University School of Automation and Robotics

Machine Learning Project Report

Classification on SUSY Dataset

Professor

Dr. Amit Choudhary

Priyanshu Singh

00719011621

B-tech in AI and ML

B1 A

BASIC INFORMATION

Title of Project: SUSY dataset

Student Name: Priyanshu singh

Branch : Artificial Intelligence & Machine Learning B1A

Enrolment Number: 0071901121

Email ID: priyanshu.00719011621@ipu.ac.in

Contact Number: 7303940384

Google Drive Link:

https://drive.google.com/drive/folders/1XqlbiNam8BlDhHenDV8ANc62rAby8b21?usp=sharing

YouTube Video Link:

https://youtu.be/v6ahKV4-1Y4

GitHub Link:

https://github.com/27priyanshu/ML-project/blob/main/Copy_of_SUSY.ipynb

SuperSymmetric Particles

Abstract:

The SUSY (SuperSymmetry) dataset available on the UCI Machine Learning Repository is a collection of simulated particle collision events designed for high-energy physics research. It was generated using Monte Carlo simulations to mimic the behavior of particle interactions at the Large Hadron Collider (LHC) experiment. The dataset comprises a total of 5.8 million instances, each described by 18 attributes.

The attributes in the SUSY dataset include eight low-level features derived from the raw detector measurements, eight high-level features obtained from the low-level features using feature engineering techniques, a binary class label indicating the presence or absence of a supersymmetric particle, and a weight attribute for statistical purposes.

The goal of using this dataset is to develop predictive models or perform data analysis tasks to uncover patterns, relationships, or anomalies in supersymmetric particle collisions. Researchers and machine learning practitioners can utilize this dataset to explore various techniques, such as binary classification to predict the presence of supersymmetric particles, exploratory data analysis to understand the data distribution, dimensionality reduction to visualize high-dimensional data, or anomaly detection to identify unusual events.

By analyzing the SUSY dataset, researchers and physicists can gain insights into the fundamental properties of particles and potentially discover new phenomena in the realm of particle physics. This dataset serves as a valuable resource for the scientific community interested in particle physics research and machine learning applications in this field.

Keywords:

Random Forest

Logistic Regression

Naïve bayes

K Neighbors Classifier

SVM

1. Introduction:

The SUSY (SuperSymmetry) dataset is a collection of simulated particle collision events used in high-energy physics research. It was generated through Monte Carlo simulations to emulate interactions occurring at the Large Hadron Collider (LHC), one of the world's most powerful particle accelerators.

The dataset consists of approximately 5.8 million instances, with each instance described by 18 attributes. These attributes include eight low-level features derived from raw detector measurements, eight high-level features obtained through feature engineering techniques, a binary class label indicating the presence or absence of a supersymmetric particle, and a weight attribute used for statistical purposes.

The primary purpose of the SUSY dataset is to enable the development and evaluation of machine learning models and data analysis techniques related to supersymmetric particle collisions. Researchers can leverage this dataset to perform tasks such as binary classification to predict the presence of supersymmetric particles, exploratory data analysis to uncover patterns and relationships within the data, dimensionality reduction for visualization purposes, and anomaly detection to identify unusual events or outliers.

By analyzing the SUSY dataset, scientists and machine learning practitioners can gain a deeper understanding of particle physics and potentially make discoveries related to the fundamental properties of particles. The dataset serves as a valuable resource for those interested in conducting research and applying machine learning techniques within the field of high-energy physics.

Random Forest:

Random Forest is an ensemble learning algorithm that combines multiple decision trees to make predictions. Each tree in the forest is trained on a random subset of the data with replacement. During prediction, the final output is determined by aggregating the predictions of individual trees through voting or averaging. Random Forests are robust against overfitting, can handle high-dimensional data, and provide measures of feature importance.

Logistic Regression:

Logistic Regression is a popular statistical model used for binary classification. It models the relationship between the input features and the probability of belonging to a certain class using the logistic function. Logistic Regression assumes a linear relationship between the features and the log-odds of the outcome. It estimates the model parameters through maximum likelihood estimation and makes predictions based on a specified threshold.

Naive Bayes:

Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem. It assumes that all features are conditionally independent of each other given the class label. Despite its simplicity and the "naive" assumption, Naive Bayes can perform well in many real-world scenarios. It calculates the probability of each class given the input features and assigns the class with the highest probability as the prediction.

K-Nearest Neighbors (KNN) Classifier:

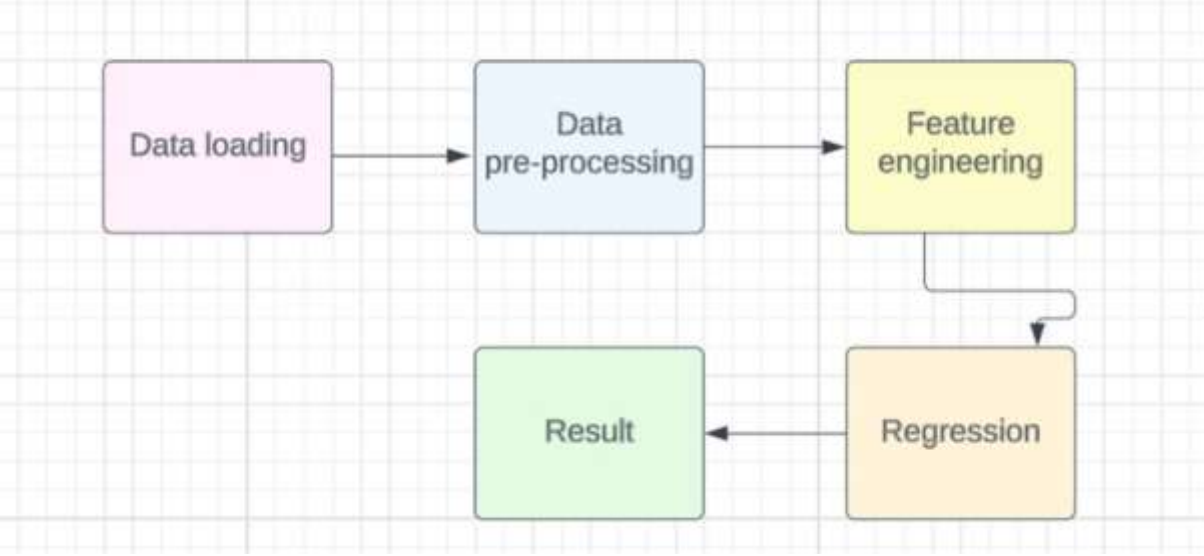
K-Nearest Neighbors is a non-parametric classification algorithm that makes predictions based on the similarity between input samples. It works by finding the K nearest neighbors (based on a distance metric) in the training data and assigning the majority class among those neighbors as the prediction. KNN does not involve explicit model training and can handle multi-class classification problems. It requires choosing an appropriate value of K and a suitable distance metric.

Support Vector Machines (SVM):

Support Vector Machines is a powerful supervised learning algorithm used for classification and regression tasks. SVM aims to find an optimal hyperplane that maximally separates the data points of different classes. It can handle linear and non-linear classification problems through the use of kernel functions. SVM finds the optimal hyperplane by maximizing the margin or by incorporating additional penalties for misclassifications. SVM is effective in high-dimensional spaces and can handle datasets with complex decision boundaries.

2. Proposed Methodology [Pictorial Diagram and Explanation of each sub modules]

Diagram / Flow Chart



a. Datasets

The first column is the class label (1 for signal, 0 for background), followed by the 18 features (8 low-level features then 10 high-level features):: lepton 1 pT, lepton 1 eta, lepton 1 phi, lepton 2 pT, lepton 2 eta, lepton 2 phi, missing energy magnitude, missing energy phi, MET_rel, axial MET, M_R, M_TR_2, R, MT2, S_R, M_Delta_R, dPhi_r_b, cos(theta_r1).

b. Pre processing

All features are float values. There are no NULL values in the rows.

Shape of data frame = `(5000000, 19)`

c. Feature Scaling

For desirable model training we have to remove features which are not in use.

But in this model we did not apply Feature Scaling

d. Train Test Split

We divide the data in Training and Testing data so that we can check the score and test our data.

e. Model Training and Testing with the help of Classification

We apply model random forest, logistic regression, svm, Knn and naïve bayes on the dataset.

f. Performance Measure

There are various Performance metrics that can help to measure performance using Accuracy Score.

3. Result & Discussion

Then we apply the various Classification models after splitting the dataset into training and testing data -

RandomForest

LogisticRegression

naive_bayes

KNeighborsClassifier

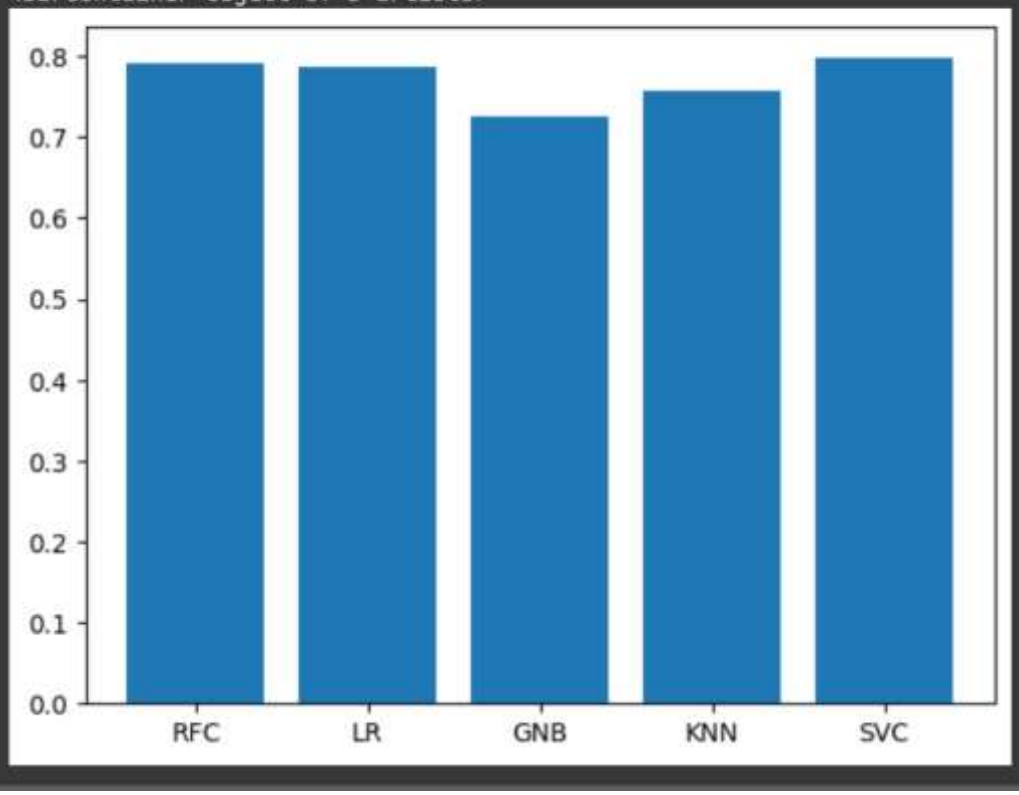
SVM

After applying the model we calculate the scores.

```
RFC :- 0.79155

LR :- 0.78585

GNB :- 0.72565

KNN :- 0.75655

SVC :- 0.7973
```

```
import matplotlib.pyplot as plt
plt.bar(acc.keys(), acc.values())
```

```
<BarContainer object of 5 artists>
```

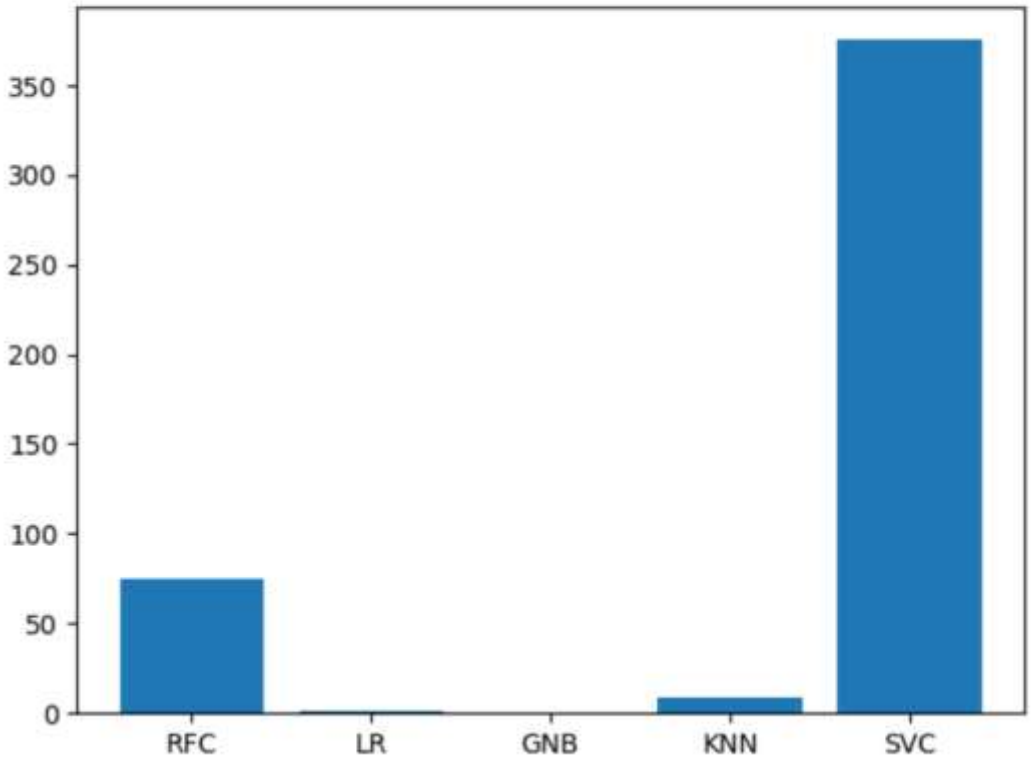


We can see the best accuracy is shown by SVM(Support Vector Machine) is `0.7973` on our data.

4. **Conclusion**

From the SUSY Project we could conclude that the Best Model that fits the data is **Support Vector Machine and** It's the most time taken model to execute**.**

```
13]: plt.bar(t.keys(), t.values())

13]: <BarContainer object of 5 artists>
```



References

1. https://journals.aps.org/prd/abstract/10.1103/PhysRevD.49.4908
2. https://www.sciencedirect.com/science/article/abs/pii/055032138890171X
3. https://arxiv.org/abs/hep-ph/0609292
4. https://pubs.aip.org/aip/jmp/article-abstract/21/7/1863/451042/Classical-supersymmetric-particles?redirectedFrom=fulltext
5. https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.83.1731