Report on Tabular Data Classification

Data Preprocessing & Feature Engineering

The datasets consisted of 3,238 features and only 315 rows, a highly dimensional setting requiring thorough preprocessing and feature reduction.

Data inspection: Initial exploration involved checking data information and statistics.

Missing values:

- 23 columns had missing data in 116 rows. These rows were removed to maintain data quality.
- Infinite values were replaced with NaNs before removal.

Since there were 3238 features and only 315 rows, it was deemed appropriate to reduce the number of features and only keep ones with high relevance to the target

Feature reduction:

- Variance thresholding < 0.01 was applied to remove near-constant features.
- Highly correlated features (correlation > 0.9) were removed to reduce redundancy.
- Scaling: StandardScaler was applied to features for models sensitive to feature scaling (logistic regression, SVM). Scaling was not required for random forest.

Feature selection:

- After the scaling phase, SelectKBest was used during experimentation to identify top 60 features before finalizing selection methods
- For logistic regression and SVM, elastic net regularization was used to select important features, resulting in 19 selected features after experimentation.
- For random forest, top 30 features were selected based on feature importance scores.
- All of the numbers were finalized after experimentation using cross-validation.

Due to high dimensionality (3238 features, 315 rows), aggressive feature selection was necessary to mitigate overfitting.

Model Architectures & Key Hyperparameters

Logistic Regression:

- **Two-stage pipeline:** ElasticNet-based feature selection (LogisticRegression with penalty='elasticnet', solver='saga') followed by final classifier.
- **Key hyperparameters tuned:** C, l1_ratio for feature selector and classifier.
- Feature selection retained 19 features after experimentation.
- Final selected hyperparameters:
  C=10, l1_ratio=0.9 for the ElasticNet-based feature selector.
  C=1 for the final classifier.

Support Vector Classifier (SVC):

- Linear kernel was ultimately chosen based on cross-validation results.

- Hyperparameters tuned: kernel type (linear, rbf), C, and gamma (for RBF).
- Used 19 scaled features selected from logistic regression (elastic net) phase.
- Final selected hyperparameters:
  C=1.0.

Random Forest Classifier:

- Used importance-based feature selection from a baseline Random Forest. 30 top features were retained after experimentation. Scaling was not applied for random forest due to its insensitivity to feature scaling.
- Used scikit-learn's RandomForestClassifier with class balancing.
- Final selected hyperparameters: n_estimators=300, max_depth=3, max_features=0.1, min_samples_split=15, min_samples_leaf=15.

Cross-Validation Scheme

- Stratified 5-Fold Cross-Validation was applied across all models to ensure class distribution was preserved within each fold.
- Performed using GridSearchCV with scoring based on F1-score.
- Cross-validation was applied to pipelines (feature selection + model) to avoid data leakage.
- Enabled return_train_score=True to compare training vs. validation performance and assess overfitting.

Results Summary

Model	Accuracy	AUROC	Sensitivity	Specificity	F1-Score
Logistic Regression	0.6500	0.6724	0.6905	0.6207	0.6237
Random Forest	0.5900	0.6359	0.5238	0.6379	0.5176
SVC	0.6000	0.6441	0.5476	0.6379	0.5349

Discussion

Strengths:

- Proper handling of high dimensionality through variance thresholding, correlation filtering, univariate feature selection and model based feature selection.
- Multiple models and configurations were evaluated, all with cross-validation and balanced class treatment.
- Evaluation was based on multiple metrics, enabling a nuanced view of model behavior.

Limitations

- The dataset had only 315 rows and over 3,200 features, which made it prone to overfitting and added noise.
- Despite regularization and feature selection, models did not generalize well - training scores were significantly better than testing scores.

- Class imbalance, even though mitigated with class_weight='balanced', may still have impacted classifier performance.

Potential Improvements:

- With more time, I would invest in more advanced feature engineering and selection techniques to select or create features that are more relevant to the target label.
- I would also experiment more rigorously with algorithms and hyperparameter configurations that are better suited for this kind of high-dimensional, low-sample-size problems.