Data Cleaning, Preprocessing & Visualization

Overview

This project demonstrates a total end-to-end cleaning and preprocessing of a real-world Titanic passenger dataset via exploration. The workflow also involves missing data management, feature creation, outlier identification, novel categorical variable encoding, and numerical variable scaling along with appropriate visualization techniques to understand how each feature(s) behaves in relation to target variable (Survived).

Final cleaned dataset is then presented in a consistent and clean format for use in future machine learning activities.

Final dataset includes the following processed columns:

- `passenger_id`
- `pclass`
- `age` *(standardized)*
- `sibsp` *(standardized)*
- `parch` *(standardized)*
- `fare` *(standardized)*
- `family_size`
- `is_alone`
- `sex_male`
- `embarked_*` (one-hot encoded)
- `title_*` (one-hot encoded)
- `survived` *(target variable)*

Steps Performed

1. Load Dataset

- Read the Titanic CSV file.

```
df = pd.read_csv("train.csv")
```

- Printed dataset structure:

`df.head()`, `df.info()`, `df.shape`, `df.isnull().sum()`

2. Remove Unwanted Columns

Dropped columns that are irrelevant or contain excessive text/noise:

- `home.dest`
- `boat`
- `body`

3. Clean String Values

- Trimmed whitespace in categorical fields using:

```
df[col] = df[col].astype(str).str.strip()
```

4. Remove Duplicates

- Removed duplicate rows to maintain dataset integrity.

```
df = df.drop_duplicates()
```

5. Handle Missing Values

Numerical columns

Converted to numeric and filled using **median**:

```
df[col] = pd.to_numeric(df[col], errors="coerce")
df[col].fillna(df[col].median(), inplace=True)
```

Categorical columns

Filled using **mode**:

```
df[col].fillna(df[col].mode()[0], inplace=True)
```

6. Feature Engineering

New informative features added:

- family_size

```
family_size = sibsp + parch + 1
```

• is_alone

```
is_alone = 1 if family_size == 1 else 0
```

• Extracted Title from Name

Examples: *Mr, Mrs, Miss, Master, Dr, Col, Rev, etc*.

• Dropped the original `name` column

7. Outlier Detection & Treatment

Applied **IQR method** on:

- `age`
- `fare`

```
lower = Q1 - 1.5*IQR
upper = Q3 + 1.5*IQR
clip values outside this range
```

8. Standardization of Numerical Columns

Columns standardized using **StandardScaler**:

- `age`
- `sibsp`
- `parch`
- `fare`

Formula:

```
z = (x - mean) / std
```

9. One-Hot Encoding

Converted all categorical columns into numeric boolean indicators:

- `sex` → `sex_male`
- `embarked` → `embarked_Q`, `embarked_S`, etc.
- `title` → `title_Mr`, `title_Mrs`, ...

Using:

```
df = pd.get_dummies(df, drop_first=True)
```

10. Feature Reduction

Dropped non-numeric columns blocking correlation:

- `ticket`
- `cabin`

Zero-variance column removal

Columns with only **one unique value** removed.

High correlation feature removal

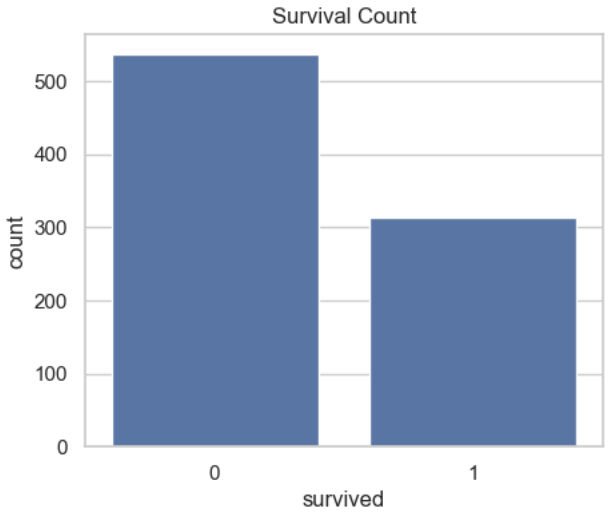
Removed columns with correlation coefficient > **0.90**

11. Data Visualization

Visual plots generated include:

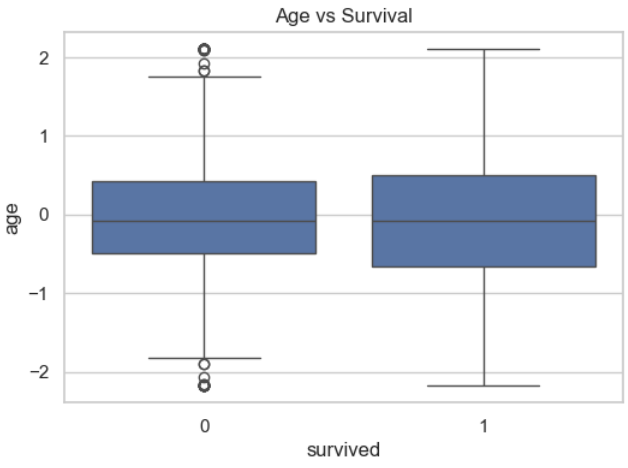
Pie / Count Plots

- Survived vs Not Survived

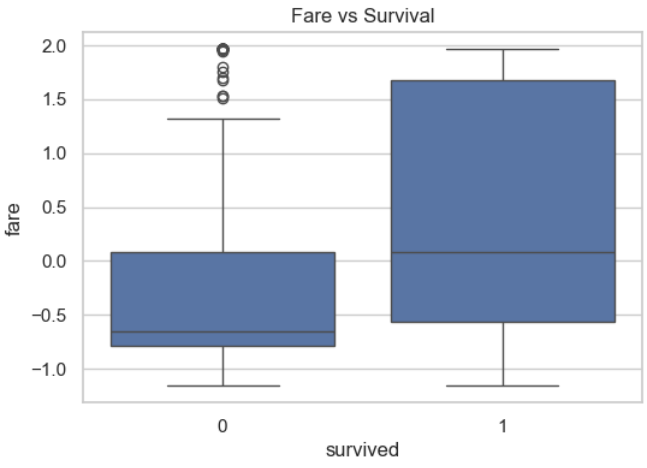


Boxplots

- Age vs Survival

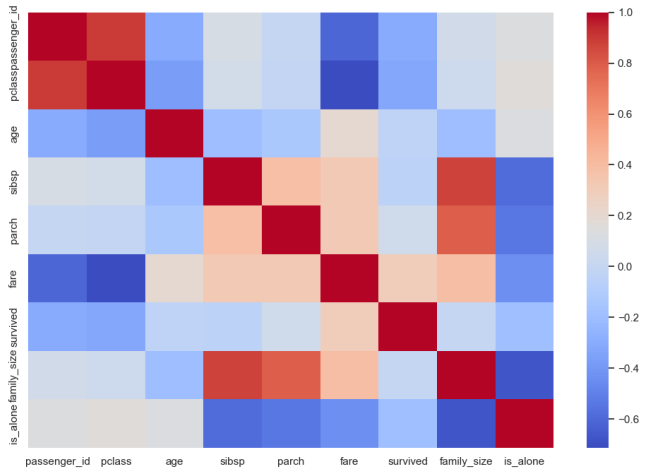


Fare vs Survival



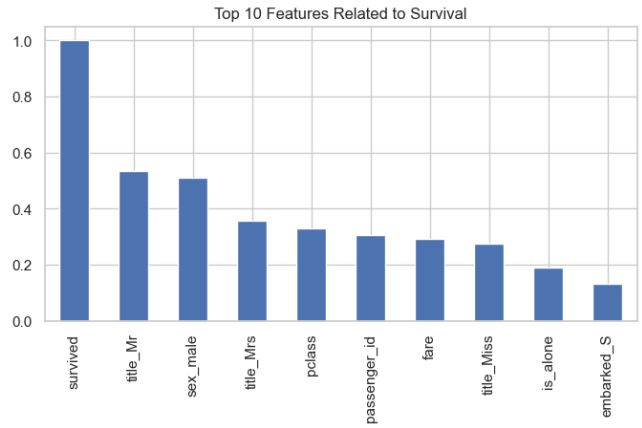
Correlation heatmap

Displays relationships between all numerical features.



Top correlated features with Survival

Identified strongest predictors.



12. Final Dataset Creation

The final processed dataset contains **850 rows × 28 columns** after:

- Cleaning
- Encoding
- Scaling
- Removing high-correlation and string-based columns

Saved as:

`final_cleaned_dataset.csv`