Project Report

Shivaansh Bhatia
Shivaans
1722599

1. Introduction to Exoplanets and the Dataset

Exoplanets:

Exoplanets, or **extrasolar planets**, are planets that orbit stars outside our solar system. Their discovery has revolutionized our understanding of planetary systems and the potential for life beyond Earth. Since the first confirmed detection in the 1990s, thousands of exoplanets have been identified, exhibiting a diverse range of sizes, compositions, and orbital characteristics. Understanding exoplanets helps scientists answer fundamental questions about planetary formation, habitability, and the uniqueness of our solar system.

The **Kepler Space Telescope**, launched by NASA in 2009, was designed to identify Earth-sized planets in the habitable zone of their stars using the **transit method**. This method detects planets by measuring slight dips in the brightness of a star caused when a planet passes in front of it. The mission collected vast amounts of data, revealing thousands of potential exoplanets and providing a rich dataset for further study.

The Dataset: Kepler Exoplanet Search Results

The dataset used in this project is extracted from the **NASA Exoplanet Archive** available on kaggle , This dataset contains observations and characteristics of **potential exoplanets** identified during the Kepler mission. The primary goal of this dataset is to distinguish between:

- **Confirmed Exoplanets**: Verified planets orbiting stars.
- **Candidates**: Planetary candidates awaiting confirmation.
- **False Positives**: Observations that, after analysis, were determined not to be planets

Objective

The objective of this data is to predict whether a given candidate,, is a **confirmed exoplanet** or a **false positive**. This is framed as a **binary classification problem**, where the **positive class** consists of confirmed exoplanets and candidates (1), while false positives are treated as the **negative class**.(0)

Input

The input consists of observational and physical characteristics (features) of planetary candidates detected by the Kepler mission. Some important features include orbital, stellar, and transit-related measurements with a bunch of columns for errors in total 43 columns.Few important features are:

Feature	Description
`koi_period`	Orbital period of the exoplanet candidate (in days).
`koi_time0bk`	Time of the first detected transit (in days, relative scale).
`koi_duration`	Duration of the observed transit (in hours).
`koi_depth`	Transit depth (the percentage of light blocked by the candidate).
`koi_prad`	Radius of the candidate planet (in Earth radii).
`koi_steff`	Effective temperature of the host star (in Kelvin).
`koi_srad`	Stellar radius of the host star (in Solar radii).

koi_slogg	Surface gravity of the host star (in log scale).

Output

The target labels were constructed by combining the following categories from the koi_disposition column.

The output is a binary classification label:

- **1**: Represents planetary candidates that are **CONFIRMED**.
- **0** : Represents planetary candidates labeled as **FALSE POSITIVE**.

Data Cleaning

Unnecessary columns such as kepler_name,kepler_id, kepoi_name etc. were removed from the dataset using **datascience,numpy and pandas library.**

Removing the third class namely CANDIDATES was necessary as these were the planets that were potential exoplanets but not confirmed and were creating class imbalance in the analysis.

Half of the FALSE POSITIVE planets were removed to create class balance, as there were double FALSE POSITIVES than the CONFIRMED.

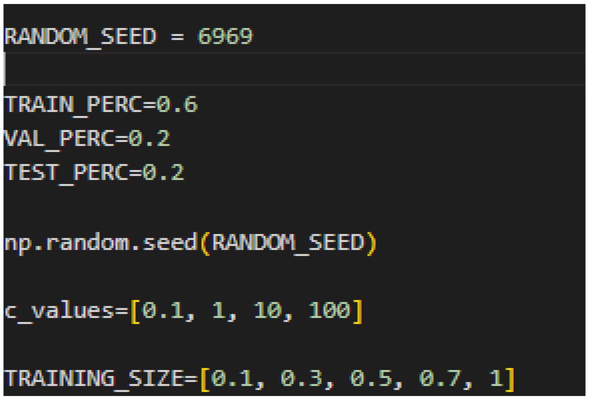
There are total 4451 samples with 40 features

MODELS

Hyper parameters:

We will test which c_ value is the best for svm model, this can be done by the inbuilt grid search method which gives the hyperparameter which fit the data the best

We will also use different size of training set to test how accurate the models train



```
RANDOM_SEED = 6969

TRAIN_PERC=0.6
VAL_PERC=0.2
TEST_PERC=0.2

np.random.seed(RANDOM_SEED)

c_values=[0.1, 1, 10, 100]

TRAINING_SIZE=[0.1, 0.3, 0.5, 0.7, 1]
```

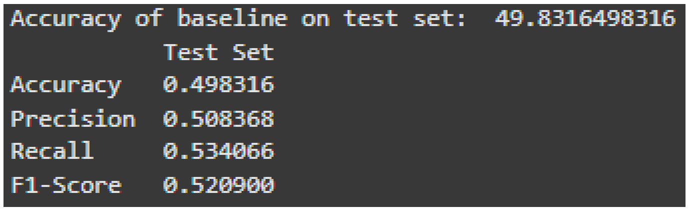
Four Models are compared in this project namely 1/k random classifier, Logistic regression, SVM and MLP.

Baseline model

A baseline model serves as a simple, reference model against which the performance of more complex machine learning models can be compared. Its purpose is to establish a **minimum performance threshold** that other models must exceed to demonstrate their effectiveness.

In this project, the baseline model is implemented as a 1/k **random classifier** that predicts the target classes (`0` or `1`) based on the class distribution in the dataset. (k= number of unique classes, in this case 2, so expected output should be %)

Report:



```
Accuracy of baseline on test set:  49.8316498316
           Test Set
Accuracy   0.498316
Precision  0.508368
Recall     0.534066
F1-Score   0.520900
```

As expected the baseline model has an accuracy of ~50% on the test samples and acts as the minimum threshold for other models.

Logistic regression

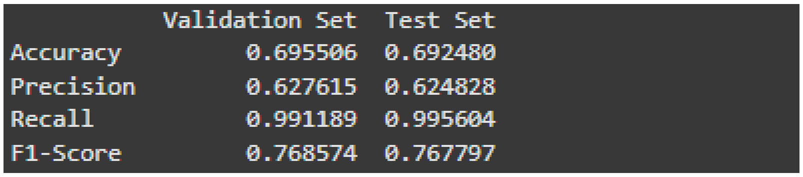
Logistic Regression is a **linear classification algorithm** that predicts the probability of a sample belonging to a specific class.. In this project, it is used to classify planetary candidates as either:

- **1 → CONFIRMED**
- **0 → FALSE POSITIVE**

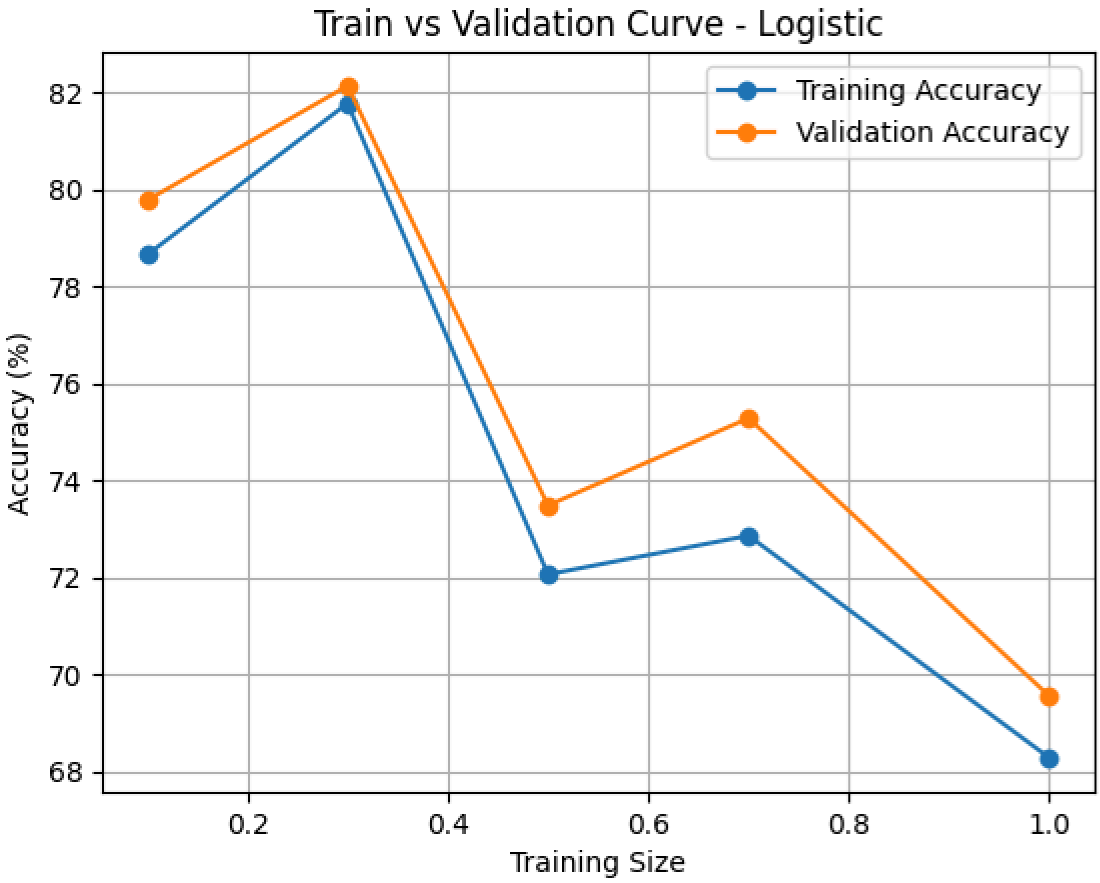
Logistic Regression models the probability of an observation belonging to the positive class using the **sigmoid function**, which ensures the output probability lies between **0 and 1**.

Report :

Accuracy on full test set - **69.2480359147**



```
           Validation Set  Test Set
Accuracy         0.695506  0.692480
Precision        0.627615  0.624828
Recall           0.991189  0.995604
F1-Score         0.768574  0.767797
```



We can see the training and validation accuracy becomes less over proportions of training set

Suggesting the decrease in accuracy of the model over time

SVM

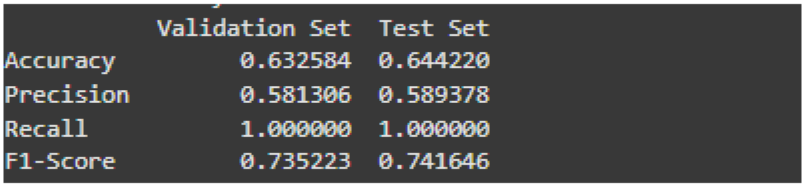
Support Vector Machine (SVM) is a **supervised machine learning algorithm** primarily used for **classification tasks**. SVM works by finding the **optimal hyperplane** that best separates the classes in the feature space. It is particularly powerful for binary classification problems and can be extended to non-linear problems using **kernels**. In this project the kernel = ‘RBF’

In this project, SVM is used to classify planetary candidates as either:

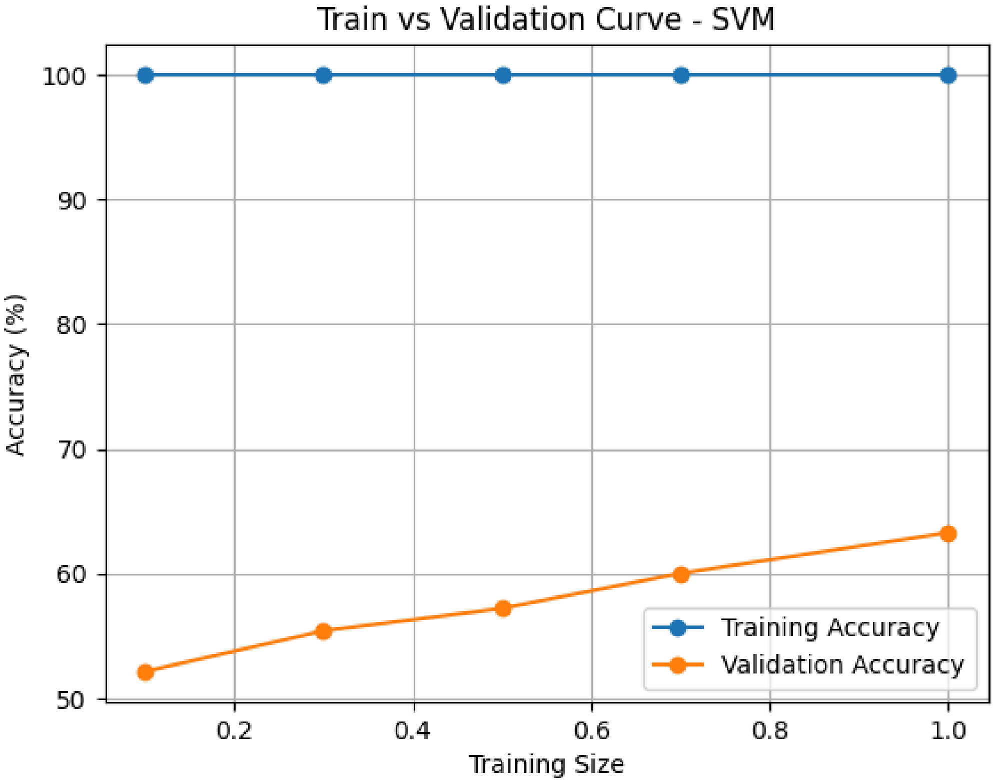
- **1 → CONFIRMED**
- **0 → FALSE POSITIVE**

Report :

With class_weights=’balanced’



```
           Validation Set  Test Set
Accuracy         0.632584  0.644220
Precision        0.581306  0.589378
Recall           1.000000  1.000000
F1-Score         0.735223  0.741646
```



We can clearly see from the graph above that the training accuracy comes out to be 100%, which is not possible which could suggest data set shortcomings such as not enough data , or overlapping features, and could also suggest overfitting.

But we can see the accuracy of validation increases with increased proportion of the dataset.

In conclusion, more the data and less overlapping features , the more accurate SVM will be.

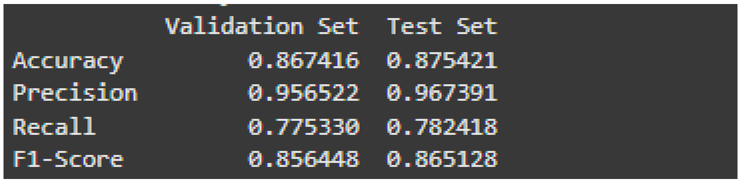
MLP

The Multi-Layer Perceptron (MLP) is a type of **artificial neural network** designed for supervised learning tasks like classification and regression. MLP consists of an **input layer**, one or more **hidden layers**, and an **output layer**. Each layer contains multiple neurons, which process the input data and pass it through activation functions to learn complex relationships in the data.

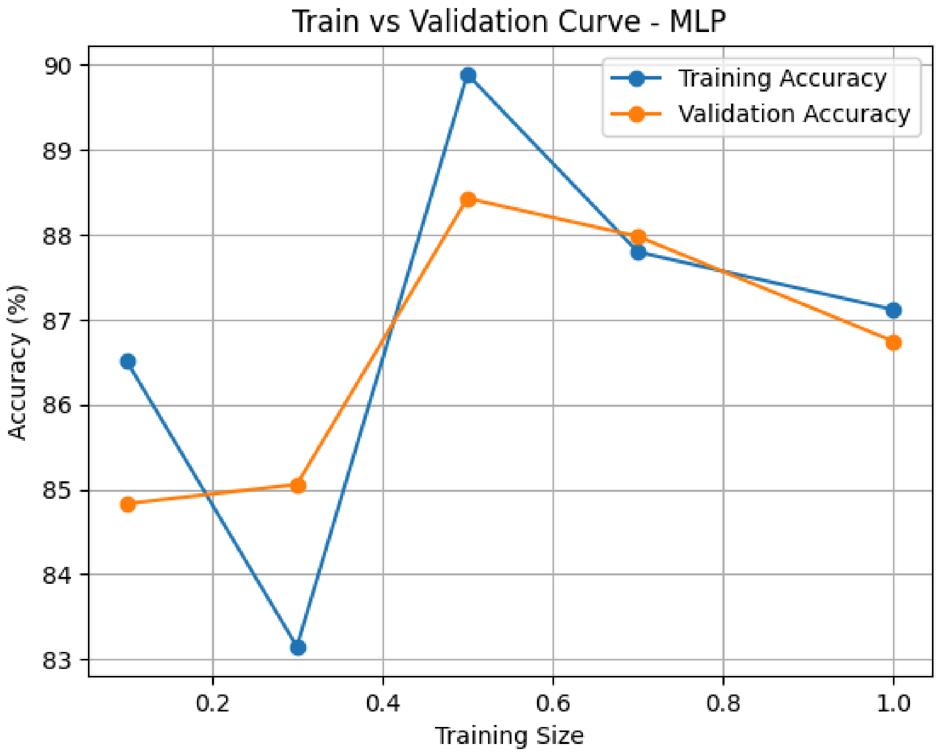
In this project, MLP is used to classify planetary candidates into two classes:

- **1 → CONFIRMED** .
- **0 → FALSE POSITIVE**.

Report :



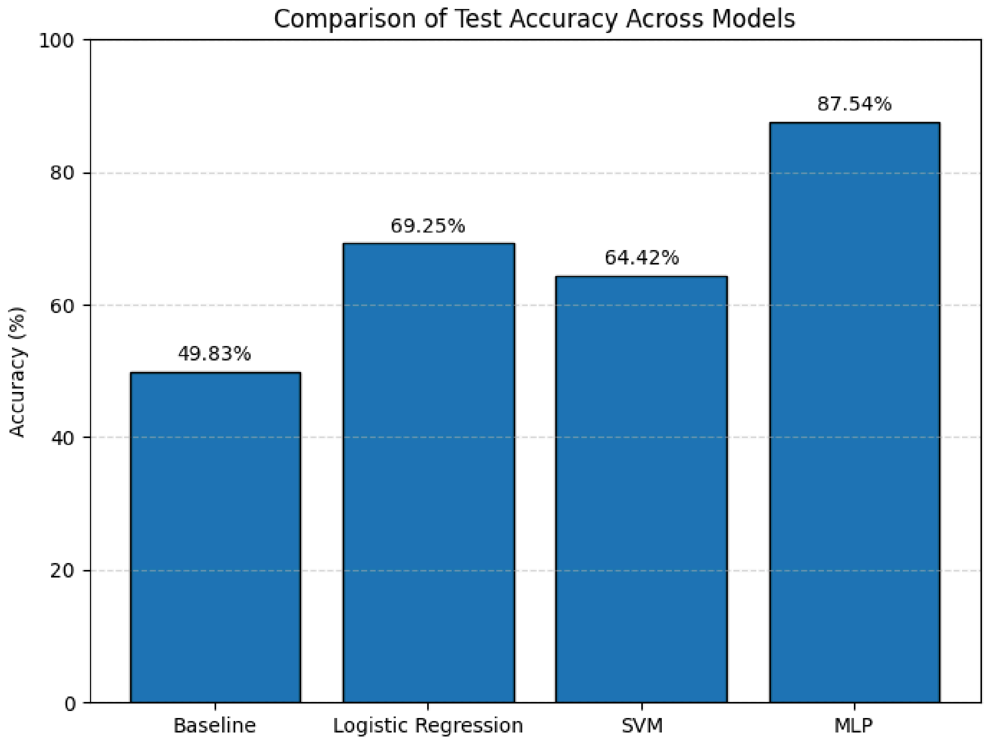
```
           Validation Set  Test Set
Accuracy         0.867416  0.875421
Precision        0.956522  0.967391
Recall           0.775330  0.782418
F1-Score         0.856448  0.865128
```



We can see that the gap between the training set and the validation set is decreasing over time, suggesting higher accuracy with more samples provided .

Conclusion

TEST ACCURACY



We can see from the graph above that all the algorithms performed better than the baseline algorithm

MLP > Logistic > SVM

Anomaly:

We observe that the accuracy of Logistic regression is better than the accuracy of SVM.
This could be due to various reasons.

- Linearity of the data: We use the rbf model of the SVM, which underperformed (linear model was taking way too long)
-
- Another reason could be, not enough sample for SVM
-
- While logistic regression performed better than SVM this could be due to overfitting as well