Stellar Classification using Pyspark

Team Members:

20MIA1117 - Pillaram Manoj
20MIA1031 - Sanjay.M

ABSTRACT

Stellar classification with PySpark is a machine learning approach that analyses large astronomical data sets and categorises stars based on their physical properties. PySpark is a distributed computing platform that allows for the parallel processing of large data sets, making it a useful tool for analysing astronomical data. Most data is pre-processed to extract important parameters like temperature, brightness, and spectral type before being sent into a machine learning method like a decision tree or neural network. From the training data, the algorithm learns patterns and relationships between features and star classification, and then applies this knowledge to classify new, unseen stars. This classification method may be more accurate and faster.

OBJECTIVE

The objective of this Stellar Classification using PySpark big data project is to use the power of PySpark to classify stars based on their spectral characteristics. The project will involve collecting and cleaning large amounts of astronomical data, performing data analysis and visualizations using PySpark, and using machine learning algorithms to classify stars based on their spectra. The goal of this project is to improve the accuracy and efficiency of stellar classification, and to provide valuable insights into the composition and evolution of stars. This project will demonstrate the ability of PySpark to handle big data and perform complex analysis, and will provide a platform for future research and discovery in the field of astronomy.

INTRODUCTION

Stellar classification is an important task in the field of astronomy, with the goal of categorising stars based on physical properties such as temperature, luminosity, and chemical composition. Understanding the evolution of the universe and the formation of galaxies requires the classification of stars.

Machine learning algorithms have been widely used to automate the process of stellar classification since the advent of big data technologies and the availability of large astronomical datasets.

Many sky survey technology projects, such as the Sloan Digital Sky Survey, have been completed and put into operation as science and technology have advanced. SDSS making data and information about various celestial bodies more accessible. Simultaneously, Big data technology can now use a variety of algorithms to train massive data sets obtained through observation and collection for efficient classification. It is critical not only to study the properties of different stars, but also to further explore the universe. There have been numerous achievements in academic research in this field recently.

We will show how PySpark, an open-source big data processing framework, can be used to perform stellar classification using machine learning algorithms in this report. PySpark is an excellent choice for this task because it provides a powerful and scalable platform for processing large datasets. In this report, we will train and evaluate various classification algorithms on a sample astronomical dataset using PySpark's machine learning library, MLlib. The goal is to find the most accurate and efficient stellar classification algorithm. To use PySpark to analyse stellar data, we must first import the necessary libraries and load the data into a PySpark DataFrame. The data can then be processed using PySpark's built-in functions.

LITERATURE REVIEW

[1]"Stellar Spectra Models Classification and Parameter Estimation Using Machine Learning Algorithms" by Miguel Flores R. and Luis J. Corral presents a study on using machine learning algorithms for the classification and estimation of parameters from observed stellar spectra. The authors evaluate the performance of these algorithms using a publicly available dataset of synthetic spectra. They find that the machine learning algorithms perform well in both classification and parameter estimation tasks, with neural networks achieving the highest accuracy for parameter estimation. They conclude that machine learning can be a valuable tool for the analysis of stellar spectra, offering a fast and effective alternative to traditional techniques.

[2]"Stellar and Pulsar Classification using Machine Learning" by Jishant Talwar, Parul Jain, Chetana Jain, and Baljeet Kaur focuses on the use of machine learning algorithms for the classification of stars and pulsars. The authors use features such as the frequency, amplitude, and phase of pulsar signals to classify them into different categories. For star classification, the authors extract features from the star's light curve, which is a plot of the star's brightness over time. Random Forest

algorithms performed best on both pulsar and star classification tasks, demonstrating the importance of feature selection in achieving high classification accuracy.

[3]"An Application of Deep Learning in the Analysis of Stellar Spectra," discusses the use of deep learning techniques for the analysis of spectra from stars. Spectral analysis is a widely used technique for studying the physical properties of stars, such as their composition, temperature, and other parameters. Deep learning algorithms can be used to perform spectral analysis on astronomical data, outperforming traditional techniques and providing more accurate results. This has the potential to improve our understanding of the physical properties of stars.

[4]"Stellar Spectral Classification Using Principal Component Analysis and Artificial Neural Networks" by Harinder P. Singh, Ravi K , presents a method for classifying stars based on their spectra. The authors propose the use of two techniques, Principal Component Analysis (PCA) and Artificial Neural Networks (ANNs), to classify the stars based on their spectral data.This paper proposes a new method for star classification using PCA and ANNs, which outperforms traditional methods in accuracy and efficiency.

[5]"Automated Stellar Spectra Classification with Ensemble Convolutional Neural Network" by Zhuang Zhao, Jiyu Wei, and Bin Jiang describes a novel method for classifying stars using spectral data. For this purpose, the authors propose using an ensemble of Convolutional Neural Networks (CNNs). They then use multiple CNNs to extract features from the spectra and use an ensemble technique to combine their predictions to improve overall classification accuracy.The authors assess the proposed method's performance on a dataset of stellar spectra and compare the results to those of other commonly used methods in the field. In terms of accuracy and robustness, they demonstrate that the proposed method outperforms traditional methods.

[6] "Stellar Spectral Classification Based on Capsule Network" by DU Li-ting a, HONG Li-hu describes a novel method for classifying stars using spectral data. For this purpose, the authors propose using a Capsule Network, a type of deep neural network. The authors employ the Capsule Network in this method to extract features from spectral data and classify stars based on these features. They also propose a new loss function designed specifically for the Capsule Network to improve its classification performance. The authors assess the proposed method's performance on a dataset of stellar spectra and compare the results to those of other commonly used methods in the field. They demonstrate that in terms of accuracy, the proposed method outperforms traditional methods.

[7]"Star-Galaxy Classification Using Deep Convolutional Neural Networks" by Edward J. Kim describes a new method for classifying celestial objects based on image data as stars or galaxies. Deep Convolutional Neural Networks (DCNNs) are proposed by the authors for this purpose.The authors use a DCNN to extract features from image data and classify the objects as stars or galaxies based on these features in this method. The authors also propose a new data augmentation technique to improve the DCNN's classification performance. This paper presents a novel approach to star-galaxy classification based on Deep Convolutional Neural Networks and demonstrates its efficacy through experiments on a dataset of celestial objects.

[8]"Application of Convolutional Neural Networks for Stellar Spectral Classification" by Kaushal Sharma , Ajit Kembhav and their collegues (2018) describes a new method for classifying stars using spectral data. Convolutional Neural Networks (CNNs) are proposed by the authors for this purpose. They used a CNN to extract features from the spectral data and classify the stars based on these features in this method. They also propose a new data augmentation technique to improve the CNN's classification performance. This paper presents a new approach for stellar spectral classification using Convolutional Neural Networks and demonstrates its effectiveness using stellar spectra experiments.

[9]"Spectral Classification using Convolutional Neural Networks" is a Master's thesis by Bc. Pavel Hala, submitted in Brno in 2014. They have used convolutional neural networks (CNNs) to classify astronomical spectra in this thesis. The goal was to investigate the use of deep learning algorithms for astronomical data analysis and to see if CNNs can improve spectral classification accuracy over traditional methods. The results of the thesis demonstrated that the CNN could accurately classify the spectra into different spectral types with high precision and recall. The author concluded that CNNs are a promising approach for spectral classification, and that further research in this area could lead to improvements in astronomical spectral analysis.Overall, this Master's thesis demonstrates the potential of deep learning algorithms for astronomical data analysis and emphasises the importance of exploring new approaches in this field.

[10]"Spectral Classification of Stars with a Convolutional Neural Network" by Hwang et al. (2019) .This paper presents a method for classifying stars using a convolutional neural network (CNN). The authors use a large dataset of stellar spectra and apply their CNN-based method to classify stars into different spectral types. They also compare their results to those of Kelly et al. (2018) and Wang et al. (2019) and show that their CNN-based approach performs similarly to the other deep learning-based methods.

[11]"An application of deep learning in the analysis of stellar spectra" by Fabbro, S .This paper tells that StarNet is a deep neural network architecture used to analyze both SDSS-III APOGEE DR13 and synthetic stellar spectra, with similar precision and accuracy as the APOGEE pipeline. It can also predict stellar parameters over a wide range of signal-to-noise ratios. The influence of spectral features on the stellar parameters is examined via partial derivatives of the StarNet model results with respect to the input spectra.

PROPOSED SYSTEM

- We will be conducting the following steps
    - Import Required Packages
    - Load the dataset
    - Perform the exploratory data analysis (EDA)
    - Prepare the dataset for training
    - Create Logistic Regression, Random Forest Classifier, K-Means Classifier and Decision Tree Classifier using Pyspark.
    - Train the model to fit the data
    - Make classifications using the trained model.
    - Develop a ANN model

IMPLEMENTATION

Data preprocessing: Preprocessing the data is the first step in any data analysis project. This entails cleaning the data, removing any missing or invalid values, and transforming the data into an analysis-ready format.

Model training: After preprocessing the data and extracting the features, the next step is to train a classification model. PySpark includes a number of machine learning algorithms for classification such as Logistic Regression, Random Forest Classifier, K-Means Classifier and Decision Tree and deep learning algorithm like ANN.

Data Visualization: Finally, it is critical to visualise the classification results. This can be accomplished with tools such as matplotlib or seaborn. PySpark includes functions for creating plots and visualisations that can be used to gain insights into the data and the model's performance.

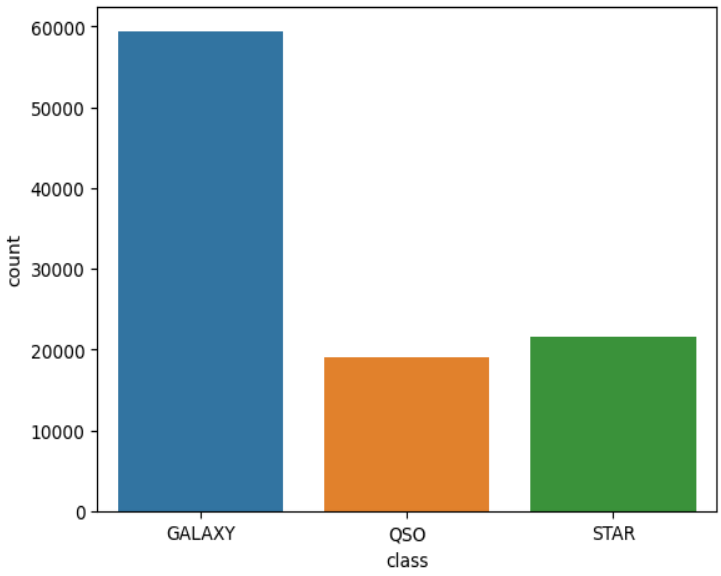
Model selection: Pick a machine learning algorithm that is suitable for your classification task. The deep learning classification methods offered by PySpark include as Logistic Regression, Random Forest Classifier, K-Means Classifier and Decision Tree and deep learning algorithm like ANN.

Model Training: Divide the dataset into training and validation sets. Utilize PySpark's MLlib package to train the chosen model using the training set. To improve model performance, experiment with various hyperparameter values.

Evaluation of the Model: Use relevant evaluation metrics, such as accuracy, precision, recall, F1-score, and confusion matrix, to assess the performance of the trained model. Examine the model's adaptability to fresh data using the validation set.

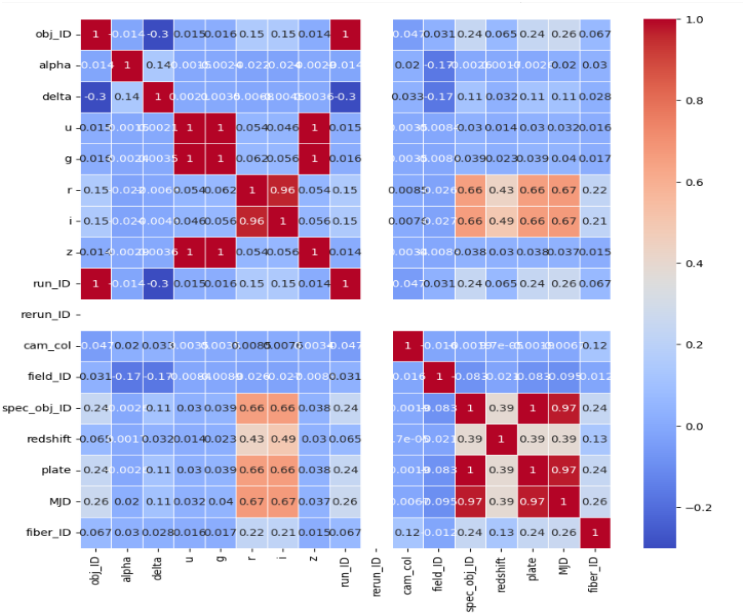
OUTPUTS

Distribution of classes(GALAXY,QSO(quasi stellar object),STAR):



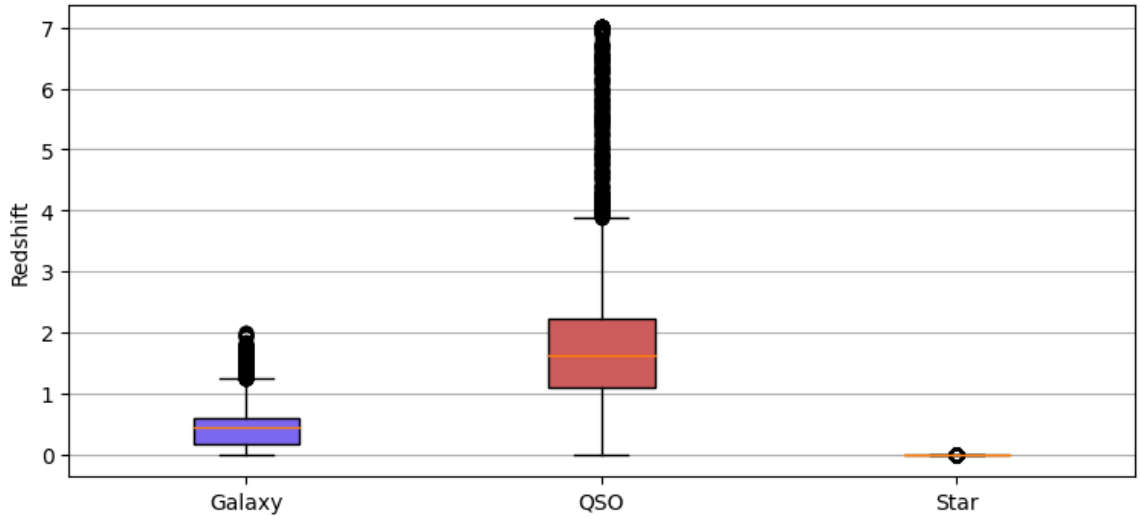
Correlation heat map:

A correlation heatmap is a graphical representation of a correlation matrix that depicts the relationship between several variables. Correlation can have any value between -1 and 1. Correlation between two random variables or bivariate data does not always reflect cause.



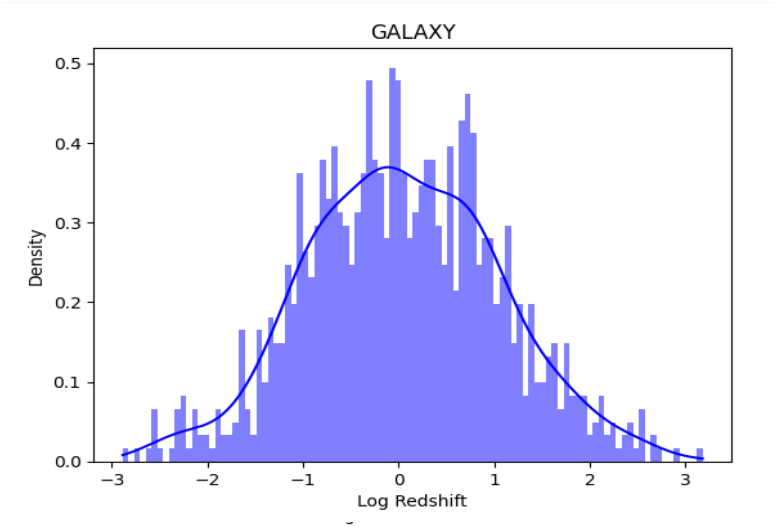
Box plot:

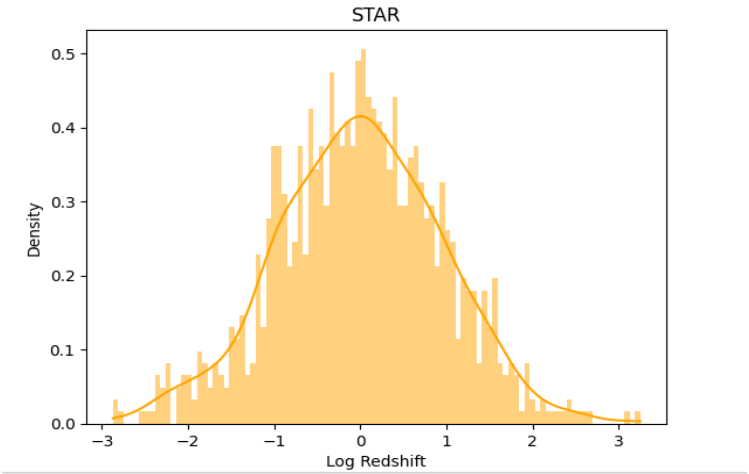
Box plot for the classes Galaxy, QSO, Star and the Y-axis labeled as Redshift.

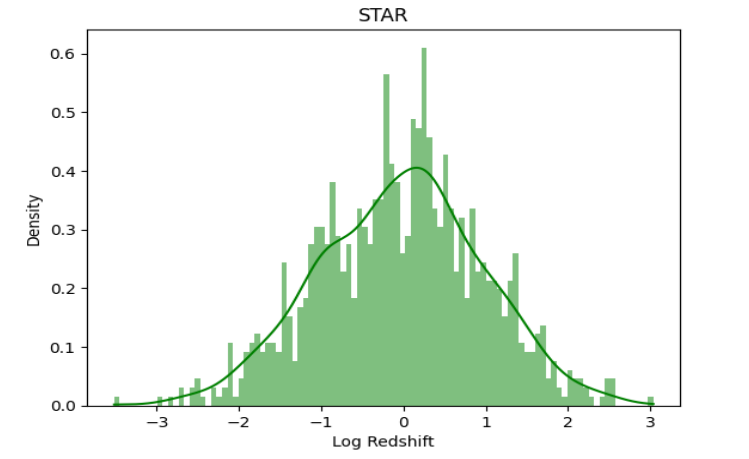


KDE plot:

A kernel density estimate (KDE) plot, similar to a histogram, is a method for visualising the distribution of observations in a dataset. KDE depicts data in one or more dimensions using a continuous probability density curve. They have named the x-label as Log Redshift and y-label as Density for the classes Galaxy, star.



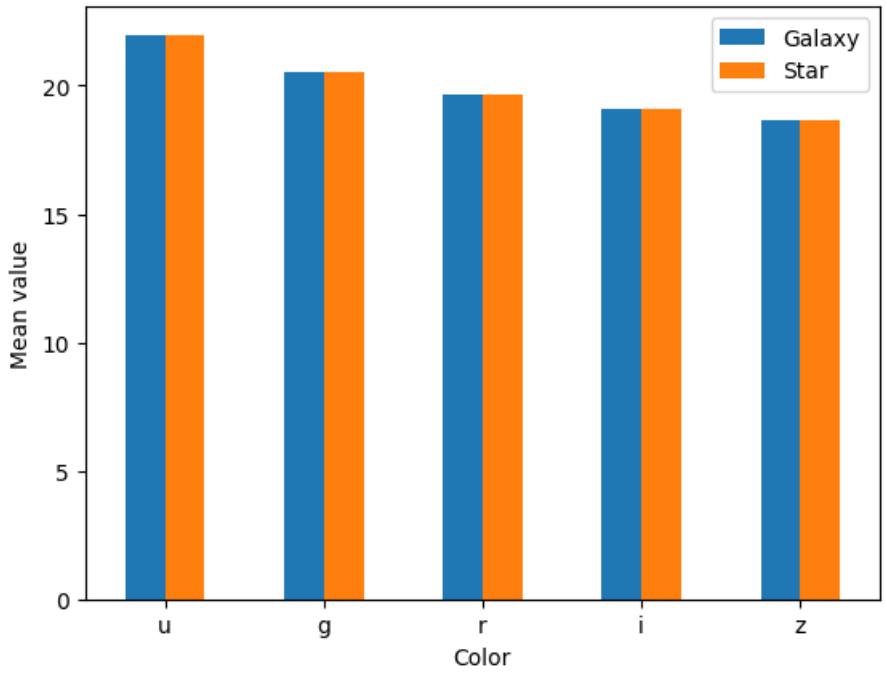




We used a MapReduce function and MapReduce transformation to classify galaxies and stars. This is the top 5 rows of the Sample classified data:

```
Sample classified data:
+--------+--------+--------+--------+--------+------+----+
|       u|       g|       r|       i|       z|galaxy|star|
+--------+--------+--------+--------+--------+------+----+
|23.87882| 22.2753|20.39501|19.16573|18.79371|     1|   0|
|24.77759|22.83188|22.58444|21.16812|21.61427|     1|   0|
|25.26307|22.66389|20.60976|19.34857|18.94827|     1|   0|
|22.13682|23.77656|21.61162|20.50454| 19.2501|     1|   0|
|19.43718|17.58028|16.49747|15.97711|15.54461|     1|   0|
+--------+--------+--------+--------+--------+------+----+
only showing top 5 rows
```

Plot for the mean colour values of galaxies , QSO and stars:



Counting the number of objects, in each class:

```
[('QSO', 18961), ('STAR', 21594), ('GALAXY', 59445)]
```

The average redshift for each class:

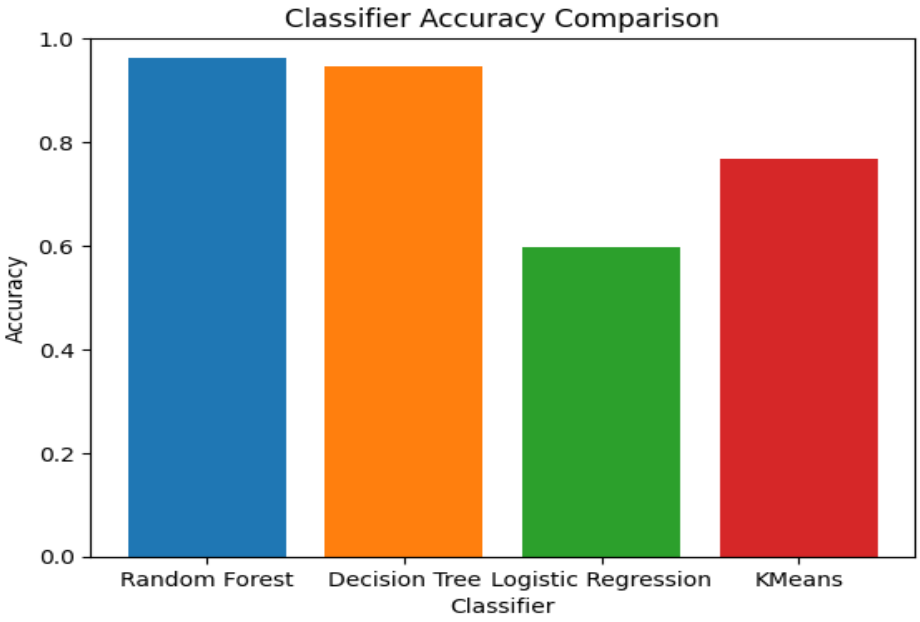
```
Found 3 distinct class-redshift pairs:
('QSO', 1.7196760892911382)
('STAR', -0.00011489720327557512)
('GALAXY', 0.4215961504447538)
```

```
[(('QSO', '>0.5'), 17986), (('STAR', '>0.5'), 13682), (('GALAXY', '0.0-0.1'), 8600),
```

First, we have named dataframe as data and we calculated the average for each class:

```
+------+-----------------+
| class|            avg_u|
+------+-----------------+
|GALAXY|22.58737929026834|
|   QSO|21.54761927324517|
|  STAR|20.68980531397607|
+------+-----------------+
```

Accuracy comparison for the ML algorithms:



RESULTS AND DISCUSSION

Logistic Regression:

Logistic Regression is a popular machine learning classification algorithm that can be implemented in Pyspark using the Logistic Regression class from the pyspark.ml.classification module.

To use the LogisticRegression class in Pyspark, you need to follow the following steps:

- Load the information into a DataFrame.
- Using StringIndexer, convert categorical variables to numerical variables.

- Using VectorAssembler, combine the features into a vector.
- Using randomSplit, divide the data into training and testing sets.
- Logistic Regression is used to define the logistic regression classifier.
- Fit is used to train the model using the training data.
- Using transform, evaluate the model using the testing data.

The accuracy for Logistic Regression is 59.71%.

Decision Tree Classifier:
A decision tree classifier is a well-known machine learning algorithm that can be used for classification and regression tasks. The Decision Tree Classifier class from the pyspark.ml.classification module can be used to implement decision tree classifiers in Pyspark.

The Decision Tree Classifier class accepts several parameters that can be tuned to improve the model's accuracy. The most critical parameters are:

maxDepth: This parameter specifies the decision tree's maximum depth. Increases in this parameter can result in overfitting, while decreases in this parameter can result in underfitting.

maxBins: The maximum number of bins used for discretizing continuous features is specified by this parameter. Increasing this parameter improves model accuracy while increasing computational complexity.

The accuracy for Decision tree classifier is 94.58%.

Random Forest Classifier:

A popular machine learning algorithm for classification tasks is the Random Forest Classifier. It is an ensemble learning method that constructs a large number of decision trees during training and outputs the class that is the mode of the individual trees' classes. PySpark is a robust distributed computing framework that allows for scalable machine learning workflows.

In PySpark, you can use the Random Forest Classifier class from the pyspark.ml.classification module to create a Random Forest Classifier model. First, import the required modules and create a SparkSession object. Then you can load your training data into a PySpark DataFrame and perform any data

preprocessing that is required.Then, create an instance of the Random Forest Classifier class and configure its hyperparameters. You can specify the number of trees in the forest, their maximum depth and the number of features to consider at each split.

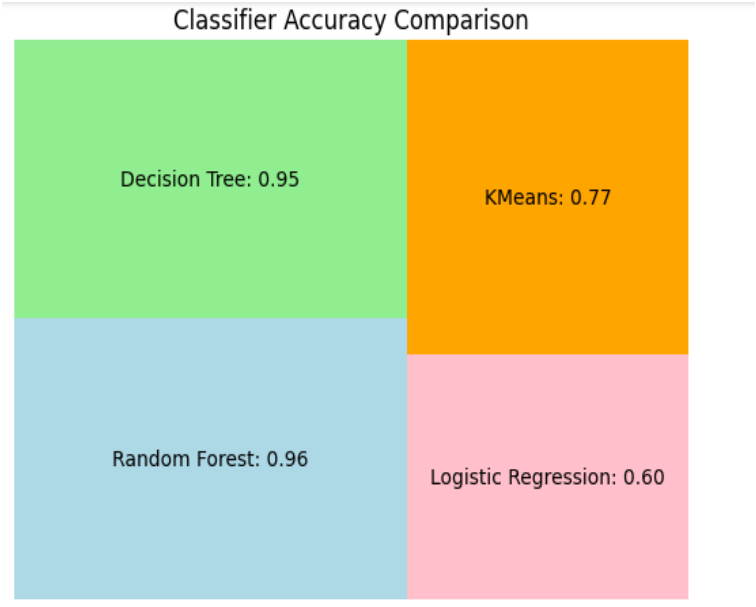
The accuracy for Random forest classifier is 96.56%.

K-Means Classifier:

K-Means is a well-known unsupervised machine learning algorithm that is commonly used for clustering tasks. Its goal is to divide a set of observations into K clusters, with each observation assigned to the cluster with the closest mean. PySpark is a robust distributed computing framework that allows for scalable machine learning workflows.

In PySpark, you can use the KMeans class from the pyspark.ml.clustering module to create a K-Means clustering model.First, import the required modules and create a SparkSession object. Then you can load your data into a PySpark DataFrame and perform any data preprocessing that is required.After that, you can make an instance of the KMeans class and configure its hyperparameters. You can specify the number of clusters (K), maximum number of iterations and initialization mode.

The accuracy for K-Means Classifier is 76.81%.



ANN:

Artificial Neural Networks (ANNs) are a type of deep learning model that may be used to perform a variety of machine learning tasks. Dataset preparation, designing the model architecture, training the model, hyperparameter tuning, model evaluation, fine-tuning, testing, results analysis, documentation, and model deployment are all part of the ANN implementation process. Backpropagation and SGD are used to train ANNs, and hyperparameters such as learning rate, batch size, and number of epochs are set to optimise model performance. Model performance is evaluated using parameters such as accuracy, precision, recall, and F1-score. Analysis and recording of results are critical for understanding model behaviour and communicating research findings to the community.
The accuracy for ANN model is 96.81%.

CONCLUSION

The classification of Stars is extremely important to know while investing. From this study it can be concluded that ML techniques such as Logistic Regression, Random Forest Classifier, K-Means Classifier and Decision Tree Classifier and deep learning model called ANN can be employed to classify the stars. We used Four ML algorithms and one deep learning algorithm using Pyspark for accuracy comparison- Logistic Regression- 59.71%, Random Forest Classifier-96.56%, Decision Tree Classifier-94.58%, K-Means Classifier-76.81%, ANN-96.81%.

There is 96.81% accuracy from the ANN.

Another models that we could depend on is the Random Forest Classifier and Decision Tree. It also works relatively fine.

By using the Pyspark, we have classified the stars using their Physical properties(temperature, luminosity).

FUTURE WORK

Future work shall include more regression models and time series data using pyspark in order to extend the precision of the stellar classification and also we can develop an app for Stellar Classification using an user interface.

REFERENCES

https://arxiv.org/pdf/2105.07110.pdf

https://www.hansshodhsudha.com/fourth-issue/April-June%20article%208.pdf

https://www.researchgate.net/publication/345694002_An_application_of_deep_learning_in_the_analysis_of_stellar_spectra

https://academic.oup.com/mnras/article/295/2/312/987486

https://downloads.hindawi.com/journals/aa/2022/4489359.pdf

https://www.sciencedirect.com/science/article/abs/pii/S0275106221000539

https://arxiv.org/pdf/1608.04369.pdf

https://arxiv.org/pdf/1909.05459.pdf

https://arxiv.org/pdf/1412.8341.pdf

https://arxiv.org/pdf/1709.09182.pdf

https://ui.adsabs.harvard.edu/abs/2018MNRAS.475.2978F/abstract