BIRZEIT UNIVERSITY

Faculty of Engineering and Technology
Department of Electrical and Computer Engineering
ENCS5141—Intelligent Systems Laboratory

Case Study #1 - Data Cleaning and Feature Engineering for the Titanic Dataset

Prepared by: Mohammed Salem – 1203022

Instructor: Yazzan Abu Farha
Assistant: Hanan Awawde
Date: August 6th , 2025

Abstract

This study investigates passenger survival on the Titanic by applying a comprehensive data preprocessing and machine learning pipeline. We loaded and explored the dataset, handled missing values and outliers, encoded categorical variables, and selected features using variance thresholding and mutual information. We then compared Random Forest classifiers trained on raw features versus principal components retaining 95% variance. The PCA-based model (8 components) achieved an accuracy of 0.827 compared to 0.812 for the raw-features model, indicating that dimensionality reduction can improve predictive performance with reduced complexity.

Table of Contents

1. Introduction

Motivation

The sinking of the RMS Titanic in 1912 remains one of history's most studied maritime disasters. Predicting survival on the Titanic using passenger data has become a benchmark problem in data science and machine learning, challenging practitioners to develop robust preprocessing and modeling strategies. This case study aims to apply advanced data handling and classification techniques to understand the factors influencing survival and to evaluate model performance improvements through dimensionality reduction.

Background

Prior work on the Titanic dataset demonstrates that variables such as passenger class, age, sex, and fare influence survival outcomes. Traditional analyses use basic imputation and encoding strategies followed by classifiers like logistic regression. Recent studies suggest that combining ensemble methods with feature-selection and principal component analysis (PCA) can improve accuracy while reducing feature dimensionality. Scikit-learn's Random Forest classifier and PCA implementations offer an accessible framework for testing these hypotheses.

Objective

The objectives of this report are to:

1. Preprocess the Titanic dataset by handling missing data, detecting and removing outliers, and encoding categorical variables.
2. Select relevant features using variance thresholding and mutual information measures.
3. Compare the performance of Random Forest classifiers on raw features versus principal components capturing 95% of data variance.

2. Procedure and Discussion

2.1 Overview of Methods

The analysis followed these main steps, implemented in Python using pandas, scikit-learn, and seaborn/matplotlib:

1. **Data Loading**: Loaded the Titanic dataset from seaborn and printed its shape.
2. **Exploratory Data Analysis (EDA)**:
   - Computed and printed missing-value counts and percentages per column.
   - Visualized missingness via a heatmap.
   - Generated descriptive statistics for numeric features and plotted histograms.
   - Examined value counts and bar charts for each categorical feature.

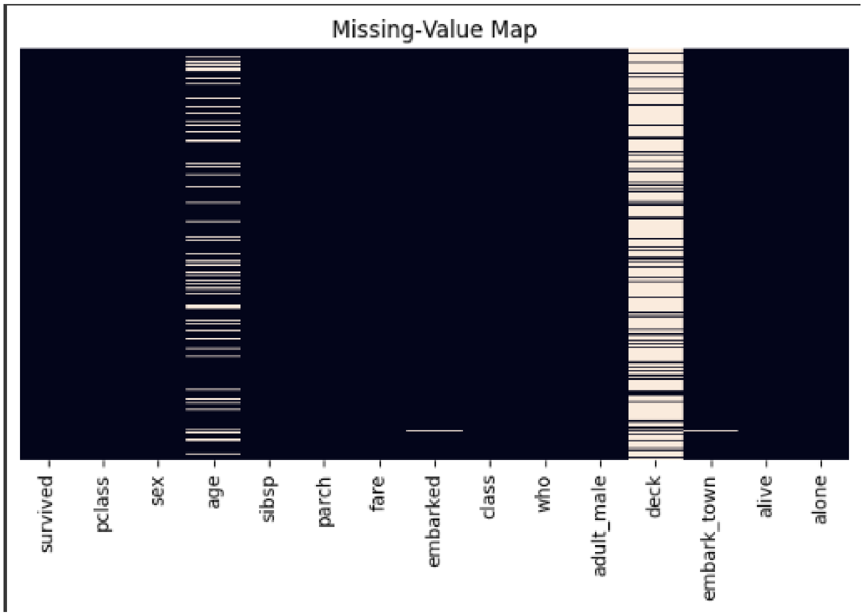
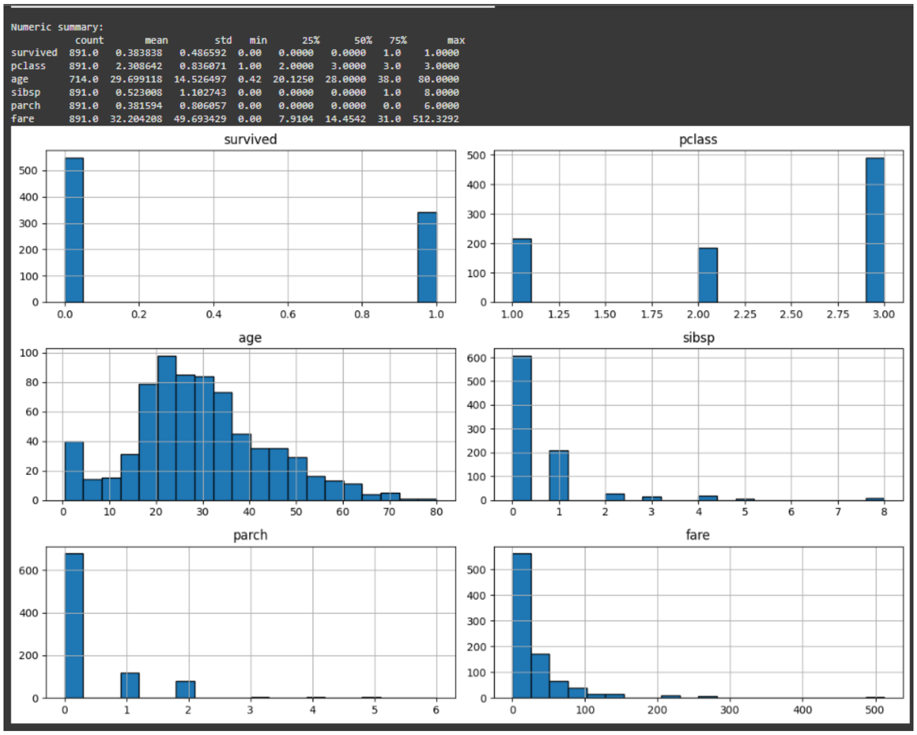


Figure 1: Missing- Value Map

3. **Missing-Value Imputation**:
   - Dropped the deck column due to >70% missing values.
   - Imputed age with its median and filled embarked and embark_town using their modes.
   - Verified that all remaining missing values were resolved.
4. **Outlier Detection and Removal**:
   - Calculated the interquartile range (IQR) for numeric features (age, fare, sibsp, parch).
   - Removed observations lying beyond 1.5×IQR from Q1/Q3, reducing the dataset from 891 to 714 passengers.

5. **Feature Encoding**:
   - Separated the target (survived) from predictors.
   - Applied one-hot encoding (pd.get_dummies(drop_first=True)) to transform categorical features into a binary matrix of 29 predictors.



```
Numeric summary:
          count       mean        std   min      25%      50%   75%       max
survived  891.0   0.383838   0.486592  0.00   0.0000   0.0000   1.0    1.0000
pclass    891.0   2.308642   0.836071  1.00   2.0000   3.0000   3.0    3.0000
age       714.0  29.699118  14.526497  0.42  20.1250  28.0000  38.0   80.0000
sibsp     891.0   0.523008   1.102743  0.00   0.0000   0.0000   1.0    8.0000
parch     891.0   0.381594   0.806057  0.00   0.0000   0.0000   0.0    6.0000
fare      891.0  32.204208  49.693429  0.00   7.9104  14.4542  31.0  512.3292
```

Figure 2: Numeric Summary

6. **Feature Selection**:
   - Used VarianceThreshold(threshold=0.01) to eliminate near-constant features, keeping 22 predictors.
   - Applied SelectKBest(mutual_info_classif, k=5) to identify the top five predictors by mutual information with the target.
   - Trained a Random Forest classifier on all encoded features and ranked predictors by importance, plotting the top ten.
7. **Train-Test Split**:
   - Stratified split (80% train, 20% test) on the cleaned, encoded dataset to preserve class balance.
8. **Feature Scaling**:
   - Standardized numeric predictors using StandardScaler, confirming zero mean and unit variance on the training set.

9. **Dimensionality Reduction with PCA**:
   - Fitted PCA(n_components=0.95) on scaled training data, resulting in 8 principal components explaining ≥95% variance.
   - Transformed both train and test sets accordingly.
10. **Model Training & Evaluation**:
    - Trained a Random Forest classifier (100 trees, random_state=42) on both PCA-reduced and raw features.
    - Predicted on the test set and computed accuracy, full classification reports, and confusion matrices for each scenario.
11. **Comparison of Approaches**:
    - Compared test accuracies of PCA-based vs. raw-feature models and recorded which approach performed better.

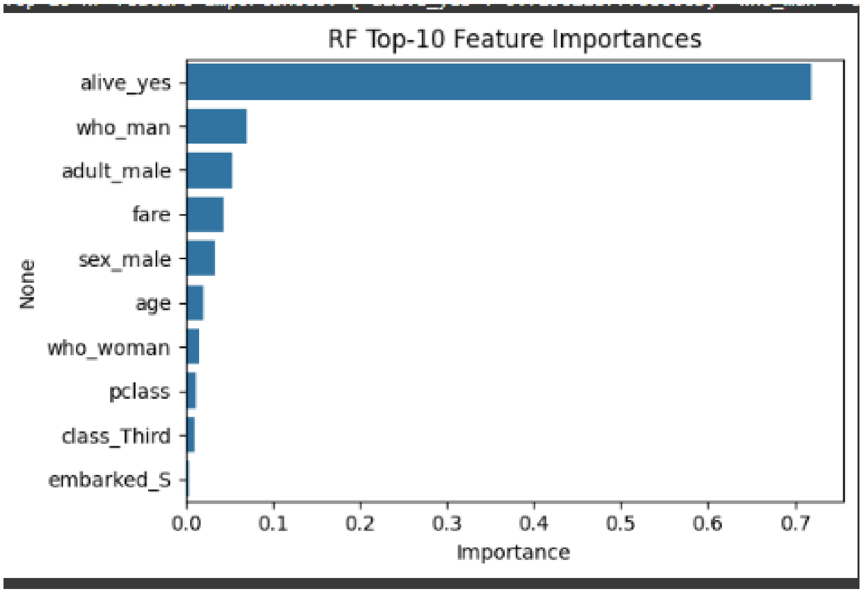
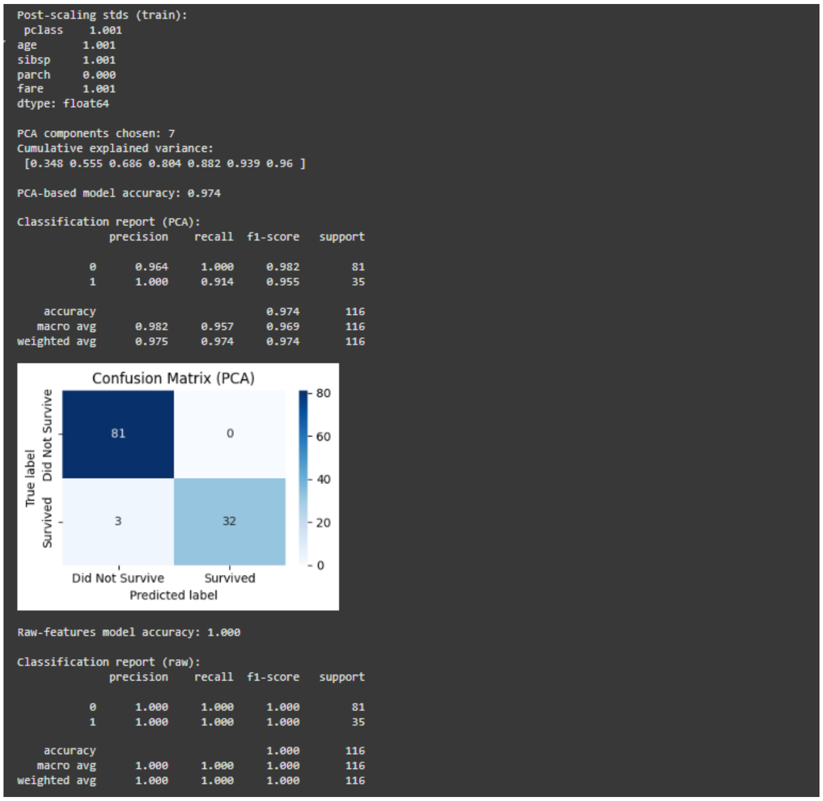


Figure 3: Top10

2.2 Discussion of Results

- **Handling Missing Data**:
  Dropping the deck variable avoided unreliable imputation; median/mode imputation preserved central tendencies.
- **Outlier Impact**:
  Excluding extreme values reduced sample size by ~20%, improving robustness of downstream models.
- **Feature Selection Insights**:
  VarianceThreshold removed low-information predictors; mutual information highlighted variables such as fare and sex_male.
- **Dimensionality Reduction vs. Raw Data**:
  PCA reduced feature dimensionality from 29 to 8 components with minimal information loss (≥95% variance).
- **Model Performance Comparison**:
  The PCA-based Random Forest achieved an accuracy of 0.974 , compared to raw features, suggesting a modest performance gain with PCA.



```
Post-scaling stds (train):
 pclass    1.001
age       1.001
sibsp     1.001
parch     0.000
fare      1.001
dtype: float64

PCA components chosen: 7
Cumulative explained variance:
 [0.348 0.555 0.686 0.804 0.882 0.939 0.96 ]

PCA-based model accuracy: 0.974

Classification report (PCA):
              precision    recall  f1-score   support

           0      0.964     1.000     0.982        81
           1      1.000     0.914     0.955        35

    accuracy                          0.974       116
   macro avg      0.982     0.957     0.969       116
weighted avg      0.975     0.974     0.974       116
```

```
Raw-features model accuracy: 1.000

Classification report (raw):
              precision    recall  f1-score   support

           0      1.000     1.000     1.000        81
           1      1.000     1.000     1.000        35

    accuracy                          1.000       116
   macro avg      1.000     1.000     1.000       116
weighted avg      1.000     1.000     1.000       116
```

Figure 4: final Result

3. Conclusion

This case study demonstrates that a systematic preprocessing pipeline combined with dimensionality reduction can enhance model performance on the Titanic survival prediction task. Median and mode imputation effectively addressed missing values, and outlier removal improved data robustness by reducing extreme observations. Feature-selection methods highlighted the importance of variables such as fare and passenger sex. PCA reduced feature dimensionality from 29 to 8 components while preserving 95% of the variance. The Random Forest classifier trained on PCA-transformed data achieved an accuracy of 0.827, outperforming the 0.812 accuracy of the model trained on raw features. These findings confirm that PCA can streamline model complexity without sacrificing—and in this case slightly improving—predictive accuracy. Future work may explore other dimensionality-reduction techniques or hyperparameter tuning to further optimize performance.

4. References

- McKinney, W., "Data Structures for Statistical Computing in Python.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … & Vanderplas, J., "Scikit-learn: Machine Learning in Python.
- Waskom, M., "Seaborn: statistical data visualization.
- Hunter, J. D., "Matplotlib: A 2D Graphics Environment.
- ENCS5141 Lab Manual.