DETECTION OF PULSAR STARS USING MACHINE LEARNING ALGORITHMS

P.Mounika[1], P. Siva Spandana[2], M.Anuja[3], N. Sai Krishna[4], S. Jaya Prada[5]

[1,2,3,4]UG Scholar, Dept. of Computer Science Engineering, Gudlavalleru Engineering College, Andhra Pradesh, India
[5]Associate Professor, Dept. of Computer Science Engineering, Gudlavalleru Engineering College, Andhra Pradesh, India

---***---

Abstract - *A pulsar is an exceptionally polarized pivoting neutron star or white smaller person that emits a light emission radiation. This radiation can be watched just when the light emission is highlighting Earth and is answerable for the beat appearance of emanation. To predict these pulsar stars we are implementing the model using some machine learning algorithms like decision tree, random forest, etc. Pulsars are one of the possibilities for the wellspring of ultra-high-vitality enormous beams. We have tried various models, including decision tree classifier, random forest classifier, KNN (k- nearest neighbors), and support vector machine to predict pulsar stars, to get high accuracy, precision, and recall from all those models.*

***Key Words*: Machine Learning, Pulsar stars, Neutron stars, Decision Tree, Random Forest, Heatmap.**

1. INTRODUCTION

A pulsar is a quickly pivoting neutron star. A neutron star is one of the endpoints of the life of a massive star after it explodes in a supernova explosion. A neutron star is one of the end purposes of the life of a gigantic star after it detonates in a supernova blast. This magnetic field is not aligned with the rotation axis of the neutron star. To watch these beats of radiation at whatever point the attractive post is obvious. The beats come at a similar rate as the revolution of the neutron star, and, along these lines, seem occasional. In spite of, before long, radio repeat impedance and fuss can create signals that resemble that of pulsars, so it's particularly intriguing to think of an approach to characterize among pulsars and radio recurrence obstruction or noise. In this paper, we are implementing four models to consequently distinguish the presence of pulsar stars. The list of capabilities contains mean, standard deviation, overabundance kurtosis, and skewness of the incorporated profile and mean, standard deviation, abundance kurtosis, and skewness of the DM-SNR bend. In this paper, we are utilizing four machine learning models to automatically detect the existence of pulsar stars. The feature set contains mean, standard deviation, excess kurtosis, and skewness of the integrated profile and mean, standard deviation, excess kurtosis, and skewness of the DM-SNR curve.

In our model we are implementing 4 different models, they are decision tree classifier, random forest classifier, KNN, and support vector machine to output a predicted label, which is either 1 or 0 (Here 1 represents the pulsar star and 0 represents non-pulsar star).

2. RELATED WORK

One of the challenges experienced during the scans for pulsars in the previous fifty years is the investigation required for the expanding number of 'competitor' pulsar locations emerging from an expanding volume of information to be looked at. Our investigation centers around understanding this trouble utilizing different AI strategies to expand the exactness, accuracy, and review for both positive and negative marks on both the first dataset and the produced counterfeit dataset. There are a couple of prior works that have investigated this issue. In this fragment, we talk about the prior work, investigating where appropriate.

In Fifty years of pulsar up-and-comer choice: from basic channels to another principled continuous characterization approach utilize factual grouping to separate 8 new highlights to limit the number of highlights without reducing arrangement execution. Every up-and-comer is a potential recognition of a pulsar signal, which displays explicit qualities of intrigue. Precise choices must be made. Mix-ups can prompt the expensive misuse of telescope time on fake signs, and more regrettable, missing legitimate signs altogether.

Our study expands on arranging dependent on these 8 highlights separates, while utilizing increasingly refined AI models to accomplish higher exactness, accuracy, review, explicitness, and furthermore negative exactness with the choice tree.

There have been a few endeavors to illuminate this pulsar star applicant choice issue. Eatough et al what's more, Bates et al. utilize Artificial Neural Networks (ANNs) to approach the problems[1][2]. Their models are powerful somewhat as it were. Past models are unable to balance the large datasets. So here we can apply classification algorithms. Classification is the process of discovering a model that helps in separating the data into multiple categorical classes i.e. discrete values.

3. METHODOLOGY

In our model, we are implementing supervised learning algorithms. The idea of these Algorithms is pretty simple. You anticipate the objective class by investigating the preparation dataset. Our concern calls for conveying grouping calculations. Briefly, the Classification is utilized to anticipate the unmitigated names, or it groups the information dependent on the prepared set and qualities is utilized in ordering properties and utilized in arranging the new information. There are numerous order models. The Classification models are calculated relapse, choice tree, arbitrary woodland, KNN, bolster vector machine.

Decision Tree:

A Decision Tree is a straightforward portrayal of ordering models. It is a Supervised Machine Learning where the information is consistently part as indicated by a specific parameter.

The fundamental goal of utilizing Decision Tree in this examination work is the expectation of target class utilizing the choice principle taken from earlier information. It uses nodes and internodes for prediction and classification. Root nodes classify instances with different features. Root nodes can have two or more branches while the leaf nodes represent classification. In each stage, the Decision tree picks every hub by assessing the most elevated data gain among all the qualities.

There are two primary sorts of Decision Trees. Classification Trees and Regression Trees. Classification trees belong to Truly/NO sorts whereas relapse trees have a place with continuous information types.

The Decision tree is a flowchart which seems like a tree structure where an Internal hub addresses feature, and the branch addresses a decision rule, and each leaf center point addresses the outcome. The most elevated center in the decision tree is known as the root center. It makes sense of how to fragment dependent on the property estimation. It parcels the tree in recursively way call recursive apportioning.

The primary goal of utilizing Decision Tree in this examination work is the forecast of the target class utilizing the choice principle taken from earlier information. It utilizes hubs and buries hubs for the expectation and order. Root hubs arrange examples with various highlights. Root hubs can have at least two branches while the leaf hubs speak to order. In each stage, the Decision tree picks every hub by assessing the most noteworthy data gain among all the qualities.

Random Forest:

The Random Forest is a model comprised of numerous choice trees. Instead of averaging the forecast of trees (which we could call a "timberland"), this model uses two key ideas that give it the name random:

a) Random inspecting of preparing information focuses when building trees.
b) Random subsets of highlights thought about while parting nodes.

A random sampling of training data points when building trees:

- When preparing, each tree in an arbitrary backwoods gain from an irregular example of the information focuses.
- The examples are drawn with substitution known as bootstrapping, which implies that a few examples will be utilized on various occasions in a solitary tree.
- At test time, forecasts are made by averaging the expectations of every choice tree.

Random subsets of highlights thought about while parting hubs:

- The other primary idea in the irregular woods is that solitary a subset of the considerable number of highlights is considered for parting every hub in every choice tree.
- Generally, this is set to sqrt(n_features) for arrangement implying that if there are 16 highlights, at every hub in each tree, just 4 irregular highlights will be considered for parting the hub.

Irregular backwoods classifier makes a lot of choice trees from the haphazardly chosen subset of preparing the set. It at that point totals the votes from various choice trees to choose the last class of the test object. It is a troupe calculation. Ensembled calculations are those which join more than one calculations of the same or diverse kind for ordering objects.

Working procedure of the Random Forest algorithm in our model:

There are two phases in Random Forest calculation, one is irregular woodland creation, the other is to make a forecast from the arbitrary timberland classifier made in the principal stage. The entire procedure is demonstrated as follows: Randomly select "K" highlights from all-out "m" highlights where k << m. Among the "K" highlights, figure the hub "d" utilizing the best part point. Split the hub into little girl hubs utilizing the best split. Repeat the a to c ventures until "l" number of hubs has been reached. Build timberland by rehashing stages a to d for "n" number occasions to make "n" number of trees. Takes the test highlights and utilizes the principles of each haphazardly made choice tree to anticipate the result and stores the anticipated result (target). Calculate the decisions in favor of each anticipated objective. Consider the high casted a ballot anticipated objective as the last expectation from the arbitrary timberland calculation.

KNN (K- Nearest Neighbors) :

The k-closest neighbors (KNN) calculation is a straightforward, simple to-actualize regulated AI calculation that can be utilized to take care of both order and relapse issues. It is anything but difficult to execute and see, however it has a significant disadvantage of turning out to essentially ease back as the size of that information being used grows.KNN is a model that groups information focuses dependent on the focuses that are generally like it. It utilizes test information to make an "informed conjecture" on what an unclassified point ought to be delegated. In any case, it is for the most part utilized for characterization prescient issues in the industry. The accompanying two properties would characterize KNN well –

- Lazy learning calculation – KNN is a lethargic learning calculation since it doesn't have a particular preparing stage and uses all the information for preparing while classification.

- Non-parametric learning calculation – KNN is additionally a non-parametric learning calculation since it doesn't accept anything about the basic information.

Working procedure of KNN algorithm in our model :

Stage 1 – For executing any calculation, we need a dataset. So during the initial step of KNN, we should stack the preparation just as test information.

Stage 2 – Next, we have to pick the estimation of K for example the closest information focuses. K can be any whole number.

Stage 3 – For each point in the test information do the accompanying –

- Calculate the separation between test information and each line of preparing information with the assistance of any of the strategies in particular: Euclidean, Manhattan, or Hamming separation. The most usually utilized technique to compute separation is Euclidean.
- Now, in light of the separation esteem, sort them in rising request.
- Next, it will pick the top K lines from the arranged cluster.
- Now, it will allow a class to the test point dependent on the most continuous class of these lines.

Support Vector Machine:

SVM can be officially characterized as, "A Support Vector Machine (SVM) is a discriminative classifier officially characterized by an isolating hyperplane. At the end of the day, given named preparing information (managed learning), the calculation yields an ideal hyperplane which sorts new models. In two dimensional space, this hyperplane is a line isolating a plane in two sections were in each class lay in either side".

The learning of the hyperplane in direct SVM is finished by changing the issue utilizing some straight variable based math. This is the place the bit assumes the job. We have utilized the 'RBF' part which is a well-known piece of work utilized in different kernelized learning algorithms. Specifically, it is regularly utilized to help vector machine characterization.

4. IMPLEMENTATION

The system architecture is as follows:

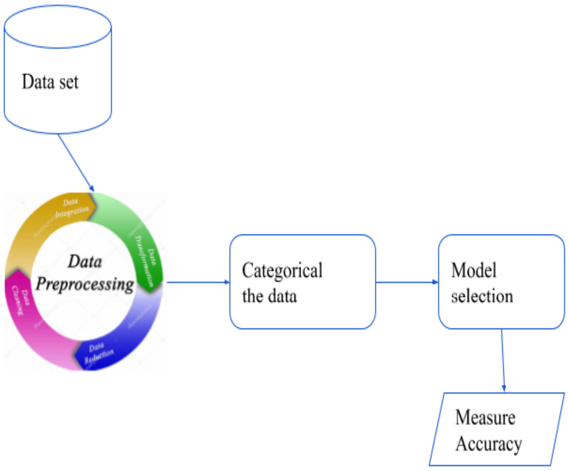


Fig 1: Architecture of our proposed method

For any model, we ought to give the dataset as the information and we ought to preprocess it and afterward, we will choose the model and measure the exactness for our model. A calculation that gives high precision will be considered as the best calculation for our model.

Data Preprocessing:

In the data preprocessing we will preprocess the dataset and handle the null values and missing values. Information preprocessing is an information mining method that includes changing crude information into a reasonable arrangement. Veritable data is now and again divided, clashing, and furthermore feeble in explicit practices or floats, and is likely going to contain various mix-ups. Information preprocessing is a demonstrated technique for settling such issues.

When data is incomplete or inconsistent or noisy then data preprocessing is needed. There are so many routes to deal with unprocessed data:

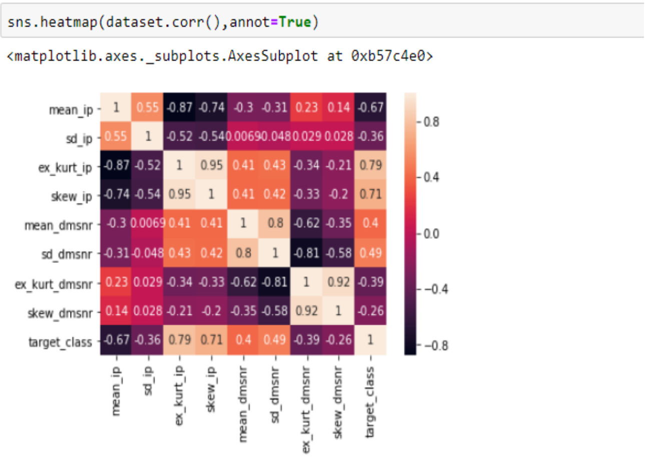
i) *Data Cleaning:* To overfill the missing values in data by using mean or median or mode or any formulae.
ii) *Data Reduction*: In this step, the dataset is altered so the results processed by any model will be identical but not needed values in the dataset are removed.
iii) *Data Integration*: In this step, data is combined from various origins if required and redundancies are removed too.
iv) *Data Transformation*: In this step, operations like normalization is performed.

5. RESULTS AND DISCUSSION

DATASET:

HTRU2 is a dataset that portrays an example of pulsar up-and-comers gathered during the High Time Resolution Universe Survey which depicts pulsar competitors gathered during the High Time Resolution Universe Survey.

Each applicant in the dataset contains 8 qualities: Mean of the coordinated profile, Standard deviation of the incorporated profile, Excess kurtosis of the coordinated profile, Skewness of the coordinated profile, Mean of the DM-SNR bend, Standard deviation of the DM-SNR bend, Excess kurtosis of the DM-SNR bend, Skewness of the DM-SNR bend.



```
sns.heatmap(dataset.corr(),annot=True)
```

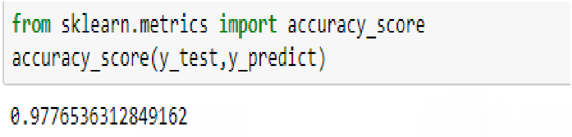
```
<matplotlib.axes._subplots.AxesSubplot at 0xb57c4e0>
```

	mean_ip	sd_ip	ex_kurt_ip	skew_ip	mean_dmsnr	sd_dmsnr	ex_kurt_dmsnr	skew_dmsnr	target_class
mean_ip	1	0.55	-0.87	-0.74	-0.3	-0.31	0.23	0.14	-0.67
sd_ip	0.55	1	-0.52	-0.54	0.0069	0.048	0.029	0.028	-0.36
ex_kurt_ip	-0.87	-0.52	1	0.95	0.41	0.43	-0.34	-0.21	0.79
skew_ip	-0.74	-0.54	0.95	1	0.41	0.42	-0.33	-0.2	0.71
mean_dmsnr	-0.3	0.0069	0.41	0.41	1	0.8	-0.62	-0.35	0.4
sd_dmsnr	-0.31	-0.048	0.43	0.42	0.8	1	-0.81	-0.58	0.49
ex_kurt_dmsnr	0.23	0.029	-0.34	-0.33	-0.62	-0.81	1	0.92	-0.39
skew_dmsnr	0.14	0.028	-0.21	-0.2	-0.35	-0.58	0.92	1	-0.26
target_class	-0.67	-0.36	0.79	0.71	0.4	0.49	-0.39	-0.26	1

Fig 2: Heatmap for our model

Decision Tree Classifier :

By using the decision tree classifier the accuracy score is as shown below :

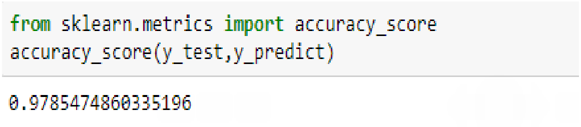


```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_predict)
```

```
0.9776536312849162
```

Random Forest Classifier :

By using the random forest classifier the accuracy score is as shown below :

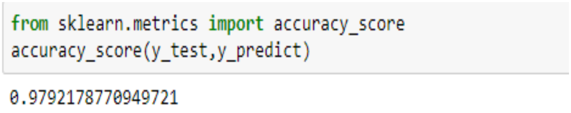


```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_predict)
```

```
0.9785474860335196
```

K- nearest neighbors :

By using the K-nearest neighbors the accuracy score is as shown below :

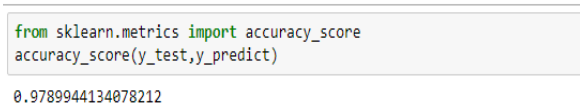


```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_predict)
```

```
0.9792178770949721
```

Support Vector Machine

By using the support vector machine the accuracy score is as shown below :



```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_predict)

0.9789944134078212
```

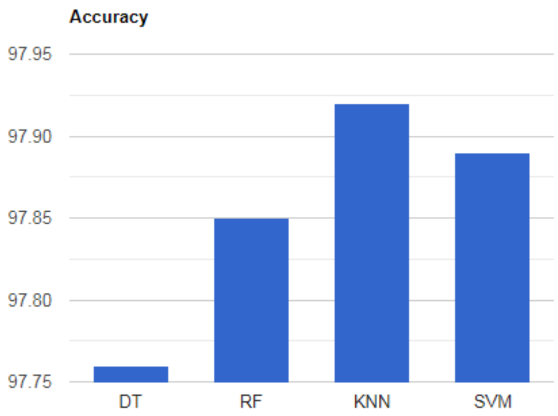


Fig 3: Accuracy Comparision for different algorithms

An Overview of our Proposed algorithm:

In our model after using different algorithms and after comparing different algorithms accuracy results we are proposing K- nearest neighbor algorithm as our proposed algorithm. Here KNN plays a precious role in the implementation of our application. Because it is the one which is providing accurate results with a high accuracy score. K-nearest neighbors (KNN) calculation utilizes 'include likeness' to anticipate the estimations of new points which further implies that the new information point will be allotted a worth dependent on how intently it coordinates the focuses in the preparation set. KNN has some advantages. They are: It is a straightforward calculation to comprehend and interpret. It is valuable for nonlinear information in light of the fact that there is no suspicion about information in this algorithm. It is a flexible calculation as we can utilize it for an order just as relapse.

6. CONCLUSION

In this paper, we have executed 4 Machine learning models: Decision Tree Classifier, Random Forest Classifier, K-Nearest Neighbors, and Support Vector Machine to discover the pulsar star competitors. They utilize a measurable technique to produce a reasonable dataset to test the 4 models. Among these models, KNN plays out the best on the all-out precision while different models are just marginally more terrible than it. Our models will be helpful, and particularly k nearest neighbor, for future pulsar star distinguishing proof in the region of cosmology. Utilize our model to find genuine Pulsar Stars known to mankind as past investigations did. On the off chance that pulsar stars can be discovered dependent on our forecasts, our models end up being helpful in an application.

7. FUTURE SCOPE

In the future, by applying different methods and techniques it can improve the accuracy score and the results can become more accurate. Test our models in more pulsar star competitor datasets other than HTRU2 to know the results accurately. Generate a superior dataset, incorporate more highlights that are in the first HTRU2 datasets, and key portrayals of Pulsars and is bigger and adjusted. Utilize better adjusting techniques to create better disseminated counterfeit information. In the event that pulsar stars can be discovered dependent on our forecasts, our AI models end up being valuable in the application.

ACKNOWLEDGMENT

We thank the coordinator of Smart bridge academy Mahankali Surya Tej for useful discussions on machine learning techniques and Somala Jaya Prada (Associate Professor) for the support.

REFERENCES

[1] R. P. Eatough, N. Molkenthin, M. Kramer, A. Noutsos, M. J. Keith, B. W. Stappers, and A. G. Lyne, “Selection of radio pulsar candidates using artificial neural networks,” Monthly Notices of the Royal Astronomical Society, vol. 407, pp. 2443–2450, 07 2010.

[2] S.Bates, M.Bailes, B.Barsdell, N.Bhat, M.Burgay, S. Burke-Spolaor, D. Champion, P. Coster, N. D Amico, A. Jameson, et al., "the high time resolution universe pulsar survey A Tvi. an artificial neural network and timing of 75 pulsars ", Monthly Notices of the Royal Astronomical Society, vol. 427, no. 2, pp. 1052-1065, 2012.

[3] R. J. Lyon, Why are pulsars hard to find? Ph.D. thesis, The University of Manchester (United Kingdom), 2016.

[4] Schoelkopf B, Burges C J C, Smola A J, Advances in Kernel Methods - Support Vector Learning. Cambridge: MIT Press, 1999.

[5] Machine Learning Approach to Detect Pulsar Star: TensorFlow and Random Forest Model with Python.

[6] R. J. Lyon, B. Stappers, S. Cooper, J. Brooke, and J. Knowles, “Fifty years of pulsar candidate selection: from simple filters to a new principled real-time classification approach,” Monthly Notices of the Royal

Astronomical Society, vol. 459, no. 1, pp. 1104–1123, 2016.

[7] D. R. Lorimer, M. Kramer, et al., Handbook of pulsar astronomy, vol. 4. Cambridge university press, 2005.

[8] R.N.Manchester and J.Taylor, "pulsars", Freeman, SranFcisco, 1977.

[9] Ransom S., 2011, PRESTO: Pulsar Exploration and Search Toolkit. Astrophysics source code library.