DAYANANDA SAGAR UNIVERSITY

KUDLU GATE, BANGALORE – 560068

Bachelor of Technology
in
COMPUTER SCIENCE AND ENGINEERING

Major Project Phase-II Report

CLASSIFICATION OF GALAXIES BASED ON MORPHOLOGY AND DETECTION OF POTENTIAL EXOPLANETS

By
Rahul Noronha - ENG18CS0222
Sahana M - ENG18CS0240
Sandesh Bhat - ENG18CS0242

Under the supervision of

Dr. Monika Goyal
Assistant Professor, CSE Department

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING,
SCHOOL OF ENGINEERING
DAYANANDA SAGAR UNIVERSITY

(2021-2022)

DAYANANDA SAGAR UNIVERSITY

School of Engineering
Department of Computer Science & Engineering
Kudlu Gate, Bangalore – 560068
Karnataka, India

CERTIFICATE

This is to certify that the Phase-II project work titled **"Classification of Galaxies based on Morphology and detection of Potential Exoplanets"** is carried out by **Rahul Noronha (ENG18CS0222), Sahana M (ENG18CS0240), Sandesh Bhat (ENG18CS0242)** bonafide students of Bachelor of Technology in Computer Science and Engineering at the School of Engineering, Dayananda Sagar University, Bangalore in partial fulfillment for the award of degree in Bachelor of Technology in Computer Science and Engineering, during the year **2021-2022**.

Dr. Monika Goyal	**Dr. Girisha G S**	**Dr. A Srinivas**
Assistant Professor Dept. of CS&E, School of Engineering Dayananda Sagar University	Chairman CSE School of Engineering Dayananda Sagar University	Dean School of Engineering Dayananda Sagar University
Date:	Date:	Date:

Name of the Examiner **Signature of Examiner**

1.

2.

DECLARATION

We, **Rahul Noronha (ENG18CS0222), Sahana M (ENG18CS0240), Sandesh Bhat (ENG18CS0242),** are students of the eighth semester B.Tech in **Computer Science and Engineering**, at School of Engineering, **Dayananda Sagar University**, hereby declare that the phase-II project titled **"Classification of Galaxies based on Morphology and detection of Potential Exoplanets"** has been carried out by us and submitted in partial fulfillment for the award of degree in **Bachelor of Technology in Computer Science and Engineering** during the academic year **2021-2022**.

Student	**Signature**
Name1: Rahul Noronha	
USN : ENG18CS0222	
Name2 : Sahana M	
USN : ENG18CS0240	
Name3: Sandesh Bhat	
USN : ENG18CS0242	

Place : Bangalore

Date :

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this project work.

First, we take this opportunity to express our sincere gratitude to School of Engineering & Technology, Dayananda Sagar University for providing us with a great opportunity to pursue our Bachelor's degree in this institution.

We would like to thank **Dr. A Srinivas. Dean**, **School of Engineering & Technology**, **Dayananda Sagar University** for his constant encouragement and expert advice. It is a matter of immense pleasure to express our sincere thanks to **Dr. Girisha G S, Department Chairman**, **Computer Science, and Engineering**, **Dayananda Sagar University,** for providing the right academic guidance that made our task possible.

We would like to thank our guide **Dr. Monika Goyal, Assistant Professor**, **Dept. of Computer Science and Engineering**, **Dayananda Sagar University**, for sparing his/her valuable time to extend help in every step of our project work, which paved the way for smooth progress and the fruitful culmination of the project.

We would like to thank our Project Coordinator **Dr. Meenakshi Malhotra** and **Dr. Bharanidharan N**, and all the staff members of Computer Science and Engineering for their support.

We are also grateful to our family and friends who provided us with every requirement throughout the course. We would like to thank one and all who directly or indirectly helped us in the Project work.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

NASA	National Aeronautics and Space Administration.
RAM	Random Access Memory
GB	Giga Bytes
AMD	Advanced Micro Devices
GPU	General
TPU	Tensor Processing Units
PyPI	Python Package Index
CSS3	Cascading Style Sheets 3
HTML5	Hyper Text Markup Language 5
ANN	Artificial Neural Network
ResNet50	Residual Neural Network
XG Boost	eXtreme Gradient Boost
KNN	K-Nearest Neighbors
EDA	Exploratory Data Analysis
ANFIS	Adaptive Neuro-Fuzzy Inference System
CNN	Convolutional Neural Network
MAE	Mean Absolute Error

LIST OF FIGURES

Figure number Description of the figure Page No.

LIST OF TABLES

ABSTRACT

In modern Astronomy due to the abundance of data, both numerical and pictorial, it is becoming increasingly difficult to manually analyze and classify the data. One such attempt is the crowdsourced Galaxy Zoo project. Galaxy Zoo 1, and Galaxy Zoo 2 aim to perform the morphological classification of a large number of galaxies. Since the manual classification of the galaxies even with the help of several people around the world takes a long time, it has become essential to come up with an automated process to replace this manual task. In the first part of this study, we classify the galaxies based on their morphologies into their various shapes namely Spiral, Barred Spiral, Elliptical, and irregular by using transfer learning and incremental learning. We follow a question-based approach which contains a 37-class vector that is created using the probability distribution of the responses to the 11 questions in the Galaxy Zoo project to give a Decision Tree obtained from the Hubble classification schema. We combine the ResNet50 model of ImageNet with another befitting model from ImageNet to create a hybrid model. In the second part of the study, we determine the potential candidate for an exoplanet using Caltech's NASA Exoplanet Archive. Dimensionality Reduction is performed to select the best among the 42 features using forward feature selection along with other feature selection techniques and trained over a Multi-Layered Perceptron (MLP) and other machine learning models like Logistic Regression, Tree-based, and ensemble learning models.

CHAPTER 1

INTRODUCTION

Chapter 1 INTRODUCTION

Understanding how and why we are here is one of the fundamental questions for the human race. Part of the answer to this question lies in the origins of galaxies, such as our own Milky Way. In order to better understand how the different shapes (or morphologies) of galaxies relate to the physics that create them, such images need to be sorted and classified.

The Kepler Space Observatory is a NASA-built satellite that was launched in 2009. The telescope is dedicated to searching for exoplanets in star systems besides our own, with the ultimate goal of possibly finding other habitable planets besides our own.

1.1 PURPOSE

The Galaxy Zoo [1] project is a crowd-sourced astronomy galaxy classification endeavour whose results can have significant benefits to astronomers. This project is focused on using Machine Learning and transfer learning [2] techniques to detect a galaxy from an image obtained from a telescope, and to classify them based on their Morphologies, and identify if an observation is a real candidate for a potential exoplanet [3] [4].

The end-product is to develop an application with an efficient and easy to use Interface which can automate the process of detection and classification of the galaxies based on the image input by the user.

1.1.1 Intended Audience

The project helps Astronomers and Astrophysicists to get insights about the Expansion of our Universe based on Redshift and Blueshift (**Appendix A**) and about the formation of the galaxies after the Big Bang. This project is also intended to help in academics and the field of research.

1.2 TABLES

Table 1.1 Rules based mapping for Galaxy Zoo onto Hubble Types [1]

Task	Question	Response	Next
01	Is the galaxy simply smooth and rounded, with no sign of a disk?	smooth	07
		features or disk	02
		star or artifact	end
02	Could this be a disk viewed edge-on?	yes	09
		no	03
03	Is there a sign of a bar feature through the centre of the galaxy?	yes	04
		no	04
04	Is there any sign of spiral arm pattern	yes	10
		no	05
05	How prominent is the central bulge, compared with the rest of the galaxy?	no bulge	06
		just noticeable	06
		obvious	06
		dominant	06
06	Is there anything odd	yes	08
		no	end
07	How rounded is it?	complete round	06
		in between	06
		cigar-shaped	06
08		ring	end

	Is the odd feature a ring, or is the galaxy disturbed or irregular?	lens or arc	end
		disturbed	end
		irregular	end
		other	end
		merger	end
		dust lane	end
09	Does the galaxy have a bulge at its centre? If so, what shape?	rounded	06
		boxy	06
		no bulge	06
10	How tightly wound do the spiral arms appear?	tight	11
		medium	11
		loose	11
11	How many spiral arms are there?	1	05
		2	05
		3	05
		4	05
		more than four	05
		can't tell	05

1.3 SCOPE

Morphology (**Appendix B**) is still a logical starting point for understanding galaxies. Sorting galaxies into their morphological categories is like sorting stars into spectral types and can lead to important astrophysical insights. Any theory of galaxy formation and evolution will have to, at some point, account for the bewildering array of galactic forms. Galaxy morphology is strongly correlated with galactic star formation history. Classical morphology recognizes these differences in an ordered way. Finding exoplanets opens a vast exploration area to look for other habitable worlds. The scope of the project is to build an easy-to-use tool where astrophysicists and space enthusiasts alike can join and classify the Hubble type (**Appendix C**) of the galaxy images or find out if the planet is an exoplanet (**Appendix D**) based on some planetary measurement data.

CHAPTER 2

PROBLEM DEFINITION

Chapter 2 PROBLEM DEFINITION

Classify the Galaxy Morphologies based on the Hubble Scheme by adopting rules based approach and using supervised Machine learning to identify candidates for potential Exoplanets using features from Kepler data.

CHAPTER 3

LITERATURE REVIEW

Chapter 3 LITERATURE REVIEW

Table 3.1 Literature Review Summary

Sl. no.	Title	Authors	Description	Year
1	A rules-based and Transfer Learning approach for deriving the Hubble type of a galaxy from the Galaxy Zoo data	M. Z Variawa; T. L van Zyl; M. Woolway	The paper compares two approaches to Hubble Type classification of a galaxy. Firstly, a rules based approach that uses a Galaxy Zoo decision tree based on crowd sourced responses to 37 questions which serve as input vectors to manually map to a Hubble type, and secondly a Transfer Learning based approach where the Galaxy Zoo data is used as an initial input for the ResNet50 which is then passed to a model which is trained using the RSA catalogue data which contained a much smaller number of correctly classified values (1246 galaxies).	2020
2	Comparing Generalization Using Crowd-Sourced vs Expert Labels for Galaxies Classification	M. Z Variawa; T.L. van Zyl; M. Woolway	This paper uses the Galaxy Zoo 2 data to train a base ResNet-50 model for transfer learning which we use to predict the Hubble types of Galaxies in the Revised Shapley-Ames catalogue. This shows the lower accuracy of crowd sourced	2020

			labelling based models for data than data which undergoes expert labelling like the RSA catalogue of bright galaxies.	
3	Classification of Galaxy Morphologies using Artificial Neural Network	Manish Biswas; Ritesh Adlak	This paper proposes the use of the artificial neural network (ANN) to classify the galaxy morphologies into three classes. This Machine Learning algorithm classifies galaxies into a spiral, elliptical and irregular using data (images of galaxies) collected from Sloan Digital Sky Survey and Galaxy Zoo project .	2018
4	Habitability of Exoplanets using Deep Learning	R. Jagtap, U. Inamdar, S. Dere, M. Fatima and N. B. Shardoor	This paper has proposed a deep learning model which provides an automatic way to examine catalogs of interplanetary objects and, detect and classify planets based on their habitable behaviour. The ASTRONET deep learning model is used to predict anomalies in astronomical bodies.	2021h

5	Refining Exoplanet Detection Using Supervised Learning and Feature Engineering	M. Bugueño, F. Mena and M. Araya	The paper mainly deals with the supervised machine learning approach to analyse light curves. it aims to cut down the tedious time consuming techniques currently used by the scientists by automating the processes using machine learning and feature engineering	2019

CHAPTER 4

PROJECT DESCRIPTION

Chapter 4 PROJECT DESCRIPTION

4.1 PROPOSED DESIGN

The high-level project design of our project can be seen in the figure 4.1.1. The project has two parts, namely potential exoplanet detection and galaxy morphology classification.

In Exoplanet detection we first get the data from Caltech NASA Kepler Objects of Interest (KOI) dataset. We then clean the Dataset imported as a DataFrame. We then perform exploratory data analysis techniques on the data. Then we compare the performance of the various models starting from Multilayer perceptron to the Machine learning models like Logistic Regression, Naive Bayes', Decision Tree, XG Boost and so on. Then we calculate the evaluation metrics and select the best performing model to host on a website as an Exoplanet classifier.

In Galaxy morphology classification we gather the dataset from Kaggle for Galaxy Zoo 1 and from Galaxy Zoo Project data release website for Galaxy Zoo 2 project. These include images along with CSVs. Clean the CSV data by importing it into a DataFrame, then apply various image transformation and cleaning techniques on the Galaxy images. Use transfer learning and train these images along with the CSV on an ImageNet model like ResNet50, or alternatively using a Hybrid model like the InceptionResNetV2 for transfer learning. Train for several epochs and then save the best model based on the performance evaluation metrics to be used to deploy as a Galaxy classifier on the website.

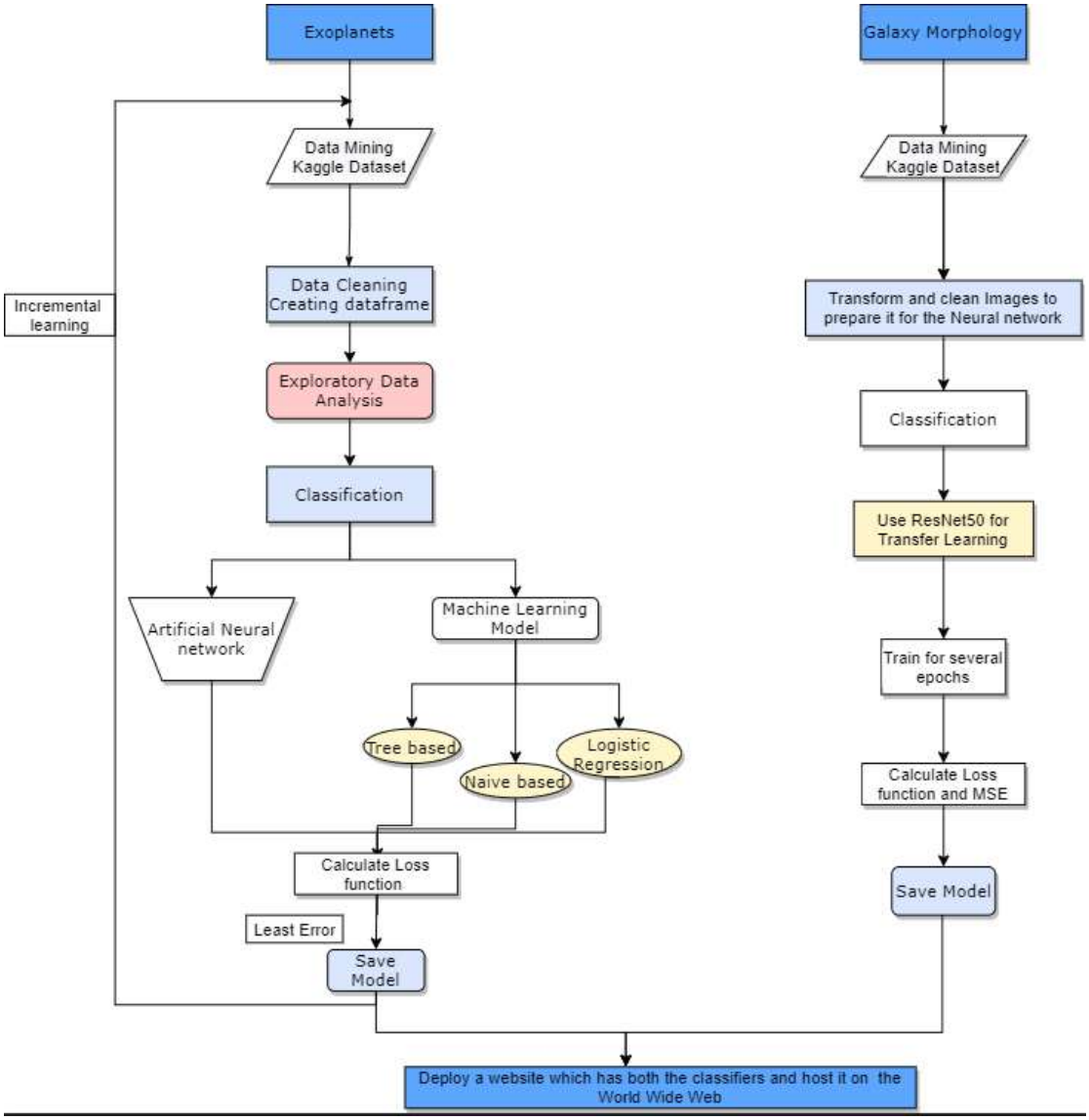


Figure 4.1.1 Project Flow including both the components of the two classification tasks.

4.2 ASSUMPTIONS AND DEPENDENCIES

The Kepler Telescope was able to find planets by looking for small dips in the brightness of a star when a planet transits in front of it. It is possible to measure the size of the planet based on the depth of the transit and the star's size. It is a fair assumption to make that the devices are calibrated perfectly and there is no error in the calculation of the parameters required to train a machine learning model obtained from open-source datasets. Another assumption is that the application is used on a computer with enough performance ability, and the use of an up-to-date internet browser. The additional dependencies are Python PyPI modules which tend to get some feature changes from version to version due to their robustness and continuous bug fixes.

CHAPTER 5

REQUIREMENTS

Chapter 5 REQUIREMENTS

5.1 FUNCTIONAL REQUIREMENTS

- Web scraping/ Data Mining: Collecting all the images and other data from open-source websites.
- Feature Engineering.
- Image and data pre-processing module to prepare the data we mined before subjecting it to Model building.
- Machine Learning model and ANN (Artificial Neural Network) models, out of which we will select the best one for deployment, for potential exoplanet classification and a transfer learning model with incremental learning for Galaxy Morphology.
- Evaluating the model's based on the loss function and save the one with minimal error and save the model.
- Model to perform incremental learning in Exoplanet classification.
- Develop an application using React.JS and Flask to automate the prediction.
- Deploy it on a cloud server.

5.2 NON-FUNCTIONAL REQUIREMENTS

5.2.1 Performance Requirements

The classification models built need to have a good accuracy in classifying data points and we can measure them using various statistical tools.

The application needs to be able to respond to user queries in a short span of time accurately and perform relative to the amount of resources used under stated conditions.

5.2.2 Software Quality Attributes

The built application must provide easy availability, correctness, flexibility, and usability to the user to achieve specific goals with effectiveness, efficiency, and satisfaction in a specified context of use. The application must operate as intended despite the presence of hardware or software faults.

5.3 SOFTWARE REQUIREMENTS

Jupyter notebook / Google Colab

Python3.x.x

Scikit Learn

TensorFlow

PyTorch

Keras

Matplotlib

Seaborn

React.js

HTML5

CSS3

JavaScript

Flask

&Other Python3 PyPI modules.

5.4 HARDWARE REQUIREMENTS

8 GB RAM and above recommended

10 GB internal storage and above recommended

Intel core i3 processor and above

AMD Ryzen 3 processor and above recommended

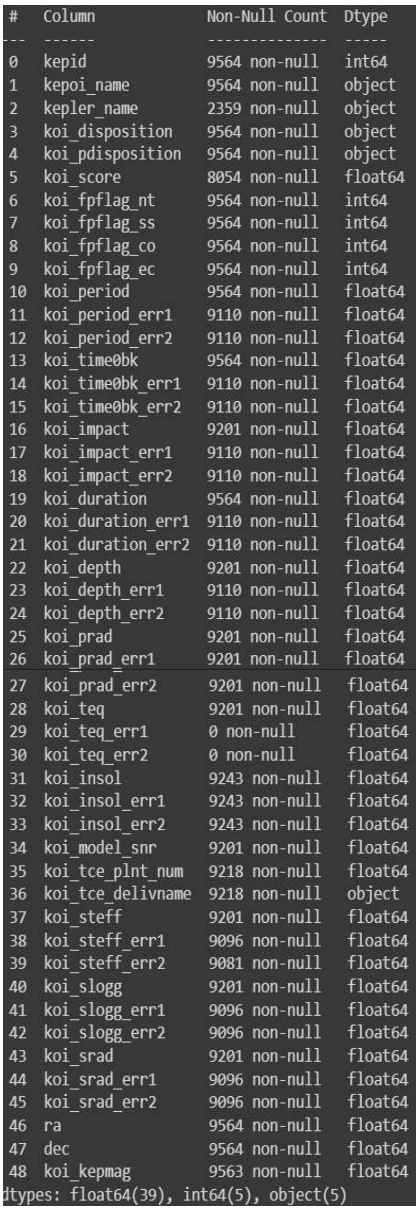
CHAPTER 6

METHODOLOGY

Chapter 6 METHODOLOGY

Exoplanet (Potential Exoplanet Classification):

- Data Mining from Kaggle or Caltech's NASA Exoplanet Archive to obtain 48 features in the Kepler data.



#	Column	Non-Null Count	Dtype
0	kepid	9564 non-null	int64
1	kepoi_name	9564 non-null	object
2	kepler_name	2359 non-null	object
3	koi_disposition	9564 non-null	object
4	koi_pdisposition	9564 non-null	object
5	koi_score	8054 non-null	float64
6	koi_fpflag_nt	9564 non-null	int64
7	koi_fpflag_ss	9564 non-null	int64
8	koi_fpflag_co	9564 non-null	int64
9	koi_fpflag_ec	9564 non-null	int64
10	koi_period	9564 non-null	float64
11	koi_period_err1	9110 non-null	float64
12	koi_period_err2	9110 non-null	float64
13	koi_time0bk	9564 non-null	float64
14	koi_time0bk_err1	9110 non-null	float64
15	koi_time0bk_err2	9110 non-null	float64
16	koi_impact	9201 non-null	float64
17	koi_impact_err1	9110 non-null	float64
18	koi_impact_err2	9110 non-null	float64
19	koi_duration	9564 non-null	float64
20	koi_duration_err1	9110 non-null	float64
21	koi_duration_err2	9110 non-null	float64
22	koi_depth	9201 non-null	float64
23	koi_depth_err1	9110 non-null	float64
24	koi_depth_err2	9110 non-null	float64
25	koi_prad	9201 non-null	float64
26	koi_prad_err1	9201 non-null	float64
27	koi_prad_err2	9201 non-null	float64
28	koi_teq	9201 non-null	float64
29	koi_teq_err1	0 non-null	float64
30	koi_teq_err2	0 non-null	float64
31	koi_insol	9243 non-null	float64
32	koi_insol_err1	9243 non-null	float64
33	koi_insol_err2	9243 non-null	float64
34	koi_model_snr	9201 non-null	float64
35	koi_tce_plnt_num	9218 non-null	float64
36	koi_tce_delivname	9218 non-null	object
37	koi_steff	9201 non-null	float64
38	koi_steff_err1	9096 non-null	float64
39	koi_steff_err2	9081 non-null	float64
40	koi_slogg	9201 non-null	float64
41	koi_slogg_err1	9096 non-null	float64
42	koi_slogg_err2	9096 non-null	float64
43	koi_srad	9201 non-null	float64
44	koi_srad_err1	9096 non-null	float64
45	koi_srad_err2	9096 non-null	float64
46	ra	9564 non-null	float64
47	dec	9564 non-null	float64
48	koi_kepmag	9563 non-null	float64

dtypes: float64(39), int64(5), object(5)

Figure 6.1 Features in Kepler Data

- Data Pre-processing
    - Irrelevant columns were dropped.

```
Dropping Irrelevant Columns from the dataframe

[ ]  1  df.drop(columns=['kepid', 'kepoi_name', 'kepler_name', 'koi_teq_err1', 'koi_teq_err2'], inplace=True)
```

Figure 6.2 Drop irrelevant columns

    - Handled null values - Regression Imputation was performed to handle the missing values.

```
def random_imputation(df, feature):

    number_missing = df[feature].isnull().sum()
    observed_values = df.loc[df[feature].notnull(), feature]
    df.loc[df[feature].isnull(), feature + '_imp'] = np.random.choice(observed_values, number_missing, replace = True)

    return df
num = 0
random.shuffle(fillna)
deliveryName = list(df['koi_tce_delivname'])
for i in range(len(deliveryName)):
  if deliveryName[i] == -1:
    deliveryName[i] = fillna[num]
    num += 1
deliveryName
```

Figure 6.3 Regression Imputation

        - Processed categorical data.

- Performed Dimensionality Reduction
    - Quasi constant - Quasi constant features are those features where one of the values are dominant 99.9%.
    - Information gain - Information Gain is the measure of how much information a feature provides about a class.
    - Correlation matrix - Correlation matrix is used to demonstrate a linear relationship between the coefficients of different variables.
    - Forward Feature Selection – In Forward feature Selection variant features are sequentially added to an empty set of features until the addition of extra features does not reduce the criterion.

- Used supervised algorithm and tested the results on validation dataset
    - Multi-Layered Perceptron
        - The model uses 3 hidden layers with ReLU activation function and 1 output layer with Sigmoid function.
        - 6 units were used in each layer.
        - Adam Optimizer is used as an optimizer
        - Loss function used is mean squared error.
    - Decision Tree
    - Logistic Regression
    - XG Boost
    - Random forest
- Trained and saved the model.
- Deploy on the web as a classifier tool.

Galaxy Morphology (Galaxy Zoo Data) Finding Hubble type of a Galaxy:

- Obtained the dataset. (Kaggle)
    - First we download the data which are stored in zip format on Kaggle under the competition.

```
import zipfile

zipFilesTrain = ['images_training_rev1', 'training_solutions_rev1']
zipFileTest = ['images_test_rev1']
labels = ['all_ones_benchmark', 'all_zeros_benchmark', 'central_pixel_benchmark']
for train in zipFilesTrain:
    with zipfile.ZipFile('../Data/GalaxyZoo1/' + train + '.zip', 'r') as zip_ref:
        zip_ref.extractall('../Data/GalaxyZoo1/train')
for test in zipFileTest:
    with zipfile.ZipFile('../Data/GalaxyZoo1/' + test + '.zip', 'r') as zip_ref:
        zip_ref.extractall('../Data/GalaxyZoo1/test')
for label in labels:
    with zipfile.ZipFile('../Data/GalaxyZoo1/' + label + '.zip', 'r') as zip_ref:
        zip_ref.extractall('../Data/GalaxyZoo1/labels')
with zipfile.ZipFile('../Data/GalaxyZoo2/galaxy-zoo-2-images.zip', 'r') as zip_ref:
    zip_ref.extractall('../Data/GalaxyZoo2')
```

Figure 6.4 Unzipped the images and data Galaxy Zoo 1 using Python script

- The Galaxy Zoo 2 dataset CSV is saved as DataFrame using the pandas compression parameter compression = ‘gzip’.

```
labels = pd.read_csv('/content/drive/MyDrive/Major Project/Galaxy_Morphology/Data/Galaxy_Zoo-2/gz2_classes.csv', compression='gzip')
labels
```

Figure 6.5 CSV saved in DataFrame using compression=’gzip’

Figure 6.6 Typical galaxy image

- Pre-processing of data from Galaxy Zoo 2 dataset.
  - Irrelevant columns were dropped

```
labels.drop(['ra', 'dec', 'rastring', 'decstring', 'sample', 'gz2_class', 'total_classifications', 'total_votes'], inplace = True ,axis = 1)

columnsToDrop = labels.columns
columnsToDrop = list(columnsToDrop[1:])

for columns in columnsToDrop:
    if columns[len(columns) - 8:] == 'debiased':
        columnsToDrop.remove(columns)

labels.drop(columnsToDrop, axis = 1, inplace=True)
```

Figure 6.7 Dropping irrelevant columns

- The columns in the dataset having the last few letters as 'debiased' are the debiased columns of the dataset and these are dropped.

```
columnsToDrop = labels.columns
columnsToDrop = list(columnsToDrop[1:])

for columns in columnsToDrop:
    if columns[len(columns) - 8:] == 'debiased':
        columnsToDrop.remove(columns)

labels.drop(columnsToDrop, axis = 1, inplace=True)
```

Figure 6.8 Dropped debiased columns.

- Downloaded the ResNet 50 model from TensorFlow.
  - The ResNet50 ImageNet model is downloaded from TensorFlow website after removing it's top layer by setting include_top to False.

```
img_shape = (224, 224, 3)
resnet_model = ResNet50(include_top=False, input_shape=img_shape)
```

Figure 6.9 Getting ResNet50 without top layer using TensorFlow

- Applied filters and changed the RGB image size to 224 x 224.
  - The Images of the Galaxies are processed using keras ImageDataGenerator and they are resized to 224x224 pixels in size before training and split into training and validation images using validation split property.

```
datagenerator = ImageDataGenerator(
    fill_mode='nearest',
    cval=0,
    rescale=1/255,
    rotation_range=90,
    width_shift_range=0.1,
    height_shift_range=0.1,
    horizontal_flip=True,
    vertical_flip=True,
    validation_split=0.02)
```

Figure 6.10 Image processing and resizing

  - The data associated with each image is stored in CSV format and this probability distribution of the 37 classes of Galaxy Zoo project are input

using flow_from_dataframe which is a method of ImageDataGenerator class.

```
train_generator = datagenerator.flow_from_dataframe(
    dataframe=traindf,
    directory="../Data/GalaxyZoo1/train/images_training_rev1/",
    x_col="id",
    y_col=classes,
    subset="training",
    batch_size=16,
    seed=123,
    shuffle=True,
    class_mode="raw",
    target_size=(224, 224))

validation_generator = datagenerator.flow_from_dataframe(
    dataframe=traindf,
    directory="../Data/GalaxyZoo1/train/images_training_rev1/",
    x_col="id",
    y_col=classes,
    subset="validation",
    batch_size=16,
    seed=123,
    shuffle=True,
    class_mode="raw",
    target_size=(224, 224))

STEP_SIZE_TRAIN = train_generator.n // train_generator.batch_size
STEP_SIZE_VALID = validation_generator.n // validation_generator.batch_size
```

Figure 6.11 Data associated with images are input using flow_from_dataframe

- Used ResNet50 (**Appendix E**) for Transfer Learning.
- Adam Optimizer is used as an optimizer.
- A Class is set up to store model checkpoints, define early stopping parameters of training and to set the callbacks.
- Trained for 30 epochs and saved the model in hdf5 format.
- Calculated the validation loss and validation accuracy.
- Plotted a graph with validation loss and training loss against number of epochs.

CHAPTER 7

EXPERIMENTATION

Chapter 7 EXPERIMENTATION

Potential Exoplanet Detection

- Performed Dimensionality Reduction
    - Quasi Constant – 5 features were removed.

```
from sklearn.feature_selection import VarianceThreshold
quasiConstant = VarianceThreshold(threshold=0.01)
quasiConstant.fit(X)
quasiConstantColumns = [x for x in df.iloc[:,:-1].columns if x not in df.iloc[:,:-1].columns[quasiConstant.get_support()]]
quasiConstantColumns
```

```
['Orbital Period Upper',
 'Orbital Period Lower',
 'Transit Epoch Upper',
 'ransit Epoch Lower',
 'Stellar Surface Gravity Lower']
```

Figure 7.1 Quasi constant features removed

- Information Gain – Remove features with information gain below 0.025.

```
def plotFeatureImportance(df, importance, figsize=(15, 15), color='teal'):
    import matplotlib.pyplot as plt
    %matplotlib inline

    plt.figure(figsize=figsize)
    feat_importances = pd.Series(importances, df.columns[0:len(df.columns)-1])
    feat_importances.plot(kind = 'barh', color = color)
    plt.rc_context({'axes.edgecolor':'#581845', 'xtick.color':'black', 'ytick.color':'black', 'figure.facecolor':'#F1D8DD'})
    plt.show()
```

Figure 7.2 Information gain used and features with value below 0.025 removed

- Correlation matrix – Removes features greater than correlation of 0.85.

```
corrMatrix = corrMatrix.where(np.triu(np.ones(corrMatrix.shape), k=1).astype(np.bool))
[column for column in corrMatrix.columns if any(corrMatrix[column] > 0.85)]
```

```
['Insolation Flux Upper']
```

Figure 7.3 Correlation matrix used and values above 0.85 removed

- Forward Feature Selection

```
def sequentialFeatureSelector(df, classifier, X, y):
    ffs = SequentialFeatureSelector(classifier,k_features='best',forward=True,n_jobs=1)
    ffs.fit(X, y)
    features = list(ffs.k_feature_names_)
    features = list(map(int, features))

    selectedColumns = []
    cols = list(df.columns)
    print(features)
    for index in features:
        if index > 34:
            continue
        selectedColumns.append(cols[index])
    selectedColumns.append('Disposition Using Kepler Data')
    print(selectedColumns)
    return df[selectedColumns]
```

Figure 7.4 Forward feature selection

- Used supervised algorithm and tested the results on validation dataset
  - Logistic Regression

```
def LR_Classifier(self):
    print("\nTraining the Logistic Regression model...\n")
    LRClassifier = LogisticRegression()
    LRClassifier.fit(self.X_train , self.y_train)
    self.displayConfusionMatrix(LRClassifier)
```

Figure 7.5 Logistic Regression for Exoplanet Detection

  - Random Forest

```
def RF_Classifier(self):
    print("\nTraining the Random Forest model...\n")
    RFClassifier = RandomForestClassifier(n_estimators = 50, criterion = 'gini',random_state = 41)
    RFClassifier.fit(self.X_train , self.y_train)
    self.displayConfusionMatrix(RFClassifier)
```

Figure 7.6 Random Forest for Exoplanet Detection

  - Naive Bayes

```
def NB_Classifier(self):
    print("\nTraining the Gaussian Naive Bayes model...\n")
    NBClassifier = GaussianNB()
    NBClassifier.fit(self.X_train , self.y_train)
    self.displayConfusionMatrix(NBClassifier)
```

Figure 7.7 Naïve Bayes for Exoplanet Detection

- Decision Tree

```
def DT_Classifier(self):
    print("\nTraining the Decision Tree model...\n")
    DFClassifier = DecisionTreeClassifier(criterion = 'entropy', random_state = 41)
    DFClassifier.fit(self.X_train , self.y_train)
    self.displayConfusionMatrix(DFClassifier)
```

Figure 7.8 Decision Tree for Exoplanet Detection

- XG Boost

```
def XG_boost(self):
    XGBclassifier = XGBClassifier()
    XGBclassifier.fit(self.X_train, self.y_train)
    self.displayConfusionMatrix(XGBclassifier)
```

Figure 7.9 XG Boost for Exoplanet Detection

Galaxy Morphology

- Adam optimizer with learning rate of 0.001 and decay of $5x10^{-4.}$

```
optimizer = keras.optimizers.Adam(learning_rate=0.001, decay=5e-4)

model.compile(optimizer, loss='mse', metrics=["accuracy"])
```

Figure 7.10 Adam Optimizer for Galaxy Morphology

- Loss function used is mean squared error.

```
model.compile(optimizer, loss='mse', metrics=["accuracy"])
```

Figure 7.11 Loss function of Mean squared error is used

- A class is set up to store model checkpoints, define early stopping parameters of training and to set the callbacks.

```
from keras.callbacks import Callback
from keras.callbacks import ModelCheckpoint, Callback, EarlyStopping


class LossHistory(Callback):
    def on_train_begin(self, logs={}):
        self.losses = []
        self.val_losses = []

    def on_batch_end(self, batch, logs={}):
        self.losses.append(logs.get('loss'))
        self.val_losses.append(logs.get('val_loss'))


early_stopping = EarlyStopping(
    monitor='val_loss', patience=4, verbose=1, mode='auto')

history = LossHistory()

from keras.callbacks import ModelCheckpoint
checkpointer = ModelCheckpoint(
    filepath='../../weights.hdf5', verbose=2, save_best_only=True)
```

Figure 7.12 Callbacks, model checkpoint and early stopping parameters are set up in a class LossHistory

- Trained for 30 epochs and saved the model in hdf5 format.

Figure 7.13 weights.hdf5 stores the trained model

- Unavailability of powerful compute GPUs. To tackle this CUDA and CuDNN drivers were downloaded, and the model were trained on laptop's 4 GB NVIDIA GPU.

Potential Exoplanet Detection

- The Supervised model, especially the tree-based algorithm seemed to be overfitted while training due to several features in the dataset but was eventually dealt with by performing dimensionality reduction.
- The validation accuracy reduced drastically when trained after freezing all layers except the top layer of the ResNet50 for Galaxy Zoo 1.

- Unavailability of powerful compute GPUs. To tackle this CUDA and CuDNN drivers were downloaded, and the model were trained on laptop's 4 GB NVIDIA GPU.

Potential Exoplanet Detection

- The Supervised model, especially the tree-based algorithm seemed to be overfitted while training due to several features in the dataset but was eventually dealt with by performing dimensionality reduction.
- The validation accuracy reduced drastically when trained after freezing all layers except the top layer of the ResNet50 for Galaxy Zoo 1.

CHAPTER 8

TESTING AND RESULTS

Chapter 8 TESTING AND RESULTS

- Exoplanet
    - Logistic Regression
        - After dimensionality reduction, 8 features were used for Logistic Regression.
        - After comparative analysis Logistic Regression gave best results for all the tests.

Accuracy Score : 0.9870347135089921

	precision	recall	f1-score	support
CANDIDATE	0.98	0.99	0.99	1143
FALSE POSITIVE	0.99	0.98	0.99	1248
accuracy			0.99	2391
macro avg	0.99	0.99	0.99	2391
weighted avg	0.99	0.99	0.99	2391

Figure 8.1 Accuracy and other scores for Logistic Regression

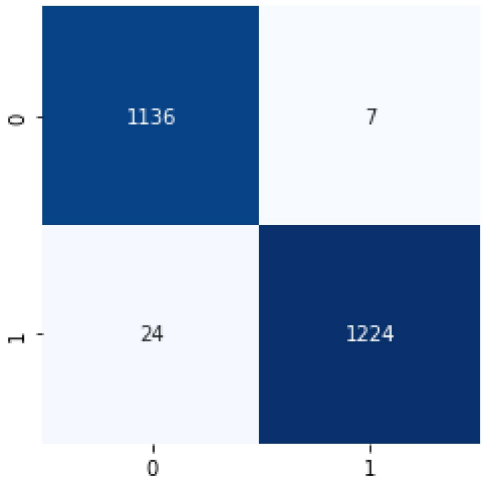


Figure 8.2 Confusion matrix for Logistic Regression

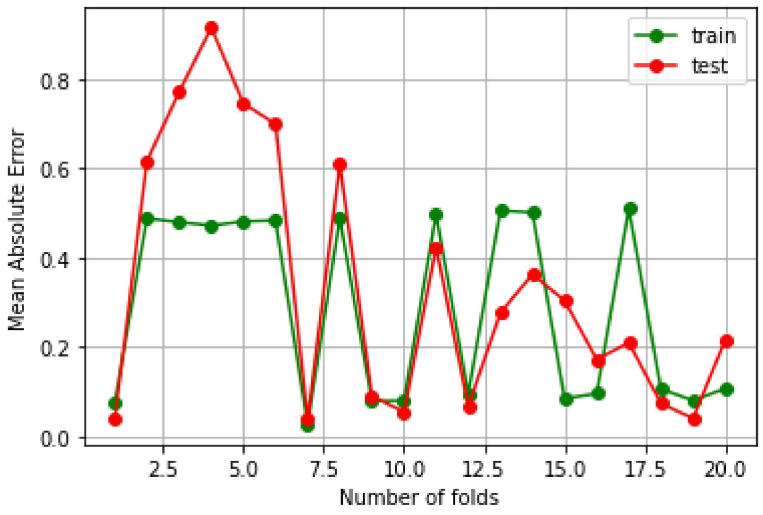


Figure 8.3 MAE vs No. Of folds for 20-fold cross validation for Logistic Regression

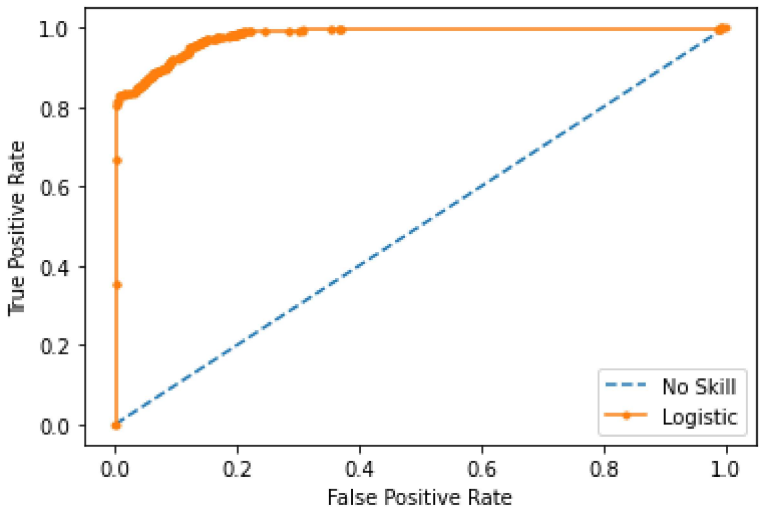


Figure 8.4 ROC curve for Logistic Regression

- Naive Bayes

```
Accuracy Score :  0.9849435382685069

                precision    recall  f1-score   support

     CANDIDATE       0.98      0.99      0.98      1143
FALSE POSITIVE       0.99      0.98      0.99      1248

      accuracy                           0.98      2391
     macro avg       0.98      0.99      0.98      2391
  weighted avg       0.98      0.98      0.98      2391
```

Figure 8.5 Accuracy and other scores for Naïve Bayes

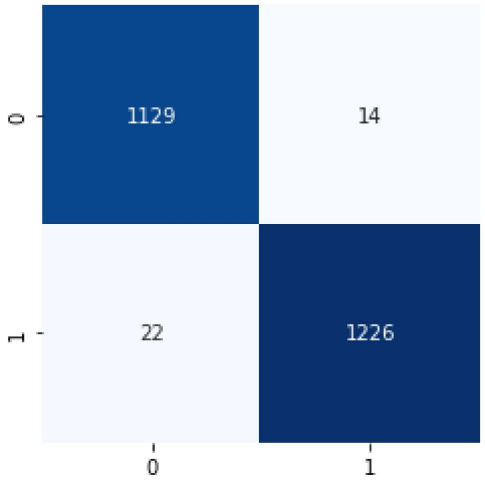


Figure 8.6 Confusion matrix for Naïve Bayes

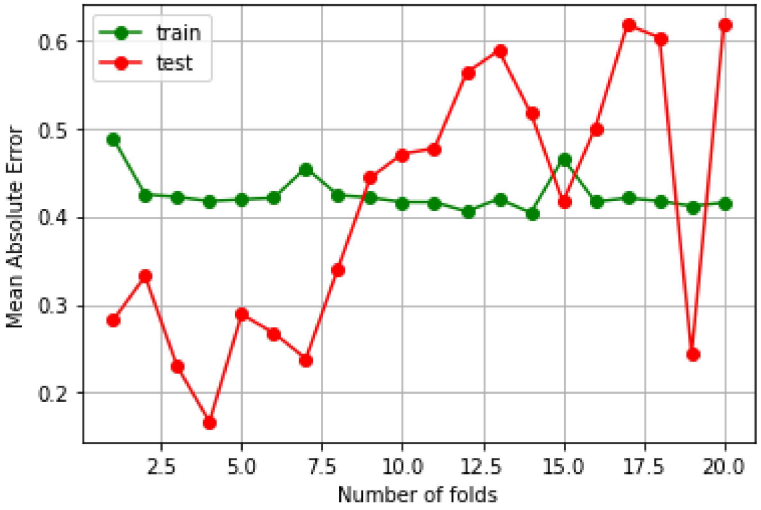


Figure 8.7 MAE vs No. Of folds for 20-fold cross validation for Naïve Bayes

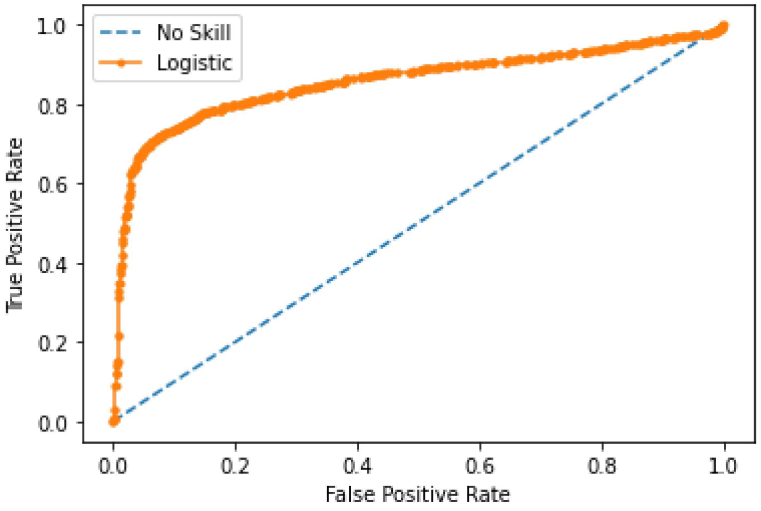


Figure 8.8 ROC Curve for Naive Bayes

- Random Forest

Accuracy Score : 0.9899623588456713

	precision	recall	f1-score	support
CANDIDATE	0.98	1.00	0.99	1143
FALSE POSITIVE	1.00	0.98	0.99	1248
accuracy			0.99	2391
macro avg	0.99	0.99	0.99	2391
weighted avg	0.99	0.99	0.99	2391

Figure 8.9 Accuracy and other scores for Random Forest

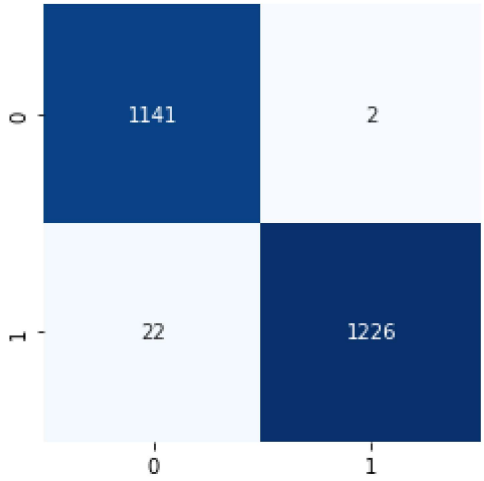


Figure 8.10 Confusion Matrix for Random Forest

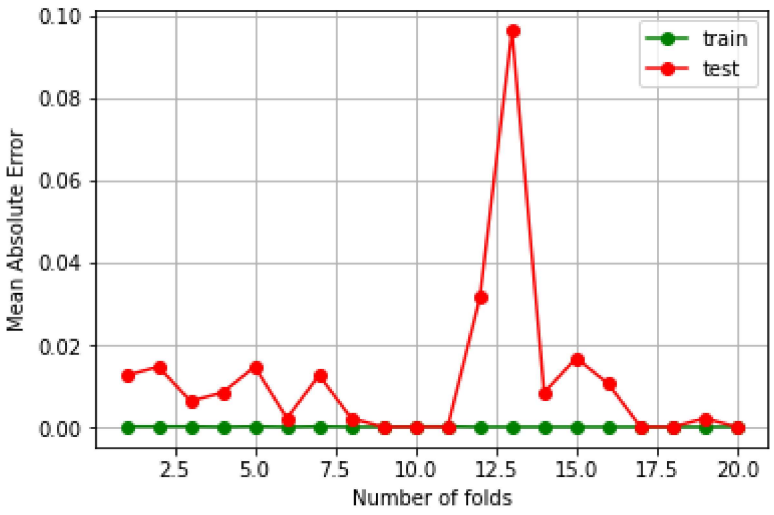


Figure 8.11 MAE vs No. Of folds for 20-fold cross validation for Random Forest

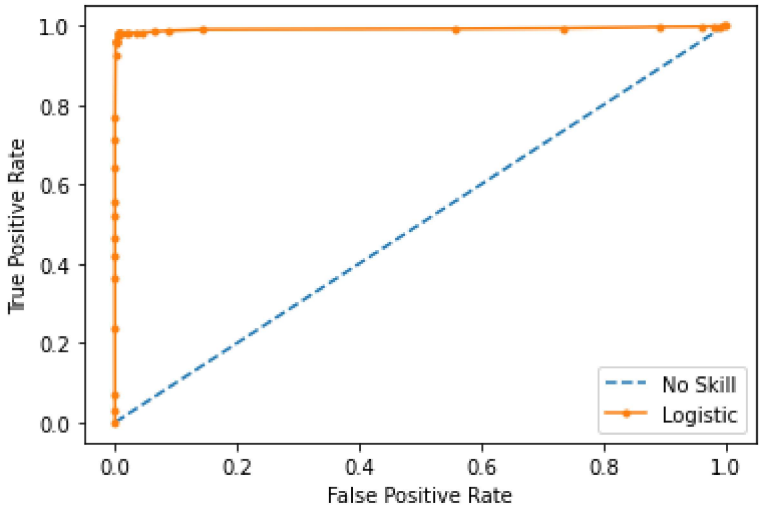


Figure 8.12 ROC Curve for Random Forest

- Decision Tree

Accuracy Score : 0.9824341279799247

	precision	recall	f1-score	support
CANDIDATE	0.98	0.98	0.98	1143
FALSE POSITIVE	0.98	0.98	0.98	1248
accuracy			0.98	2391
macro avg	0.98	0.98	0.98	2391
weighted avg	0.98	0.98	0.98	2391

Figure 8.13 Accuracy and other scores for Decision Tree

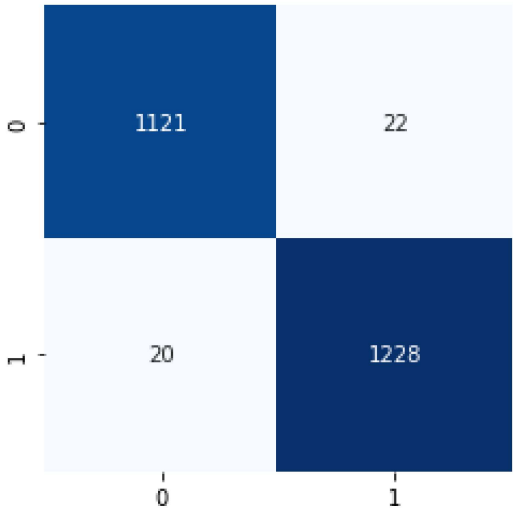


Figure 8.14 Confusion Matrix for Decision Tree

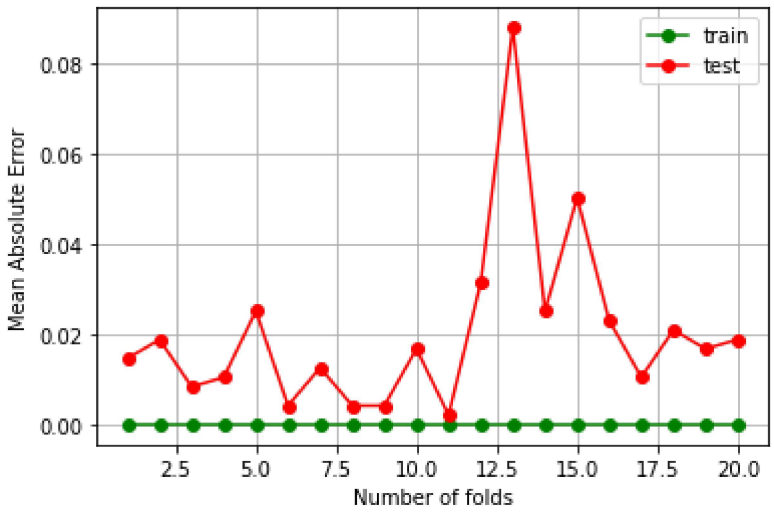


Figure 8.15 MAE vs No. Of folds for 20-fold cross validation

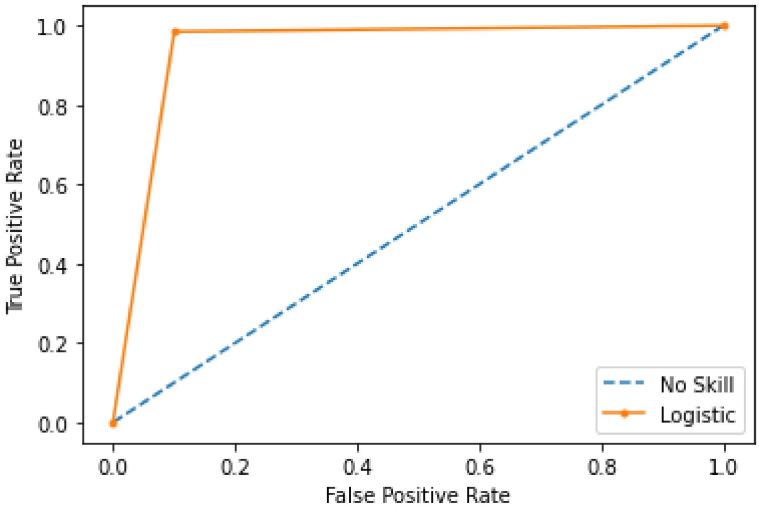


Figure 8.16 ROC Curve for Decision Tree

- XG Boost

Accuracy Score : 0.9891258887494772

	precision	recall	f1-score	support
CANDIDATE	0.98	1.00	0.99	1143
FALSE POSITIVE	1.00	0.98	0.99	1248
accuracy			0.99	2391
macro avg	0.99	0.99	0.99	2391
weighted avg	0.99	0.99	0.99	2391

Figure 8.17 Accuracy and other scores for XG Boost

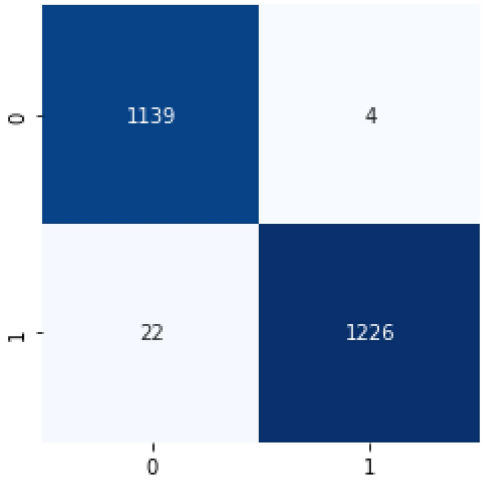


Figure 8.18 Confusion Matrix for XG Boost

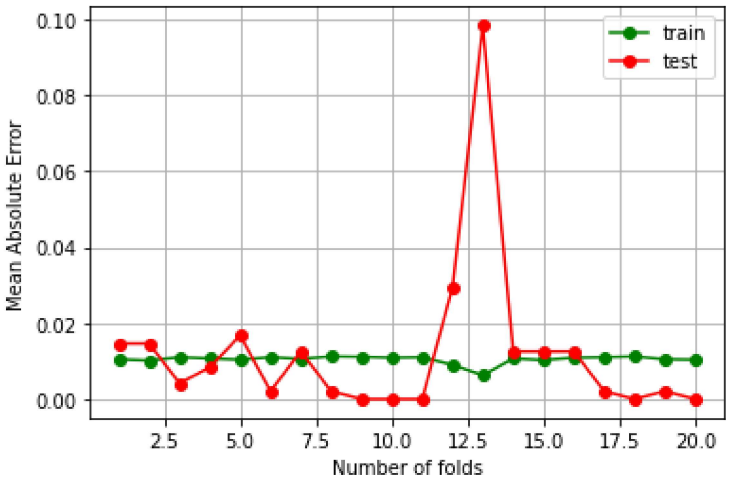


Figure 8.19 MAE vs No. Of folds for 20-fold cross validation

- Multi-Layered Perceptron
    - Loss after 30 epochs – 0.035
    - Accuracy – 0.96

Accuracy Score : 0.9759539989545217

	precision	recall	f1-score	support
CANDIDATE	0.99	0.96	0.98	935
FALSE POSITIVE	0.97	0.99	0.98	978
accuracy			0.98	1913
macro avg	0.98	0.98	0.98	1913
weighted avg	0.98	0.98	0.98	1913

Figure 8.20 Accuracy and other scores for Multi-layered perceptron

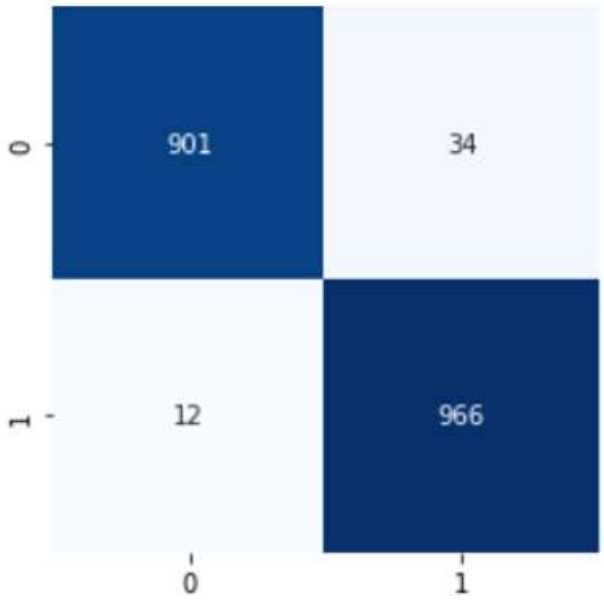


Figure 8.21 Confusion Matrix for Multi-layered perceptron

Table 8.1 Comparison of Results of all the existing and proposed models

	Models	Accuracy Score	Precision	recall	f1-score	ROC AUC
Existing Model as in[5]	SVM	0.9681	0.9309	0.973	0.9515	0.9694
	KNN	0.9371	0.854	0.9704	0.9085	0.9458
	Random Forest	0.9896	0.9955	0.9721	0.9837	0.985
Proposed Model	MLP	0.9749	0.97	0.97	0.97	-
	Logistic Regression	0.9870	0.98	0.99	0.99	0.978
	Random Forest	0.9899	0.99	0.99	0.99	0.992
	Decision Tree	0.9824	0.98	0.98	0.98	0.942
	Naive Bayes	0.9849	0.98	0.99	0.98	0.857
	XG Boost	0.9891	0.99	0.99	0.99	-

- Galaxy Morphology

After using the ResNet50 architecture without transfer learning the accuracy score obtained is 0.81 with a training loss and accuracy of 0.0088 and 0.7955 respectively and validation loss and accuracy of 0.0087 and 0.8100 respectively.

- Trained model gives a validation accuracy of 81%

```
Epoch 1/30
3771/3771 [==============================] - ETA: 0s - loss: 0.0454 - accuracy: 0.3113
Epoch 1: val_loss improved from inf to 0.02185, saving model to ../..\weights.hdf5
3771/3771 [==============================] - 2504s 662ms/step - loss: 0.0454 - accuracy: 0.3113 - val_loss: 0.0218 - va
l_accuracy: 0.6086
•
•
•
•
Epoch 30: val_loss improved from 0.00893 to 0.00869, saving model to ../..\weights.hdf5
3771/3771 [==============================] - 1008s 267ms/step - loss: 0.0088 - accuracy: 0.7955 - val_loss: 0.0087 - va
l_accuracy: 0.8100
```

Figure 8.22 Accuracy of ResNet50 model after training for 30 epochs

- The plot is done for validation loss and training loss vs number of epochs trained.

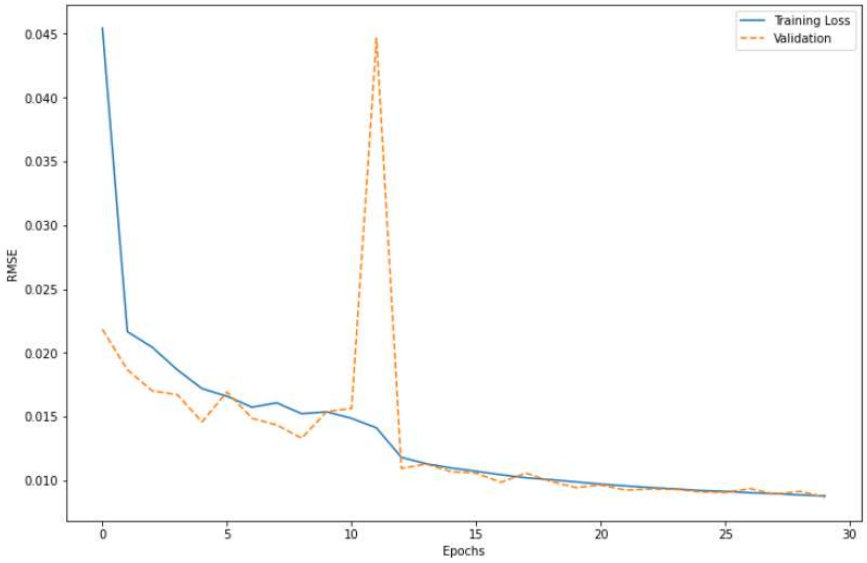


Figure 8.23 Graph of validation loss and training loss vs number of epochs trained.

Table 8.2 Comparison of Galaxy morphology methods tried. * Indicates the best accuracy.

ResNet50 Frozen Layer Galaxy Zoo 1	Validation_accuracy	Epochs
*Color images 37 classes	0.8100	30
Frozen Layer Hybrid model InceptionResnetV2		
Color images 11 classes	0.5379	39

- Model Deployment

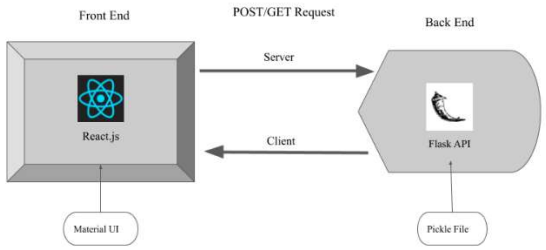


Figure 8.24 React and Flask Server Deployment model architecture

We construct a web application using React.js and the flask API to entirely automate the predictions. React is a JavaScript package that allows us to design reusable user interface components. We design a form with Material UI and use POST and GET requests to communicate the form data to the server. Material UI is a collection of components that may be used to design a wide range of user interfaces. We have used usestate and other React capabilities without needing to create a class with React hooks. Flask is a web framework written in Python for developing online applications. We have created a virtual environment in Python3.8 and installed all the necessary packages. The models that have been trained using exoplanet data will be compared, and the best model will be selected to make predictions for the online application. The server will make predictions and then return the results to the frontend for display.

Finally in order to deploy the application we set up the Gunicorn configuration. Application Deployment (also known as Software Deployment) is the process of installing, configuring, and enabling a specific application or collection of applications

to a certain URL on a server. The application(s) becomes publicly accessible on the URL once the deployment procedure is completed. Many developers use Gunicorn, a Python WSGI HTTP server, to deploy their Python applications. Because typical web servers don't know how to run Python programmes, this WSGI (Web Server Gateway Interface) is required. A WSGI allows you to deploy your Python programmes reliably, which is ideal for your needs. If your Python programme requires it, you can also set up many threads to serve it.

- Web Page

  The web page is developed using various web development tools like Flask, React and CSS. Our web page is a user-friendly website which could be used to classify galaxies and detect potential Exoplanets.

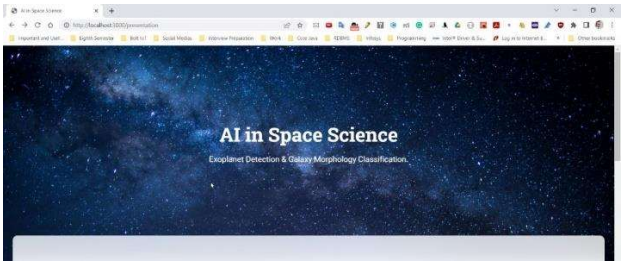


Figure 8.25 The opening screen of the main page

- The main page of the website provides the user to choose if the user wants to classify galaxies or detect potential exoplanets.

Figure 8.26 Option for user to select classify Galaxy morphology or Confirm an Exoplanet candidate.

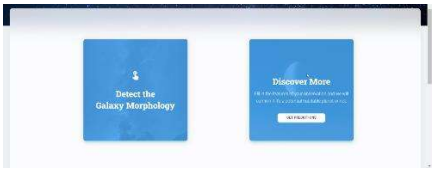


Figure 8.27 Option to get Predictions for Exoplanet

Figure 8.28 Option to classify Galaxy based on morphology

- The main page includes refer documentation which is a research article which includes thorough methodology for the models employed and accuracy of the models.

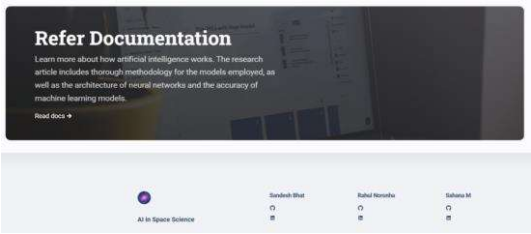


Figure 8.29 Documentation on main page of the website

- When we input the galaxy image and click submit, we get the class of the galaxy image as output

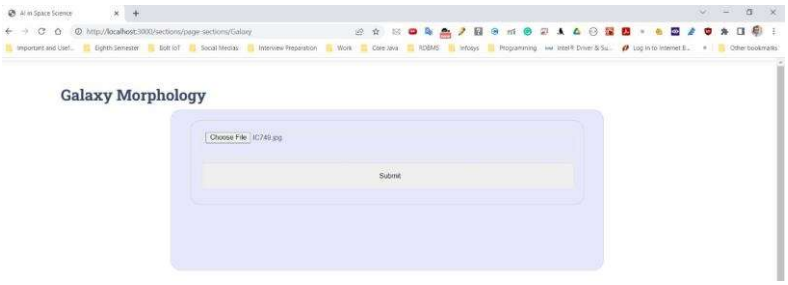


Figure 8.30 Input page for Galaxy images

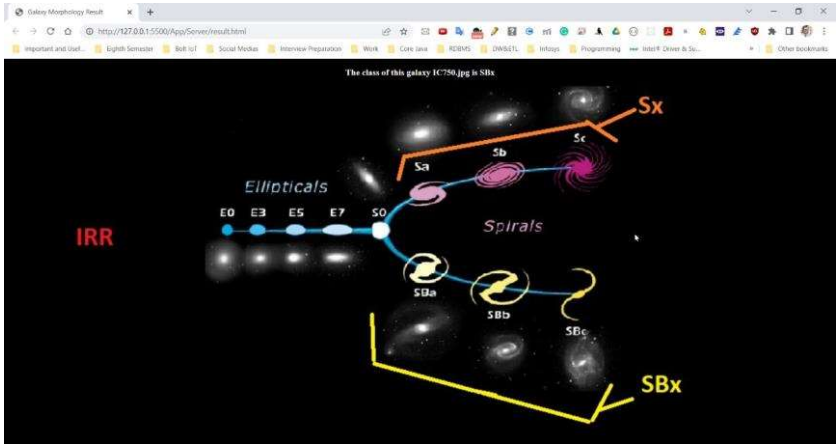


Figure 8.31 Result page for Galaxy image classifier

- When we pass the numerical parameters of the Exoplanet model via the frontend, we get the output stating if it is a habitable exoplanet or not.

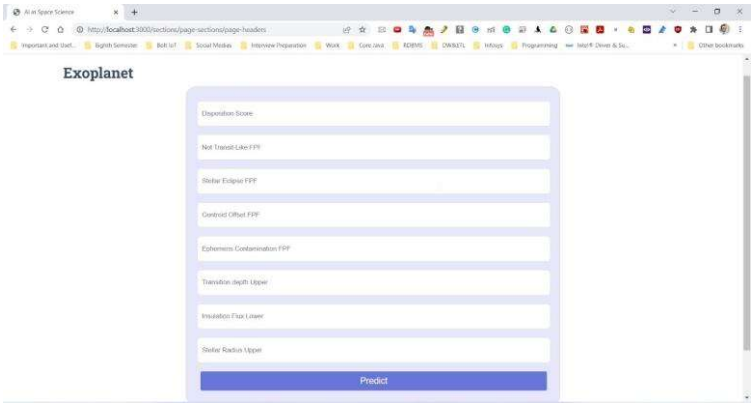


Figure 8.32 Input page for Exoplanet data

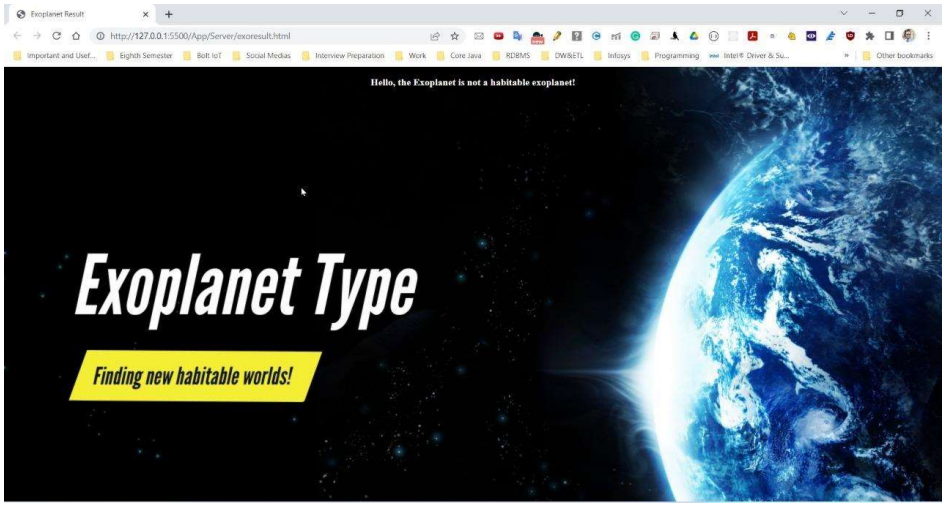


Figure 8.33 Result page for Exoplanet classifier

Chapter 9 CONCLUSIONS AND FUTURE WORK

The accuracy of the models can be improved by training for more number of epochs. We can try and build a custom architecture on top of the base ResNet50 model rather than combining it with other ImageNet models to try and improve the performance on this dataset.

Our supervised learning models successfully detect a potential exoplanet candidate with a greater accuracy than the existing work. Random Forest gives the maximum accuracy scores but Logistic regression performs the best for all the tests. When compared to other supervised machine learning models, the MLP does not always provide the best accuracy, but it does provide the least false negatives in the confusion matrix. The feature selection methods used provide the best subset from the original features. We have chosen the features that have the greatest impact on the predictions in order to improve the model's training efficiency. The authors tried the following approaches to feature selection. Quasi constant, Mutual Information Gain, Correlation coefficient, Forward Feature Selection. Once the best features were selected the performance of the different models were done. There is scope of using light curves to predict if an exoplanet is a potential exoplanet candidate. The transit method can be combined with the optimal features selected through the proposed methods to train a neural network as mentioned.

REFERENCES

[1] M. Z. Variawa, T. L. van Zyl and M. Woolway, "A rules-based and Transfer Learning approach for deriving the Hubble type of a galaxy from the Galaxy Zoo data," 2020 IEEE 23rd International Conference on Information Fusion (FUSION), 2020, pp. 1-7, doi: 10.23919/FUSION45008.2020.9190462.

[2] Sánchez et al. Transfer learning for galaxy morphology from one survey to another, Monthly Notices of the Royal Astronomical Society, Volume 484, Issue 1, March 2019, Pages 93–100, https://doi.org/10.1093/mnras/sty3497

[3] Ismael Araujo (2020). Using Machine Learning to Find Exoplanets with NASA's Data; https://towardsdatascience.com/using-machine-learning-to-find-exoplanets-with-nasas-dataset-bb818515e3b3

[4] L. Ofman, A. Averbuch, Adi Shliselberg, Idan Benaun, David Segev, Aron Rissman (2021). Automated identification of transiting exoplanet candidates in NASA Transiting Exoplanets Survey Satellite (TESS) data with machine learning methods, Physics, Computer Science, 2021, doi: 10.1016/j.newast.2021.101693

[5] George Clayton Sturrock, Brychan Manry, Sohail Rafiqi : Machine Learning Pipeline for Exoplanet Classification, 2019

APPENDIX A

REDSHIFT AND BLUESHIFT

In physics, a redshift is an increase in the wavelength, and corresponding decrease in the frequency and photon energy, of electromagnetic radiation. The opposite change, a decrease in wavelength and simultaneous increase in frequency and energy, is known as a negative redshift, or blueshift.

APPENDIX B

MORPHOLOGY

Galaxy morphological classification is a system used by astronomers to divide galaxies into groups based on their visual appearance. There are several schemes in use by which galaxies can be classified according to their morphologies, the most famous being the Hubble sequence, devised by Edwin Hubble and later expanded by Gérard de Vaucouleurs and Allan Sandage. However, galaxy classification and morphology are now largely done using computational methods and physical morphology.

APPENDIX C

HUBBLE TYPE

Edwin Hubble invented a classification of galaxies and grouped them into four classes: spirals, barred spirals, ellipticals and irregulars. He classified spiral and barred spiral galaxies further according to the size of their central bulge and the texture of their arms.

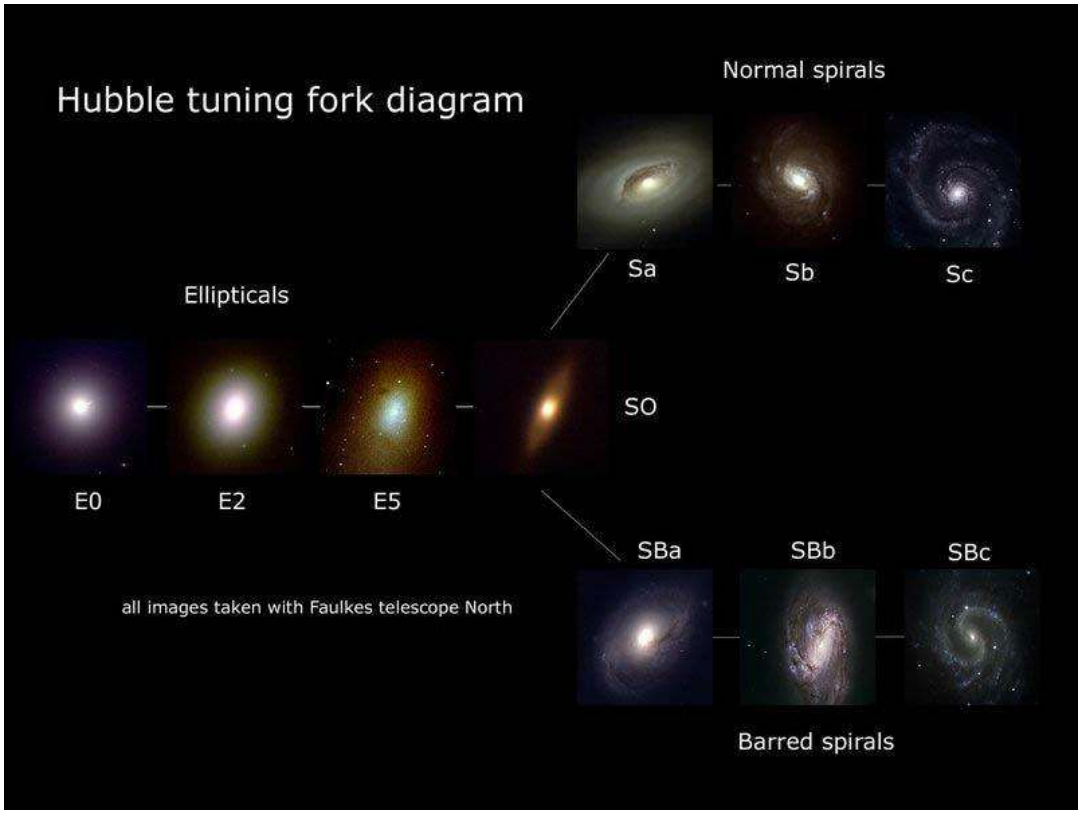


Figure C.1 Hubble Type Classification of Galaxies proposed by Edwin Hubble

APPENDIX D

EXOPLANET

An exoplanet is any planet beyond our solar system. Most orbit other stars, but free-floating exoplanets, called rogue planets, orbit the galactic center and are untethered to any star.

APPENDIX E

RESNET50

ResNet50 is a variant of the ResNet model which has 48 Convolution layers along with 1 MaxPool and 1 Average Pool layer. It has 3.8 x 10^9 Floating points operations. This architecture can be used on computer vision tasks such as image classification, object localisation, object detection.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure E.1 ResNet50 Architecture

CODE

The code for the Machine learning and transfer learning training can be found here

https://github.com/GalaxyMorphologyAndExoplanetDetection/ML_Modelling

The code for the Flask server and React frontend can be found here

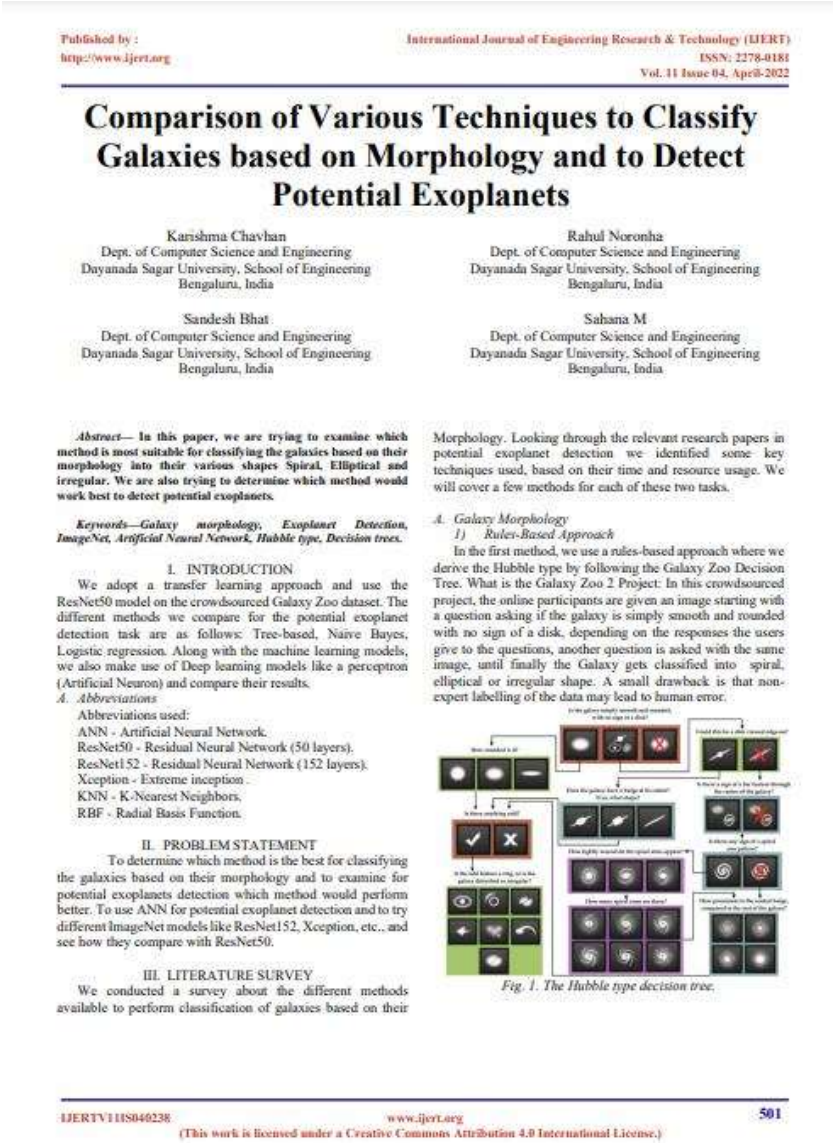
https://github.com/GalaxyMorphologyAndExoplanetDetection/Exoplanets-and-Galaxy-Morphology-classification-using-ML-main

FUNDING AND PUBLISHED PAPER DETAILS

Analysis paper published in International Journal of Engineering Research and Technology (IJERT).

Rahul Noronha , Sahana M , Sandesh Bhat , Karishma Chavhan, 2022, Comparison of Various Techniques to Classify Galaxies based on Morphology and to Detect Potential Exoplanets, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 11, Issue 04 (April 2022)

https://www.ijert.org/comparison-of-various-techniques-to-classify-galaxies-based-on-morphology-and-to-detect-potential-exoplanets



Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 04, April-2022

Comparison of Various Techniques to Classify Galaxies based on Morphology and to Detect Potential Exoplanets

Karishma Chavhan
Dept. of Computer Science and Engineering
Dayanada Sagar University, School of Engineering
Bengaluru, India

Rahul Noronha
Dept. of Computer Science and Engineering
Dayanada Sagar University, School of Engineering
Bengaluru, India

Sandesh Bhat
Dept. of Computer Science and Engineering
Dayanada Sagar University, School of Engineering
Bengaluru, India

Sahana M
Dept. of Computer Science and Engineering
Dayanada Sagar University, School of Engineering
Bengaluru, India

***Abstract*— In this paper, we are trying to examine which method is most suitable for classifying the galaxies based on their morphology into their various shapes Spiral, Elliptical and irregular. We are also trying to determine which method would work best to detect potential exoplanets.**

Keywords—Galaxy morphology, Exoplanet Detection, ImageNet, Artificial Neural Network, Hubble type, Decision trees.

I. INTRODUCTION

We adopt a transfer learning approach and use the ResNet50 model on the crowdsourced Galaxy Zoo dataset. The different methods we compare for the potential exoplanet detection task are as follows: Tree-based, Naive Bayes, Logistic regression. Along with the machine learning models, we also make use of Deep learning models like a perceptron (Artificial Neuron) and compare their results.

A. Abbreviations

Abbreviations used:
ANN - Artificial Neural Network.
ResNet50 - Residual Neural Network (50 layers).
ResNet152 - Residual Neural Network (152 layers).
Xception - Extreme inception .
KNN - K-Nearest Neighbors.
RBF - Radial Basis Function.

II. PROBLEM STATEMENT

To determine which method is the best for classifying the galaxies based on their morphology and to examine for potential exoplanets detection which method would perform better. To use ANN for potential exoplanet detection and to try different ImageNet models like ResNet152, Xception, etc., and see how they compare with ResNet50.

III. LITERATURE SURVEY

We conducted a survey about the different methods available to perform classification of galaxies based on their Morphology. Looking through the relevant research papers in potential exoplanet detection we identified some key techniques used, based on their time and resource usage. We will cover a few methods for each of these two tasks.

A. Galaxy Morphology

1) Rules-Based Approach

In the first method, we use a rules-based approach where we derive the Hubble type by following the Galaxy Zoo Decision Tree. What is the Galaxy Zoo 2 Project: In this crowdsourced project, the online participants are given an image starting with a question asking if the galaxy is simply smooth and rounded with no sign of a disk, depending on the responses the users give to the questions, another question is asked with the same image, until finally the Galaxy gets classified into spiral, elliptical or irregular shape. A small drawback is that non-expert labelling of the data may lead to human error.

Fig. 1. The Hubble type decision tree.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)