Titanic Survival Prediction: Summary Report

Key Findings & Insights

Key patterns:

- **Pclass** and **Sex** are strong predictors: survival was higher for women and passengers in higher classes.
- **Age** distribution shows children had better survival chances.
- **Fare** ⟶ higher fare, better survival.

Feature engineering steps included:

- Filling missing values (`Age, Embarked, Fare`).
- Creating new features: Cabin_known, Title, FamilySize, AgeBin.
- Encoding categorical variables (Sex, Embarked, Title, AgeBin).
- Dropping irrelevant columns (PassengerId, Name, Ticket).

**Correlation analysis confirmed that Sex, Fare, and engineered features like Title and FamilySize had strong relationships with survival.

Model Performance Summary

- I built a Decision Tree and Random Forest classifier from scratch.
- Initial training on an 80/20 split (train/test) gave an accuracy ≈ 83%

- Hyperparameter tuning was performed (hyperparameters → n_estimators, max_depth, min_samples_split).
    - The top 5 configurations gave accuracy around 0.83+.
    - For example, n_estimators=200, max_depth=10, min_samples_split=2 appeared among the top results with an accuracy of 0.843.

Results Interpretation

- The model achieved an accuracy of **83.2%**.
- Precision: **0.83**, Recall: **0.74**, F1-score: **0.78**
- After hyperparameter tuning, accuracy improved to **84.3%** [for `n_estimators` = 200 and `max_depth` = 10.
- **Feature importance**(by correlation and `feature_importance_pairs`): Age, Fare, and Pclass dominated the model's predictions. They were followed by `SibSp` and `Parch`.

Business Insights

- Women and children were prioritized in rescue (clear survival bias).
- High-class passengers(1st and 2nd class passengers) had better access to lifeboats and safety.
- Fare also added a bias. Higher-paying passengers had better survival.
- Actionable Insight*:* In real situations, evacuation procedures must be designed such that they avoid bias based on gender, ticket price, or ticket class, ensuring **equal survival chances** for all passengers.

Model Limitations

- Accuracy only improved slightly (83.2% → 84.3%) after tuning,indicating **overfitting**.
- **Cross-validation** was not used (i.e, testing the model across multiple splits; we only split it once, where 80% → training and 20% →testing
- Since a single split doesn't prove generalisation, cross-validation of data is necessary

---

Challenges Faced & How I Solved Them

- **Computation time:** Due to the extensive amount of data points, the computing time was very long; as a result, I had to switch my runtime to "T4 GPU". After running the code multiple times to test the output,the daily data limit of using the T4 GPU was exceeded.

  Solution: I had to use "CPU" and wait for the output :((

- **Lack of knowledge:** I had absolutely no idea that Decision Tree and Random Forest Classifier could be implemented from scratch. I found that to be the trickiest part of the entire task. Secondly, I also found identifying important features kind of tricky.
  Solution: To be very honest, I turned to YouTube tutorials and AI for this part.

Suggestions for Future Improvement

- Using **K-Fold Cross-Validation** instead of a one train-test split to get a clearer estimate. K-Fold CV basically divides the dataset into 'K' equal subsets. In this process, the model is trained on 'K-1' folds and validated on the remaining single fold, and this process is repeated K times, with each fold serving as the validation set once.
- Using **SciKit Learn's Random Forest Classifier Library** can often **improve accuracy** by several percentage points compared to a basic from-scratch version, mainly due to advanced optimizations and richer tuning options.
