ASSIGNMENT-15

**A brief report summarizing the comparative analysis results and practical implications**

**Comparative Analysis of LightGBM and XGBoost on Titanic Dataset**

**1. Introduction**

In this analysis, we compared the performance of two popular machine learning algorithms — **LightGBM** (Light Gradient Boosting Machine) and **XGBoost** (Extreme Gradient Boosting) — on the Titanic dataset. The goal was to predict whether a passenger survived the Titanic disaster, based on various features such as age, gender, class, and fare.

**2. Data Preprocessing**

Before training the models, we performed several key preprocessing steps:

* **Missing Value Handling**: We imputed missing values for columns like Age and Embarked using the median and mode respectively.
* **Encoding Categorical Variables**: We applied **One-Hot Encoding** to categorical variables like Sex, Embarked, and Pclass to convert them into numerical features.
* **Feature Engineering**: We dropped non-predictive features like Name, Ticket, and Cabin from the dataset.
* **Data Splitting**: The dataset was split into training (80%) and testing (20%) sets.

**3. Model Building**

Both **LightGBM** and **XGBoost** models were trained on the preprocessed dataset:

* **LightGBM** is known for being highly efficient in terms of speed and memory usage, particularly on large datasets with many features.
* **XGBoost** is widely regarded as the go-to algorithm for many Kaggle competitions due to its robustness and high accuracy, though it tends to be slower than LightGBM on large datasets.

Both models were trained using the default hyperparameters first, and then **hyperparameter tuning** (e.g., adjusting the number of trees, learning rate, and maximum depth) was performed using **Grid Search** and **Randomized Search** to further improve performance.

**4. Evaluation Metrics**

We evaluated both models using the following performance metrics:

* **Accuracy**: Proportion of correct predictions.
* **Precision**: The proportion of positive predictions that were actually correct.
* **Recall**: The proportion of actual positives that were correctly predicted.
* **F1-Score**: The harmonic mean of precision and recall, providing a balanced measure of performance.

**5. Results**

| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
| --- | --- | --- | --- | --- |
| **LightGBM** | 0.83 | 0.83 | 0.77 | 0.80 |
| **XGBoost** | 0.82 | 0.80 | 0.76 | 0.78 |

* **LightGBM** achieved slightly better performance across all metrics, particularly in terms of accuracy and F1-score.
* **XGBoost**, while also a strong performer, showed slightly lower precision, indicating it may be more conservative in predicting survival, leading to more false positives.

**6. Model Performance Insights**

* **LightGBM** outperformed **XGBoost** in terms of overall accuracy and F1-score, making it a more suitable choice for this dataset if speed and performance are prioritized.
* **XGBoost** provided competitive results but may require more fine-tuning to match LightGBM's performance on this specific dataset. It’s possible that further hyperparameter tuning could improve its performance to rival LightGBM's.

**7. Practical Implications**

* **LightGBM** is particularly well-suited for:
  + Large datasets with many features.
  + Applications where speed and memory efficiency are critical, especially in real-time predictions or production environments.
* **XGBoost** is often the algorithm of choice in competitions or for applications where high accuracy is the primary goal, and where the model is being trained on relatively smaller datasets or where model explainability is critical.
* **Model Deployment**: Given the comparable performances, **LightGBM**'s faster training time and reduced memory usage make it a better candidate for deployment in environments with large datasets or limited computational resources.

**8. Conclusion**

Both **LightGBM** and **XGBoost** demonstrated strong predictive power in the Titanic dataset. While both algorithms are powerful, the choice between them depends on specific project needs:

* For speed and efficiency, **LightGBM** is a superior choice.
* For high accuracy, especially in smaller, more complex datasets, **XGBoost** is likely to provide better results.