📘 APR Mini Project Report

Support Vector Machine (SVM) and Support Vector Regression (SVR)

Course: APR
Project Type: Mini Project

🧩 1. Introduction

In this mini-project, we implemented and evaluated **Support Vector Machine (SVM)** for classification and **Support Vector Regression (SVR)** for regression tasks using the **Pulsar Star dataset**.
The goal was to identify pulsar candidates based on their radio emission characteristics and explore how Support Vector-based algorithms perform on real-world astronomical data.

🎯 2. Objective

- To apply **SVM** for binary classification of pulsar candidates.
- To apply **SVR** for regression analysis on the same dataset.
- To analyze and compare model performance using metrics and visualizations.
- To understand the effect of data preprocessing on model performance.

📂 3. Dataset Description

Two separate files were used for this project:

- pulsar_data_train.csv — Training dataset
- pulsar_data_test.csv — Testing dataset

Each dataset contains multiple features representing statistical properties of radio emissions.

Column Type	Description
Numerical Features	Mean, Standard Deviation, Skewness, Kurtosis, etc.
Target	target_class → 1 for pulsar star, 0 for non-pulsar

⚙️ 4. Methodology

4.1 Data Preprocessing

- Loaded the datasets using **Pandas**.
- Handled **missing values** by imputing feature columns with their mean.
- Dropped rows containing missing target_class values.
- Standardized the features using **StandardScaler** to improve model performance.

4.2 Handling Missing and Empty Data

To ensure robustness:

- If any dataset became empty after cleaning, a **train-test split** was performed from the available training data.

4.3 Model Development

Two models were developed:

1. **Support Vector Machine (SVM)** with RBF kernel for classification.
2. **Support Vector Regression (SVR)** with RBF kernel for regression.

4.4 Model Training

```
svm_clf = SVC(kernel='rbf', C=1.0, gamma='scale', random_state=42)
svm_clf.fit(X_train_scaled, y_train)

svr_reg = SVR(kernel='rbf', C=1.0, gamma='scale')
svr_reg.fit(X_train_scaled, y_train)
```

📊 5. Model Evaluation

5.1 SVM Classification Results

The SVM classifier was evaluated using:

- **Accuracy**
- **Confusion Matrix**
- **Classification Report (Precision, Recall, F1-score)**

```
y_pred = svm_clf.predict(X_test_scaled)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

Visualization:

```
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap='Blues')
plt.title("SVM Confusion Matrix")
```

5.2 SVR Regression Results

For regression, we used:

- **Mean Squared Error (MSE)**
- **R^2 Score**

```
y_pred_reg = svr_reg.predict(X_test_scaled)
mse = mean_squared_error(y_test, y_pred_reg)
r2 = r2_score(y_test, y_pred_reg)
```

Plot:

```
plt.scatter(y_test, y_pred_reg, alpha=0.6)
plt.xlabel("Actual")
plt.ylabel("Predicted")
plt.title("SVR Regression: Actual vs Predicted")
```

Comments

- **Our original model test accuracy is 0.9832 while GridSearch CV score on test-set is 0.9835.**
- **So, GridSearch CV helps to identify the parameters that will improve the performance for this particular model.**

- **Here, we should not confuse best_score_ attribute of grid_search with the score method on the test-set.**
- **The score method on the test-set gives the generalization performance of the model. Using the score method, we employ a model trained on the whole training set.**
- **The best_score_ attribute gives the mean cross-validation accuracy, with cross-validation performed on the training set.**

🧠 **Results and conclusion**

1. **There are outliers in our dataset. So, as We increase the value of C to limit fewer outliers, the accuracy increased. This is true with different kinds of kernels.**
2. **We get maximum accuracy with rbf and linear kernel with C=100.0 and the accuracy is 0.9832. So, we can conclude that our model is doing a very good job in terms of predicting the class labels. But, this is not true. Here, we have an imbalanced dataset. Accuracy is an inadequate measure for quantifying predictive performance in the imbalanced dataset problem. So, we must explore confusion matrix that provide better guidance in selecting models.**
3. **ROC AUC of our model is very close to 1. So, we can conclude that our classifier does a good job in classifying the pulsar star.**
4. **We obtain higher average stratified k-fold cross-validation score of 0.9789 with linear kernel but the model accuracy is 0.9832. So, stratified cross-validation technique does not help to improve the model performance.**
5. **Our original model test accuracy is 0.9832 while GridSearch CV score on test-set is 0.9835. So, GridSearch CV helps to identify the parameters that will improve the performance for this particular model.**

TEAM MEMBERS

Aayush Bharambe-2201CS02

Abhinav Gupta 2201CS03

Ade Manikanta -2201CS05

Aman Kumar – 2201CS13

Amitesh Raj – 2201CS14

Anshul Chauhan – 2201CS17

Sahil Kumar - 2201CS60

Vikram Balai – 2201CS76

Vinay Prajapati – 2201CS77

Bhavik Netam - 2201CS84

Prantik Biswas - 2201AI28

Pratik Amrit - 2201AI29