Machine Learning for Business CIS – 508

Final In class Assignment – Ujwala

1. Accuracy, precision, recall and F1 score on the test set for each classifier in default mode

Default Decision tree

```
Model: DecisionTree
Accuracy: 0.8182
Precision: 0.8333
Recall: 0.8333
F1 Score: 0.8333
Confusion Matrix:
[[5 1]
 [1 4]]

Classification Report:
              precision    recall  f1-score   support

           1       0.83      0.83      0.83         6
           2       0.80      0.80      0.80         5

    accuracy                           0.82        11
   macro avg       0.82      0.82      0.82        11
weighted avg       0.82      0.82      0.82        11
```

Default Random forest

```
Model: RandomForest
Accuracy: 0.7273
Precision: 0.7143
Recall: 0.8333
F1 Score: 0.7692
Confusion Matrix:
[[5 1]
 [2 3]]

Classification Report:
              precision    recall  f1-score   support

           1       0.71      0.83      0.77         6
           2       0.75      0.60      0.67         5

    accuracy                           0.73        11
   macro avg       0.73      0.72      0.72        11
weighted avg       0.73      0.73      0.72        11
```

Default KNN

```
Model: KNN
Accuracy: 0.4545
Precision: 0.0000
Recall: 0.0000
F1 Score: 0.0000
Confusion Matrix:
[[0 6]
 [0 5]]

Classification Report:
              precision    recall  f1-score   support

           1       0.00      0.00      0.00         6
           2       0.45      1.00      0.62         5

    accuracy                           0.45        11
   macro avg       0.23      0.50      0.31        11
weighted avg       0.21      0.45      0.28        11
```

Default SVC

```
Model: LinearSVC
Accuracy: 0.7273
Precision: 0.7143
Recall: 0.8333
F1 Score: 0.7692
Confusion Matrix:
[[5 1]
 [2 3]]

Classification Report:
              precision    recall  f1-score   support

           1       0.71      0.83      0.77         6
           2       0.75      0.60      0.67         5

    accuracy                           0.73        11
   macro avg       0.73      0.72      0.72        11
weighted avg       0.73      0.73      0.72        11
```

2. Best parameters found through hyper parameter tuning of the Random Forest classifier.

```
Best Parameters found by RandomizedSearchCV:
{'n_estimators': 200, 'min_samples_split': 5, 'min_samples_leaf': 4, 'max_depth': 20}
```

3. Average accuracy, precision, recall and F1 score on the test set from the hyper parameter tuned Random Forest classifier.

```
Tuned Random Forest Metrics:
Accuracy: 0.64
Precision: 0.75
Recall: 0.50
F1 Score: 0.60

Confusion Matrix for Tuned Random Forest:
[[3 3]
 [1 4]]

Classification Report for Tuned Random Forest:
              precision    recall  f1-score   support

           1       0.75      0.50      0.60         6
           2       0.57      0.80      0.67         5

    accuracy                           0.64        11
   macro avg       0.66      0.65      0.63        11
weighted avg       0.67      0.64      0.63        11
```

4. Comparison between result of default results with the best hyper parameter tuned results for the Random Forest classifier

   The default model performs better across all key metrics, including accuracy (0.7273 vs. 0.6364), recall (0.8333 vs. 0.50), and F1 score (0.7692 vs. 0.6). Although **precision slightly improves** for the tuned version (0.75 vs. 0.7143), it is not sufficient to offset the overall decline in performance.
   The drop in performance is likely due to overfitting or suboptimal hyper parameter choices from the random search, exacerbated by the small test dataset size, which makes the evaluation results sensitive to small changes. Additionally, hyper parameter tuning might have reduced the model's generalization ability due to an overemphasis on certain training set patterns.
   A larger test set or more refined hyper parameter search might yield better results.

5. top 5 features

```
Top 5 Important Features from Random Forest:
Albumin         0.141730
Bilirubin       0.114195
Varices_no      0.087366
PROTIME         0.085206
Ascites_yes     0.058633
dtype: float64
```

6. Results Accuracy report

	Accuracy	Precision	Recall	F1 Score	Confusion Matrix
LinearSVC	0.727273	0.714286	0.833333	0.769231	[[5, 1], [2, 3]]
DecisionTree	0.818182	0.833333	0.833333	0.833333	[[5, 1], [1, 4]]
RandomForest	0.727273	0.714286	0.833333	0.769231	[[5, 1], [2, 3]]
KNN	0.454545	0.0	0.0	0.0	[[0, 6], [0, 5]]
Tuned RandomForest	0.636364	0.75	0.5	0.6	[[3, 3], [1, 4]]
Stacking	0.636364	0.75	0.5	0.6	[[3, 3], [1, 4]]

The stacking model's results (accuracy: 0.6364, precision: 0.75, recall: 0.50, F1 score: 0.6) are slightly better than the tuned Random Forest classifier in terms of precision but are otherwise similar in performance. However, the stacking model underperforms compared to the default Decision Tree and Random Forest models in all metrics, especially accuracy (0.8182 and 0.7273, respectively).

The stacking model's performance may have been hindered by overfitting or the inability of the Multilayer Perceptron (MLP) final estimator to generalize effectively on the small test dataset. Additionally, the weak performance of some base models (e.g., KNN) may have negatively impacted the ensemble's overall results. A larger dataset or more robust base classifiers could potentially improve the stacking model's performance.

7. Performance Improvement

- To improve results, could have done hyper parameter tuning for all base models to enhance their performance.
- Replacing weaker models, such as KNN, with stronger alternatives like Gradient Boosting.
- Optimize the final estimator by tuning or switching to simpler models like Logistic Regression.
- Refine the dataset through feature selection, engineering, and balancing techniques like SMOTE for small or imbalanced data.
- Use cross-validation stacking with out-of-fold predictions to avoid data leakage and ensure robust performance evaluation.
- A larger test set could also provide more accurate insights into model performance.