ML Classification – Titanic Survival Prediction (Detailed Design Document)

1. Project Overview

This project focuses on solving a **binary classification problem** using classical Machine Learning algorithms. The goal is to predict whether a passenger survived the Titanic disaster based on demographic and travel-related features such as age, gender, passenger class, fare, etc.

This assignment is intentionally chosen because: - It is **interviewer-friendly** and widely recognized - The problem statement is easy to explain - It allows demonstration of **end-to-end ML workflow** - Multiple algorithms and evaluation metrics can be applied and compared

In interviews, this project helps validate: - Your understanding of ML fundamentals - Your ability to preprocess real-world data - Your reasoning behind model selection and evaluation

---

2. Problem Statement

Given historical passenger data from the Titanic, we aim to predict the target variable:

- **Survived** (1 = Survived, 0 = Did not survive)

This is a **supervised learning problem** because labeled data is available, and specifically a **binary classification task** because there are only two possible outcomes.

---

3. Dataset Description

Dataset Source

The dataset is obtained from **Kaggle – Titanic: Machine Learning from Disaster**.

Files used: - `train.csv` – used for training and evaluation

Important Columns

- `Survived`: Target variable
- `Pclass`: Passenger class (1st, 2nd, 3rd)
- `Sex`: Gender of passenger
- `Age`: Age in years (contains missing values)
- `SibSp`: Number of siblings/spouses aboard
- `Parch`: Number of parents/children aboard
- `Fare`: Ticket fare
- `Embarked`: Port of embarkation (categorical)

Why this dataset is good for ML learning

- Contains **both numerical and categorical features**
- Has **missing values**, simulating real-world data
- Requires **feature preprocessing**
- Allows multiple ML algorithms to be compared

---

4. ML Workflow Overview

The project follows a structured Machine Learning pipeline:

1. Data Loading
2. Exploratory Data Analysis (EDA)
3. Feature Engineering & Preprocessing
4. Train-Test Split
5. Model Training
6. Model Evaluation
7. Model Comparison and Final Model Selection

All four machine learning models were evaluated using a consistent set of performance metrics to ensure a fair and meaningful comparison. The goal of this step was not only to identify the most accurate model, but also to understand how each model behaves in terms of false positives and false negatives.

The following metrics were used:

- Accuracy: Measures overall correctness of the model predictions.
- Precision: Indicates how many of the predicted survivors were actually survivors.
- Recall: Indicates how many of the actual survivors were correctly identified by the model.
- F1-score: Harmonic mean of precision and recall, providing a balanced evaluation.

Model Performance Summary

Model	Accuracy	Precision	Recall	F1-score
Logistic Regression	0.77	0.72	0.67	0.69
Random Forest	**0.82**	0.78	**0.72**	**0.75**
SVM	0.81	**0.87**	0.59	0.71
Gradient Boosting	0.81	0.82	0.65	0.73

Interpretation of Results

Logistic Regression served as a strong baseline model. It is easy to interpret and computationally efficient, but its performance was lower than the ensemble-based models.

Support Vector Machine achieved the highest precision, meaning it was very confident when predicting survival. However, its lower recall indicates that it missed several actual survivors.

Gradient Boosting showed competitive performance across all metrics but did not outperform Random Forest in overall balance.

Random Forest delivered the best overall performance with the highest accuracy, recall, and F1-score, making it the most balanced and reliable model.

Final Model Selection

Four classification models were evaluated: Logistic Regression, Random Forest, Support Vector Machine (SVM), and Gradient Boosting.

Model performance was compared using Accuracy, Precision, Recall, and F1-score to ensure a comprehensive evaluation rather than relying on accuracy alone.

Random Forest achieved the highest F1-score (0.75) and Recall (0.72), indicating its superior ability to correctly identify survivors while maintaining strong precision.

Although SVM achieved the highest precision, it suffered from low recall, meaning it failed to detect a significant portion of actual survivors. Gradient Boosting performed competitively but did not surpass Random Forest in overall balance.

Therefore, **Random Forest was selected as the final model** for this project due to its balanced and robust performance across all evaluation metrics.

jupyter Classification-Pipline Last Checkpoint: 3 days ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)

[22…]
```python
def evaluate_model(model, X_test, y_test):
    """
    Evaluates a trained classification model and returns key metrics.
    """
    y_pred = model.predict(X_test)

    return {
        "Accuracy": accuracy_score(y_test, y_pred),
        "Precision": precision_score(y_test, y_pred),
        "Recall": recall_score(y_test, y_pred),
        "F1-score": f1_score(y_test, y_pred)
    }
```

[23…]
```python
model_results = []

model_results.append({
    "Model": "Logistic Regression",
    **evaluate_model(log_model, X_test, y_test)
})

model_results.append({
    "Model": "Random Forest",
    **evaluate_model(rf_model, X_test, y_test)
})

model_results.append({
    "Model": "Support Vector Machine",
    **evaluate_model(svm_model, X_test, y_test)
})

model_results.append({
    "Model": "Gradient Boosting",
    **evaluate_model(gb_model, X_test, y_test)
})

comparison_df = pd.DataFrame(model_results)
comparison_df
```

[23…]

	Model	Accuracy	Precision	Recall	F1-score
0	Logistic Regression	0.770950	0.718750	0.666667	0.691729
1	Random Forest	0.815642	0.781250	0.724638	0.751880
2	Support Vector Machine	0.810056	0.872340	0.594203	0.706897
3	Gradient Boosting	0.810056	0.818182	0.652174	0.725806

[ ]:
```python
### Model Selection

Based on the comparison table, Random Forest achieved the highest overall
performance across Accuracy, Precision, Recall, and F1-score.

Therefore, Random Forest is selected as the final model for this project.
```

8. Conclusion and Key Learnings

This project successfully demonstrates a complete end-to-end Machine Learning classification workflow, starting from raw data understanding to final model selection. By working on the Titanic Survival Prediction problem, we explored how real-world data requires careful preprocessing, thoughtful feature handling, and the use of multiple evaluation metrics rather than relying on accuracy alone.

The comparative analysis of four algorithms — Logistic Regression, Random Forest, Support Vector Machine, and Gradient Boosting — highlighted the strengths and weaknesses of different modelling approaches. While simpler models such as Logistic Regression provided interpretability and a strong baseline, ensemble methods showed superior performance by capturing complex patterns in the data.

Random Forest was selected as the final model due to its balanced performance across accuracy, precision, recall, and F1-score. This choice reflects an important industry practice: selecting models based on business-relevant trade-offs and robustness, not just raw accuracy.

Overall, this assignment serves as a solid foundation project that demonstrates both theoretical knowledge and practical Machine Learning skills expected at an entry-to-mid professional level.