar star candidates had in common. We were also able to predict if a particular candidate with certain feature values would be a pulsar star by applying the trained model pipeline.

II. SUPPORT VECTOR MACHINE

Suppose some given data points each belong to one of two classes, and the goal is to decide which class a new data point will be in. In the case of support-vector machines, a data point is viewed as a p-dimensional vector (a list of p numbers), and we want to know whether we can separate such points with a $(p-1)$-dimensional hyperplane. This is called a linear classifier. There are many hyperplanes that might classify the data. One reasonable choice as the best hyperplane is the one that represents the largest separation, or margin, between the two classes. So we choose the hyperplane so that the distance from it to the nearest data point on each side is maximized. If such a hyperplane exists, it is known as the maximum-margin hyperplane and the linear classifier it defines is known as a maximum-margin classifier; or equivalently, the perceptron of optimal stability.

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. A good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin, the lower the generalization error of the classifier.

Whereas the original problem may be stated in a finite-dimensional space, it often happens that the sets to

discriminate are not linearly separable in that space. For this reason, it was proposed that the original finite-dimensional space be mapped into a much higher-dimensional space, presumably making the separation easier in that space. To keep the computational load reasonable, the mappings used by SVM schemes are designed to ensure that dot products of pairs of input data vectors may be computed easily in terms of the variables in the original space, by defining them in terms of a kernel function $k(x, y)$ selected to suit the problem.

The hyperplanes in the higher-dimensional space are defined as the set of points whose dot product with a vector in that space is constant, where such a set of vectors is an orthogonal (and thus minimal) set of vectors that defines a hyperplane. The vectors defining the hyperplanes can be chosen to be linear combinations with parametersα_i of images of feature vectors x_i that occur in the data base. With this choice of a hyperplane, the points x in the feature space that are mapped into the hyperplane are defined by the relation $\sum_i \alpha_i k(x_i, x) =$ constant. Note that if $k(x, y)$ becomes small as y grows further away from x, each term in the sum measures the degree of closeness of the test point x to the corresponding data base point x_i. In this way, the sum of kernels above can be used to measure the relative nearness of each test point to the data points originating in one or the other of the sets to be discriminated. Note the fact that the set of points x mapped into any hyperplane can be quite convoluted as a result, allowing much more complex discrimination between sets that are not convex at all in the original space.

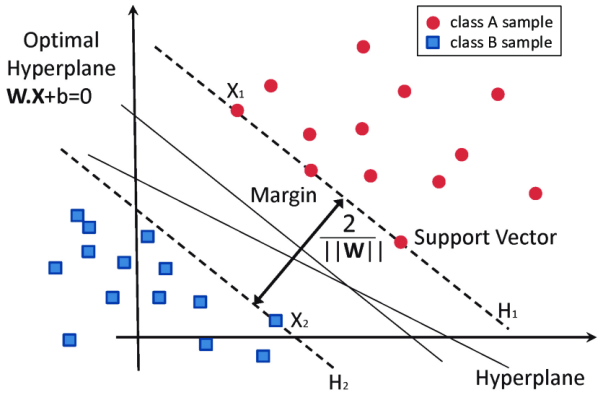

Fig. 1. Support Vector Machine

A. Linear SVM

We are given a training dataset of n points of the form: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$,

where the y_i are either 1 or 1, each indicating the class to which the point $\mathbf{x}_i$ belongs. Each $\mathbf{x}_i$ is a p-dimensional real vector. We want to find the "maximum-margin hyperplane" that divides the group of points $\mathbf{x}_i$ for which $y_i = 1$ from the group of points for which $y_i = -1$, which is defined so that the distance between the hyperplane and the nearest point $\mathbf{x}_i$ from either group is maximized.

Any hyperplane can be written as the set of points $\mathbf{x}$ satisfying

$:\mathbf{w}^T\mathbf{x} - b = 0,$

where $\mathbf{w}$ is the normal vector to the hyperplane. The parameter $\frac{b}{\|\mathbf{w}\|}$ determines the offset of the hyperplane from the origin along the normal vector $\mathbf{w}$.

1) Hard-margin: If the training data is linearly separable, we can select two parallel hyperplanes that separate the two classes of data, so that the distance between them is as large as possible. The region bounded by these two hyperplanes is called the "margin", and the maximum-margin hyperplane is the hyperplane that lies halfway between them. With a normalized or standardized dataset, these hyperplanes can be described by the equations : $\mathbf{w}^T\mathbf{x} - b = 1$ (anything on or above this boundary is of one class, with label 1) and : $\mathbf{w}^T\mathbf{x} - b = -1$ (anything on or below this boundary is of the other class, with label 1). Geometrically, the distance between these two hyperplanes is $\frac{2}{\|\mathbf{w}\|}$, so to maximize the distance between the planes we want to minimize $\|\mathbf{w}\|$. The distance is computed using the distance from a point to a plane equation. We also have to prevent data points from falling into the margin, we add the following constraint: for each i either : $\mathbf{w}^T\mathbf{x}_i - b \geq 1$, if $y_i = 1$, or : $\mathbf{w}^T\mathbf{x}_i - b \leq -1$, if $y_i = -1$. These constraints state that each data point must lie on the correct side of the margin.

This can be rewritten as : $y_i(\mathbf{w}^T\mathbf{x}_i - b) \geq 1, \quad$ for all $1 \leq i \leq n. \quad (1)$ We can put this together to get the optimization problem: : "Minimize $\|\mathbf{w}\|$ subject to $y_i(\mathbf{w}^T\mathbf{x}_i - b) \geq 1$ for $i = 1, \ldots, n$."

The $\mathbf{w}$ and b that solve this problem determine our classifier, $\mathbf{x} \mapsto (\mathbf{w}^T\mathbf{x} - b)$ where $(\cdot)$ is the sign function.

An important consequence of this geometric description is that the max-margin hyperplane is completely determined by those $\vec{x}_i$ that lie nearest to it. These $\mathbf{x}_i$ are called support vectors.

2) Soft-margin: To extend SVM to cases in which the data are not linearly separable, the hinge loss function is helpful

$: \max\left(0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i - b)\right).$

Note that y_i is the i-th target (i.e., in this case, 1 or 1), and $\mathbf{w}^T\mathbf{x}_i - b$ is the i-th output.

This function is zero if the constraint in (1) is satisfied, in other words, if $\mathbf{x}_i$ lies on the correct side of the margin. For data on the wrong side of the margin, the function's value is proportional to the distance from the margin.

The goal of the optimization then is to minimize

$: \lambda\|\mathbf{w}\|^2 + \left[\frac{1}{n}\sum_{i=1}^{n} \max\left(0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i - b)\right)\right],$

where the parameter $\lambda > 0$ determines the trade-off between increasing the margin size and ensuring that the $\mathbf{x}_i$

lie on the correct side of the margin. Thus, for sufficiently small values of λ, it will behave similar to the hard-margin SVM, if the input data are linearly classifiable, but will still learn if a classification rule is viable or not.

B. Non-Linear SVM

The algorithm of a Nonlinear classifier is formally similar, except that every dot product is replaced by a nonlinear kernel function. This allows the algorithm to fit the maximum-margin hyperplane in a transformed feature space. The transformation may be nonlinear and the transformed space high-dimensional; although the classifier is a hyperplane in the transformed feature space, it may be nonlinear in the original input space.

III. Using Support Vector Machines for Classification (to Identify Pulsar Stars)

A. Description of the Dataset

```
RangeIndex: 12528 entries, 0 to 12527
Data columns (total 9 columns):
 #   Column                                        Non-Null Count  Dtype
---  ------                                        --------------  -----
 0    Mean of the integrated profile               12528 non-null  float64
 1    Standard deviation of the integrated profile 12528 non-null  float64
 2    Excess kurtosis of the integrated profile    10793 non-null  float64
 3    Skewness of the integrated profile           12528 non-null  float64
 4    Mean of the DM-SNR curve                     12528 non-null  float64
 5    Standard deviation of the DM-SNR curve       11350 non-null  float64
 6    Excess kurtosis of the DM-SNR curve          12528 non-null  float64
 7    Skewness of the DM-SNR curve                 11903 non-null  float64
 8   target_class                                  12528 non-null  float64
dtypes: float64(9)
memory usage: 881.0 KB
```

Fig. 2. Information about the dataset

The dataset comprises of 12528 observations and 9 variables. Out of which one variable('target_class') is the dependent variable and the rest 8 are independent variables.

	Mean of the integrated profile	Standard deviation of the integrated profile	Excess kurtosis of the integrated profile	Skewness of the integrated profile	Mean of the DM-SNR curve	Standard deviation of the DM-SNR curve	Excess kurtosis of the DM-SNR curve	Skewness of the DM-SNR curve	target_class
count	12528.000000	12528.000000	10793.000000	12528.000000	12528.000000	11350.000000	12528.000000	11903.000000	12528.000000
mean	111.041841	46.521437	0.478548	1.779431	12.674758	26.351318	8.333489	105.525779	0.092034
std	25.672828	6.801077	1.064708	6.228450	29.613230	19.615842	4.535783	107.399585	0.289085
min	5.812500	24.772042	-1.738021	-1.791886	0.213211	7.370432	-3.139270	-1.976976	0.000000
25%	100.871094	42.362222	0.024652	-0.188142	1.910535	14.404353	5.803063	35.199899	0.000000
50%	115.183594	46.931022	0.223678	0.203317	2.792642	18.412402	8.451097	83.126301	0.000000
75%	127.109375	50.979103	0.473125	0.932374	5.413253	28.337418	10.727927	139.997850	0.000000
max	189.734375	91.808628	8.069522	68.101622	222.421405	110.642211	34.539844	1191.000837	1.000000

Fig. 3. Statistics of the dataset

It was noted that all the input variables are found to be of float type and are continuous. The variable 'target_class' alone is found to be a binary categorical variable (Fig.2. and Fig.3.).

B. Data Cleaning

The raw dataset was found to be having a significant amount of NaN/missing values. Hence, the rows having such columns were dropped from our dataframe.

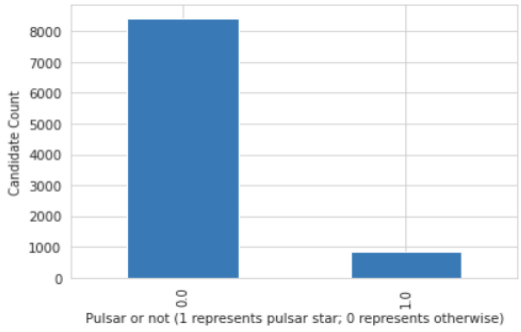

Fig. 4. Candidate Count of the two classes

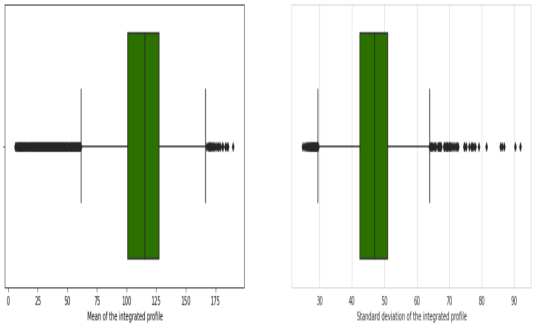

Fig. 5. Box plot to showing the distribution of quantitative data

C. Visualizations and insights from the analysis

From (Fig.4.) we infer that, A majority of candidates are not pulsar stars. The Data is imbalanced.

The box plot(Fig.5.) shows that the data contains a significant amount of outliers/ variables with a huge range of values.

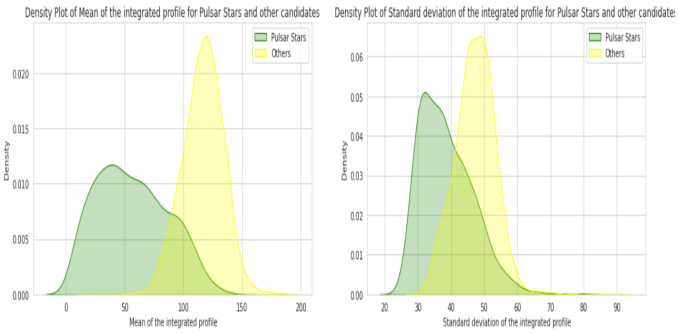

Fig. 6. Density Distribution of 'Mean of the integrated profile' and 'Standard deviation of the integrated profile' for Pulsar starts and other candidates

From (Fig.6.) we can conclude that, Pulsar Stars have a lower 'Mean of the integrated profile' when compared to other candidates. And that, Pulsar Stars have a lower 'Standard deviation of the integrated profile' when compared to other non-pulsar candidates.

From(Fig.7.) we can conclude that, Pulsar Stars exhibit a very wide range of values for the variables: 'Excess kurtosis of the integrated profile', ' Skewness of the integrated profile' and ' Mean of the DM-SNR curve'.

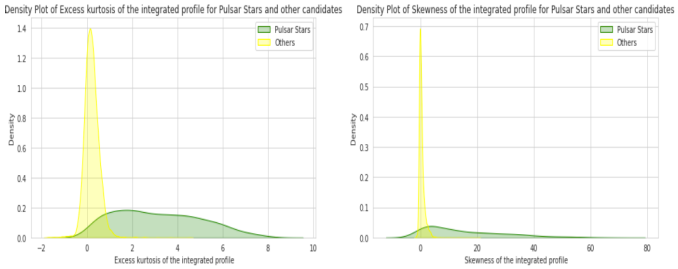

Fig. 7. Density Distribution of 'Excess kurtosis of the integrated profile', ' Skewness of the integrated profile' and ' Mean of the DM-SNR curve' for Pulsar starts and other candidates

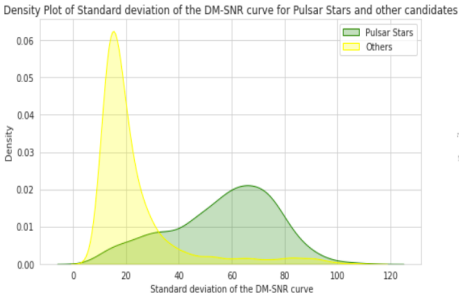

Fig. 8. Density Distribution of 'Standard deviation of the DM-SNR curve' for Pulsar starts and other candidates

From(Fig.8.) we can infer that Pulsar Stars have a higher 'Standard deviation of the DM-SNR curve' when compared to other non-pulsar candidates.

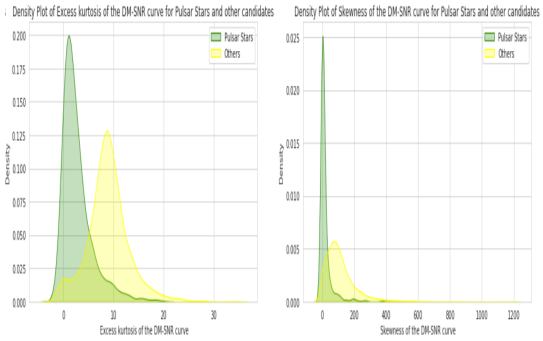

Fig. 9. Density Distribution of 'Excess kurtosis of the DM-SNR curve' and 'Skewness of the DM-SNR curve' for Pulsar starts and other candidates

From (Fig.9.), it can be said that, Pulsar Stars have a lower 'Excess kurtosis of the DM-SNR curve' and 'Skewness of the DM-SNR curve' when compared to other non-pulsar candidates.

D. Data Preprocessing

The dataset was also split into the training and validation parts with the validation data being used to calculate the accuracy and performance of the model before making predictions on the test data.

E. Building the Support Vector Machine Classifier Model

A Support Vector Machine Classifier Model has been built using the inbuilt 'SVC' method from the Skikit-Learn library. The variables- 'Mean of the integrated profile', 'Standard deviation of the integrated profile', 'Excess kurtosis of the integrated profile', 'Skewness of the integrated profile', ' Mean of the DM-SNR curve', 'Standard deviation of the DM-SNR curve', 'Excess kurtosis of the DM-SNR curve', 'Skewness of the DM-SNR curve' were taken as the continuous input variables and were used to predict the binary categorical output variable-'target_class'.

F. Model performance

The dataset was split into the training and validation parts and the training dataset was used to train the model to fit the given data. The model was found to fit the training data with an accuracy of 97.93%.

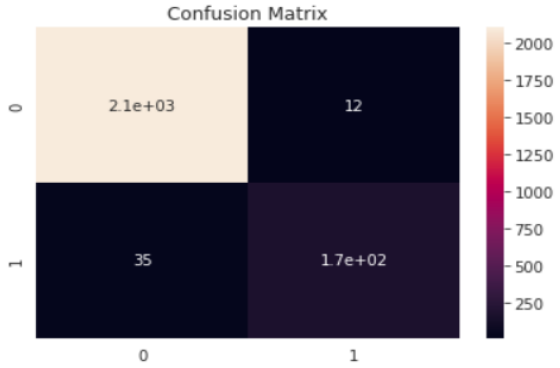

Fig. 10. Confusion matrix of our model on the validation dataset

The performance of the model was also evaluated on the validation dataset and, the accuracy of the model was found to be 97.97%. A confusion matrix of our model(Fig.10.) on the validation dataset was plotted to visualize the miss-classified data-points.

IV. CONCLUSIONS

We analyzed the role of various features- 'Mean of the integrated profile', 'Standard deviation of the integrated profile', 'Excess kurtosis of the integrated profile', 'Skewness of the integrated profile', ' Mean of the DM-SNR curve', 'Standard deviation of the DM-SNR curve', 'Excess kurtosis of the DM-SNR curve', 'Skewness of the DM-SNR curve' in estimating the 'target_class' of candidates. It was concluded that, Pulsar Stars generally have a lower 'Mean of the integrated profile' when compared to other candidates. And that, Pulsar Stars have a lower 'Standard deviation of the integrated profile' when compared to other non-pulsar

candidates. We also inferred that, Pulsar Stars exhibit a very wide range of values for the variables: 'Excess kurtosis of the integrated profile', ' Skewness of the integrated profile', ' Mean of the DM-SNR curve'. Pulsar It is also true that, in general, Pulsar Stars have a higher 'Standard deviation of the DM-SNR curve' when compared to other non-pulsar candidates. It can also be said in general that, Pulsar Stars have a lower 'Excess kurtosis of the DM-SNR curve' and 'Skewness of the DM-SNR curve' when compared to other non-pulsar candidates.

Despite the high accuracy of the model on the validation dataset(97.97%), further improvements in the model are possible. In order to further improve the overall result, an extensive hyper-parameter tuning can be done to improve the accuracy of the model for it to better fit the data. It also is possible to further improve it by possibly finding even better machine learning algorithms for the concerned task.

REFERENCES

[1] Evgeniou, Theodoros Pontil, Massimiliano. (2001). Support Vector Machines: Theory and Applications. 2049. 249-257. 10.1007/3-540-44673-7_12.

[2] Fearn, Tom. (2004). Support Vector Machines I: The Support Vector Classifier. NIR news. 15. 14. 10.1255/nirn.788.

[3] Fearn, Tom. (2004). Support Vector Machines I: The Support Vector Classifier. NIR news. 15. 14. 10.1255/nirn.788.

[4] Palancz, Bela Volgyesi, Lajos. (2004). Support vector classifier via Mathematica. 48.