CS253 Python Assignment Report

Kawaljeet Singh (220514)

April 13, 2024

1. Methodology

1.1 Data Preprocessing Steps:

1. Reading data: The code reads training and test data from CSV files using Pandas.
2. Converting amount strings to numeric: A custom function `convert_to_numeric` is defined to convert amount strings (e.g., "10 Crore+") to numeric values.
3. Handling missing values: Rows with missing values are dropped from both the training and test datasets using `dropna`.
4. Target and features definition: The target variable is defined as 'Education', while the independent variables are extracted from the dataset, excluding 'ID' and 'Candidate' columns.

1.2 Feature Engineering:

No explicit feature engineering is performed in the code. However, the categorical features are encoded using one-hot encoding, and the 'Education' feature is encoded using label encoding.

1.3 Identifying Outliers:

There is no explicit step for identifying outliers in the code. Outlier detection and treatment could be a valuable addition to improve model robustness.

1.4 Dimensionality Reduction Techniques:

There is no dimensionality reduction technique applied in the code. Techniques like PCA (Principal Component Analysis) could be considered to reduce the dimensionality of the dataset, especially if there are many features.

1.5 Normalization, Standardization, or Transformation:

No normalization or standardization is applied explicitly. However, one-hot encoding is performed for categorical variables using `OneHotEncoder`.

1.6 Other Considerations:

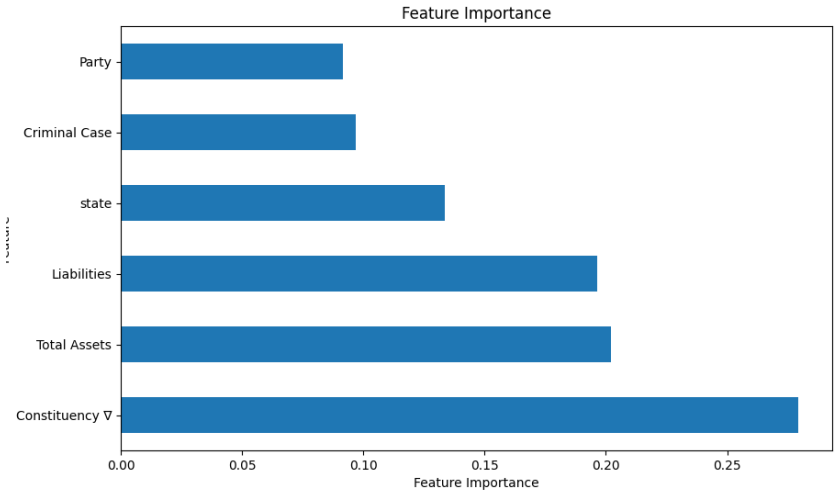
- Model Selection: The code employs a Support Vector Machine (SVM) classifier.
- Cross-Validation: Stratified K-Fold cross-validation with 5 splits is used for model evaluation.
- Grid Search: Grid search is conducted over a parameter grid to find the best hyperparameters for the SVM model.
- Evaluation Metric: The F1 score with micro averaging is used as the evaluation metric for the grid search.
- Model Deployment: Finally, the best model selected from grid search is used to make predictions on the test data, and the predictions are saved to a CSV file for submission.

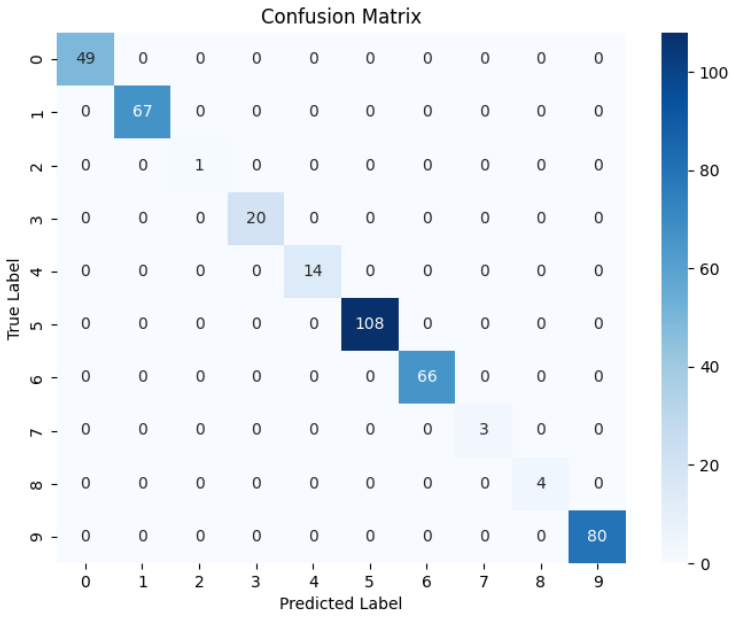
2. Experiment Details

Table 1: Summary of Models Used

Model	Support Vector Machine (SVM)
Reason for Use	SVMs are suitable for classification tasks and can handle high-dimensional data well.
Hyperparameters	C: 0.1, 1, 10 Kernel: linear, poly, rbf, sigmoid
Reason for Hyperparameter Tuning	Hyperparameters like C and kernel type significantly impact SVM model performance. Tuning these parameters helps optimize the model's predictive capability.
Cross-Validation Strategy	Stratified K-Fold (5 splits)
Reason for Cross-Validation	Cross-validation helps in estimating the model's performance on unseen data and prevents overfitting.
Evaluation Metric	F1 Score (micro)
Reason for Metric Choice	F1 Score provides a balance between precision and recall, making it suitable for imbalanced datasets and multi-class classification tasks.

2.1 Data Insights





3. Results

F1 score is 0.24841.
Leaderboard Rank: 69.

4. References

Pandas
Intro to Machine Learning
Model Selection and Boosting - Scikit-learn
Classification using SVC
Category Encoders Examples
Pipelines

5. Github repository

CS253_ML_Assignment