Titanic Survival Prediction with Random Forest

This project is a complete Titanic survival prediction pipeline using classical machine learning models. The focus is on extracting useful features, training a Random Forest model, and generating a submission file for Kaggle's Titanic challenge.

🚀Project Summary

Objective: Predict the survival of passengers on the Titanic using their demographic and ticket information.

Key Techniques:

- Feature engineering
- Data imputation
- Label encoding
- Random Forest classification
- F1 score evaluation
- Kaggle submission CSV generation

File Structure

```
Titanic_Survival_Kaggle/
├── DATASET/
│   ├── train.csv
│   ├── test.csv
│   └── submission7.csv
├── titanic_random_forest_script.py  # Main model code
├── README.md
```

Requirements

Make sure the following libraries are installed:

```
pip install pandas numpy matplotlib seaborn scikit-learn
```

Feature Engineering

Feature	Description
`Title`	Extracted from name and grouped
`Sex`	Categorical to numeric (0/1)
`Embarked`	Mapped: S=0, C=1, Q=2
`FamilySize`	Combined `SibSp + Parch + 1`
`IsAlone`	Boolean feature if passenger is alone
`AgeBand`	Age binned into 5 categories
`HasCabin`	1 if cabin data exists, 0 if 'No Cabin'
`Fare`	Median filled

Dropped columns: `PassengerId`, `Name`, `Ticket`, `Cabin`, `Age` after use.

Model Used

```
RandomForestClassifier(
    criterion='entropy',
    n_estimators=300,
    max_depth=10,
    min_samples_split=10,
    min_samples_leaf=4,
    class_weight='balanced',
    max_samples=0.8,
    random_state=42
)
```

This model was chosen after experimentation and balances overfitting well.

💒Model Evaluation

```
F1 Score for train: ~0.82
F1 Score for test:  ~0.81
Kaggle Submission Score: ~0.76 - 0.78
```

🛕 How to Use

1. Ensure your `train.csv` and `test.csv` are in the `DATASET/` folder.
2. Run the Python script or notebook.
3. It will generate `submission7.csv` with predictions.
4. Upload `submission7.csv` to Kaggle.

Future Improvements

- Add deck extraction from cabin (e.g. A, B, C...)
- Add `FarePerPerson` = Fare / FamilySize
- Try ensembling with Gradient Boosting or XGBoost
- Use `GridSearchCV` to tune hyperparameters

🏡 Author

This project was developed as part of a Kaggle ML challenge learning path. You are encouraged to fork, experiment, and extend it.

License

This code is open-source and free for non-commercial use.