SIT720 – Machine Learning

Task 5.2D

2. **Based on the pre-processed training data from question 1, create three supervised machine learning (ML) models for predicting "Status".**

    a. **Use an appropriate validation method, report performance score using a suitable metric. Is it possible that the presented result is an underfitted or overfitted one? Justify.**

        i. Based on the cross-validation results, the Logistic Regression and SVM models appear to be well-fitted, while the Random Forest model shows signs of overfitting. The Logistic Regression and SVM models have training scores that are slightly higher than their mean cross-validation scores, but the differences are within an acceptable threshold (0.1). This indicates that these models strike a good balance between fitting the training data and generalizing to unseen data. They are less likely to be underfitted or overfitted.

        ii. On the other hand, the Random Forest model has a training score of 1.0, which is significantly higher than its mean cross-validation score of 0.7546. This large difference (greater than the threshold of 0.1) suggests that the Random Forest model is likely overfitting. It performs perfectly on the training data but fails to generalize well to unseen data. Overfitting occurs when a model learns the noise and specific patterns in the training data too closely, leading to poor performance on new, unseen data.

    b. **Justify different design decisions for each ML model used to answer this question.**

        The design decisions for each machine learning model used in this question are as follows:

        Logistic Regression is a simple and interpretable model suitable for binary classification problems when the relationship between features and the target variable is expected to be linear. The max_iter parameter is set to 1000 to allow for more iterations to reach convergence. Random Forest, an ensemble

model combining multiple decision trees, is versatile and can handle high-dimensional data and capture complex relationships. The random_state parameter is set to 42 for reproducibility. SVM is a powerful model that can handle linear and non-linear classification tasks, trying to find the maximally separating hyperplane. It is a good choice when there is a clear separation between classes and when dealing with high-dimensional data.

In addition to model-specific decisions, features were scaled using StandardScaler to ensure similar feature scales, which is important for Logistic Regression and SVM. k-fold cross-validation with k=5 was used for robust performance evaluation, and accuracy was chosen as the scoring metric, suitable for balanced classification problems.

c. Have you optimised any hyper-parameters for each ML model? What are they? Why have you done that? Explain.

- **Hyperparameter optimization:**

  - No hyperparameter optimization has been performed for the machine learning models in the provided code.
  - Hyperparameter optimization involves selecting the best combination of hyperparameters for each model to improve its performance.
  - Some common hyperparameters that could be optimized include:

    - Logistic Regression: C (inverse of regularization strength), penalty (type of regularization), solver (optimization algorithm).
    - Random Forest: n_estimators (number of trees), max_depth (maximum depth of trees), min_samples_split (minimum number of samples required to split an internal node).
    - SVM: C (regularization parameter), kernel (kernel type), gamma (kernel coefficient).

- Hyperparameter optimization is important to find the best configuration that maximizes the model's performance on unseen data and avoids overfitting or underfitting.
- Techniques like grid search or random search can be used to explore different hyperparameter combinations and select the best one based on cross-validation results.

d. What can you do with the label imbalance issue?

- Label imbalance refers to a situation where the classes in the target variable are not equally represented.
- If the label imbalance is severe, it can affect the model's performance and lead to biased predictions towards the majority class.
- Some techniques to handle label imbalance include:
    - Oversampling the minority class: Duplicating or generating synthetic examples of the minority class to balance the class distribution.
    - Undersampling the majority class: Removing examples from the majority class to balance the class distribution.
    - Using class weights: Assigning higher weights to the minority class during training to give it more importance.
    - Using evaluation metrics that are sensitive to class imbalance, such as precision, recall, F1-score, or area under the precision-recall curve (AUPRC).
- The choice of technique depends on the specific problem and the characteristics of the dataset.

e. Finally, make a model recommendation based on the reported results and justify it.

Model recommendation and justification:

Based on the reported cross-validation results, the SVM model seems to be the best choice for this problem. It has the highest mean cross-validation score of 0.7723, indicating good

performance on unseen data. Additionally, the difference between the training score and the mean cross-validation score is within an acceptable threshold, suggesting that the model is well-fitted and not overfitting or underfitting.

The Logistic Regression model also performs well, with a mean cross-validation score of 0.7411 and no signs of overfitting or underfitting. However, the SVM model slightly outperforms Logistic Regression in terms of accuracy.

The Random Forest model, despite having a high training score of 1.0, shows signs of overfitting with a lower mean cross-validation score of 0.7546. This indicates that the model may be too complex and is not generalizing well to unseen data.

Therefore, based on the cross-validation results and the assessment of overfitting/underfitting, the SVM model is recommended for this problem. It achieves the highest accuracy on unseen data and shows no signs of overfitting or underfitting.

4. Analyse the importance of the features for predicting “Status” using two different approaches. Give statistical reasons of your findings.

To figure out which features are most important for predicting the "Status" variable, we can use two methods:

Correlation analysis:

- This method looks at how strongly each feature is related to the "Status" variable.
- We calculate correlation coefficients, which range from -1 to +1.
- A positive value means that as the feature value goes up, the chance of a specific status goes up.
- A negative value means that as the feature value goes up, the chance of a specific status goes down.
- We can do statistical tests to see if the correlations are significant or just due to chance.
- Features with stronger correlations and statistically significant results are considered more important.

Feature importance from machine learning models:

- Some machine learning models, like Random Forest or Gradient Boosting, can tell us how much each feature contributes to the predictions.
- We train the model on the data and get feature importance scores.
- Features with higher scores are more important for predicting the "Status" variable.
- We can compare the results from different models to get a better idea of which features are consistently important.
- These models consider how features work together to make predictions.

It's important to think about both statistical significance (is the result just due to chance?) and practical significance (does the result actually matter in the real world?). We should also use our knowledge of the problem to make sense of the results.

By using both correlation analysis and feature importance from machine learning models, we can get a good understanding of which features are most helpful for predicting the "Status" variable. This can help us choose the best features, understand the model better, and make good decisions.

Kaggle Screenshot: