Classification of exoplanets

UE17CS303 - MACHINE LEARNING ASSIGNMENT

1 Team Members

1. ARPIT AGARWAL (PES1201701084)
2. ISHAAN LAGWANKAR (PES1201700150)
3. MALAVIKKA RAJMOHAN (PES1201700794)
4. RICHA SHARMA (PES1201700662)

2 Problem Statement

Explore the efficacy of machine learning (ML) in characterizing exoplanets into different classes. The source of the data used in this work is University of Puerto Rico's Planetary Habitability Laboratory's Exoplanets Catalog (PHL-EC). Perform a detailed analysis of the structure of the data and propose methods that can be used to ffectively categorize new exoplanet samples. Contributions are two-fold; elaborate on the results obtained by using ML algorithms by stating the accuracy of each method used and propose a paradigm to automate the task of exoplanet classification for relevant outcomes.

In the process of exploring what were the distinct models that could be used to solve this issue, the pipeline was divided broadly into four categories

1. Pre-processing
2. Training
3. Testing
4. Analysis

3 Dataset

The dataset provided to perform analysis was a collection of observed explored exoplanets, with about 69 categorical and numerical attributes.

3.1 Features of the dataset

Almost all the 64 features of the dataset were used to predict the output classes for the final model. These features were first preprocessed and then based on the model, were changed according to the requirement. For example, for the Gaussian and Bernoulli Naive Bayes models, the features were first normalised and fit onto a Gaussian and Bernoulli curve to give accurate results.

3.2 Output Classes

For the analysis, 5 output classes were considered.
1. Atmosphere
2. Zone
3. Habitable
4. Mass
5. Composition

The features in itself had no correlation among each other, so we assumed all features to be independent of each other. The output labels had no correlation amongst themselves either, and hence were treated to be independent classes, and the models tried to figure out a meaningful weighted ensemble of these features to try and predict these classes.

4 Data Cleaning

Before training the models, the dataset was cleaned.
1. All the NaN values were filled with 0
2. The categorical class names were encoded to integers for ease of classification
3. The features which did not contribute to the classification were removed, the Kepler name for instance.

5 Techniques Employed

The models explored mainly dealt with classifications based on 5 broad categories given, namely the Habitable class, Zone class, Atmosphere class, Mass Class and Composition class. Most of the models presented dealt with predicting the habitable classes for planets.

5.1 K-Nearest Neighbours

The approach used in the K-Nearest Neighbours technique was basically to look at a datapoints k nearest datapoints, and predicting a class that assumes a weighted average of all these datapoints combined.

The KNN approach was explored by taking different values of k, ranging from 3 to 16, to check optimal accuracy.
The scores obtained were:

Accuracy metric	
Value of K chosen	Accuracy Score
3	0.9819
4-7	0.9806
8-12	0.9832
13-16	0.9849

The optimal value of k was chosen as 14 as it gave the highest accuracy among it's neighbouring values, and a classification score of 0.9832 was obtained on a test set which was selected by the 80-20 rule, via simple random sampling.

5.2 Decision Trees

This approach is derived from the straightforward implementation of a decision tree, where information gain and entropy of attributes is used to classify the target class. This approach did not give a good result mainly due to the skewness of the data and oversampling performed on the data. So this method was scrapped. The results are given as follows:

Accuracy metric	
Attribute Chosen	Accuracy Score
Mass	0.3170
Zone	0.0835
Composition	0.5270
Atmosphere	0.0.6221

5.3 Naive Bayes

In this, we assumed each pair of attributes to be independent of each other, to create two naive bayesian classifiers, where in we assumed the data to follow a bernoulli distribution in one, and a gaussian distribution in another. For each predictive class, both the models were run to check their efficiency in predicting the class required.

5.3.1 Bernoulli Naive Bayes

The Bernoulli Naive Bayes classifier assumes each attribute to be following a Bernoulli's distribution, and used the Bernoulli's theorem in accordance with the Bayesian rule to provide the right classification.

5.3.2 Gaussian Naive Bayes

In the Gaussian model, the data was assumed to follow a Gaussian model, and the model used the same Bayesian rule to update the classes.

5.3.3 Comparison

A side by side comparison for each class was recorded as

Accuracy metric		
Class used for prediction	Bernoulli Naive Bayes	Gaussian Naive Bayes
Mass	0.7187	0.6645
Zone	0.9683	0.9496
Composition	0.9845	0.6864
Atmosphere	0.6954	0.6761
Habitable	0.9832	0.9651

Both models seem to do well on some classes, but a combination of both of these could be used as an ensemble to predict a new class. On an average, the Bernoulli model was better than the Gaussian Model.

5.4 Support Vector Machines

The support vector machine concept was used to find an optimal hyperplane separating the datapoints into two clusters. Each attribute to predict was taken independently to get individual hyperplanes per attributes. The metrics were observed as:

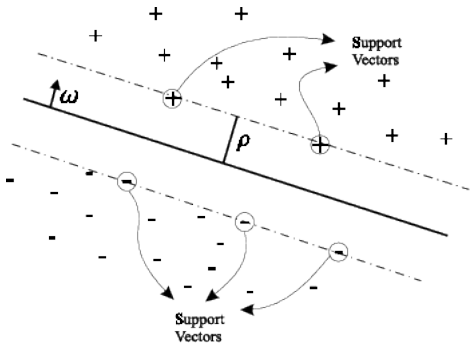

Figure 1: Support Vector Machine Hyperplane

Accuracy metric	
Attribute Chosen	Accuracy Score
Habitable	0.9858
Mass	0.9832
Zone	0.9942
Composition	0.9896
Atmosphere	0.9935

A dummy visualisation of the hyperplane is also shown.

6 Summary of results

Overall, the models were giving good accuracies, but it was observed that the accuracies for some of the models were a bit too high, leading to the possibility of overfitting. So, the accuracy of each model was validated using cross-validation methods, by creating simple random samples of the data.
For the KNN model, the accuracies for different values of K varied as shown in Figure 2.

-

The class wise accuracies for each model were given as shown in Figure 2. The SVM model seems to have outperformed the Naive Bayes and KNN models in terms of these binary classification problems.

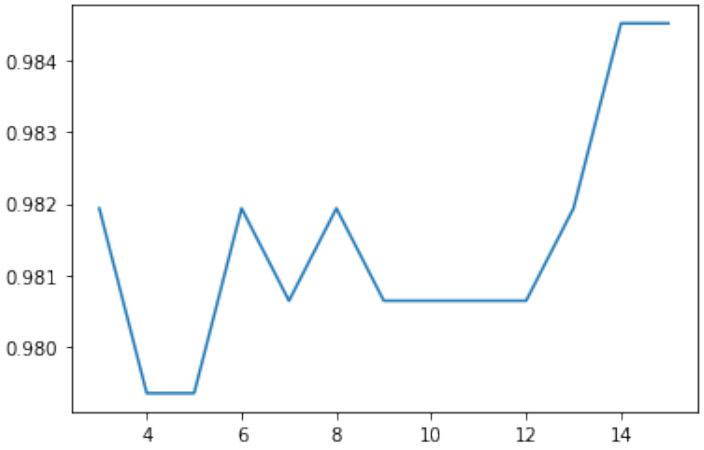

Figure 2: KNN metrics with increasing K

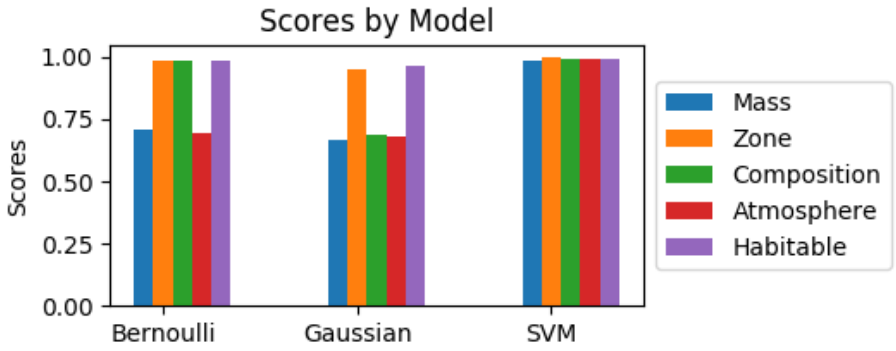

Figure 3: Accuracies for different models

7 Conclusion

In conclusion, the Support Vector Machine model seems to be the most optimal model that can be used in such a classification problem with this dataset, as it gave high accuracies for the predictable classes. There might be a case of overfitting not explored, but since the validation was created through a simple random sample, the model seems to be performing well. The model appears to be skewed towards a habitable class, because of how the dataset was created in the first place, so there were not enough samples to clearly distinguish between the non habitable and habitable class. More examples of the non habitable class may have led to less skewness, but for the existing dataset, the model seems to be working.