```
X_valid = valid_df.drop(columns=['survived', 'important_title', 'embarked_Q', 'relatives', 'fare', 'alone', 'height' ])
X_valid.head()
y_valid = valid_df['survived']
y_valid.head()
```

```
687    0
664    1
935    1
133    1
339    1
Name: survived, dtype: int64
```

```
X_train.head()
```

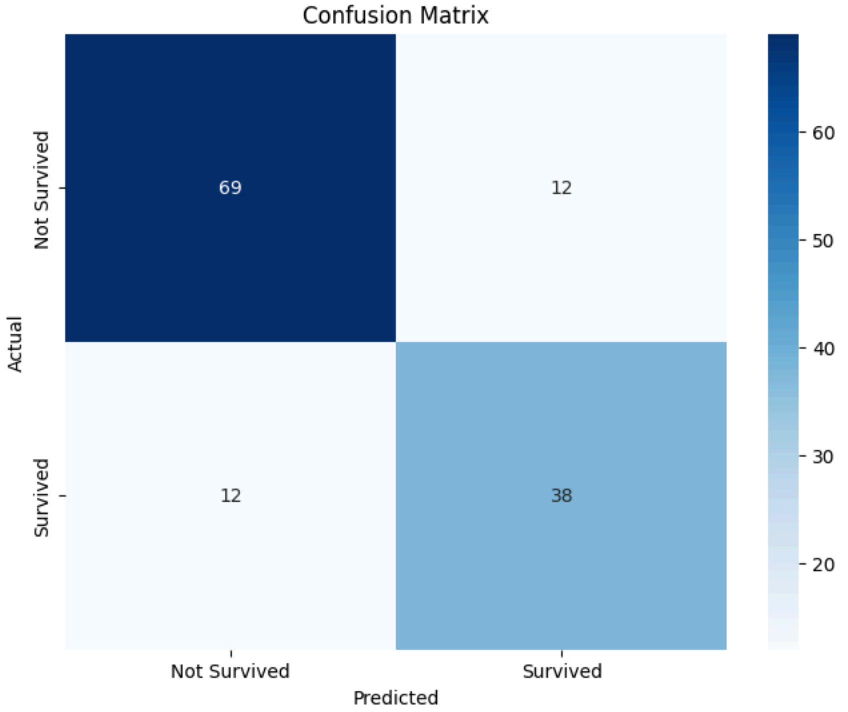
	pclass	age	sibsp	parch	sex_male	embarked_S
829	3	-1.038419	5.039298	1.830957	0.0	1.0
889	3	-0.268845	-0.509866	-0.428338	1.0	1.0
330	2	2.116833	-0.509866	-0.428338	1.0	1.0
91	1	0.115942	0.599967	-0.428338	1.0	1.0
808	3	-0.267124	-0.509866	-0.428338	1.0	1.0

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report


# logistic regression
logreg_model = LogisticRegression(penalty='l1', solver='liblinear', max_iter=1000, random_state=2024)

logreg_model.fit(X_train, y_train)

y_pred = logreg_model.predict(X_valid)


print(f'Predictions on the validation set: {y_pred}')

accuracy = accuracy_score(y_valid, y_pred)
print(f'Accuracy on the validation set: {accuracy:.4f}')
```

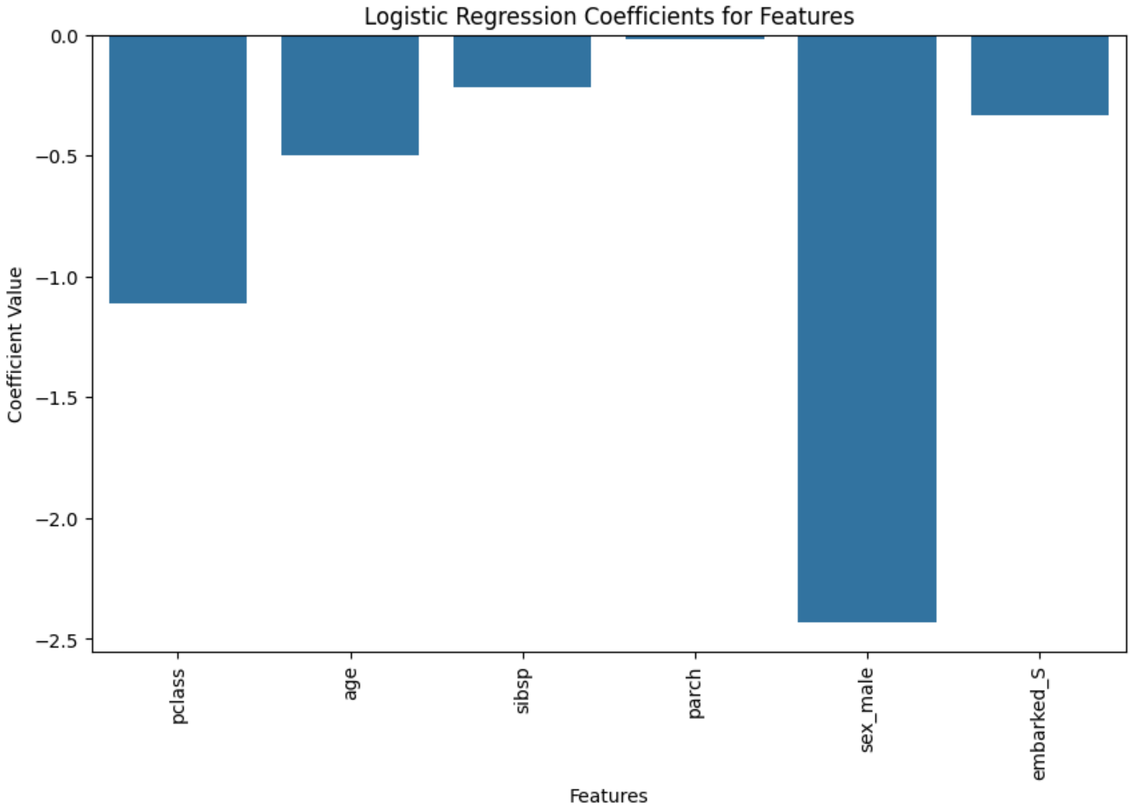
```
Predictions on the validation set: [1 0 0 0 0 0 1 1 1 0 0 1 1 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 1
 1 0 1 0 0 0 0 1 1 0 1 1 1 1 1 0 1 1 1 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 0 1 1
 1 0 1 1 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1
 0 1 0 1 0 0 1 1 1 0 1 0 1 0 0 0 0 1 0 0]
Accuracy on the validation set: 0.8168
```

```
accuracy = accuracy_score(y_valid, y_pred)
print(f'Accuracy on the validation set: {accuracy:.4f}')

# confusion matrix and coefficient values
conf_matrix = confusion_matrix(y_valid, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=['Not Survived', 'Survived'], yticklabels=['Not Survived', 'Surv
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

class_report = classification_report(y_valid, y_pred)
print(f'Classification Report:\n{class_report}')
coefficients = logreg_model.coef_[0]
features = X_train.columns

plt.figure(figsize=(10, 6))
sns.barplot(x=features, y=coefficients)
plt.title('Logistic Regression Coefficients for Features')
plt.xticks(rotation=90)
plt.xlabel('Features')
plt.ylabel('Coefficient Value')
plt.show()

print("Done")
```

```
Accuracy on the validation set: 0.8168
```




```
Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.85      0.85        81
           1       0.76      0.76      0.76        50

    accuracy                           0.82       131
   macro avg       0.81      0.81      0.81       131
weighted avg       0.82      0.82      0.82       131
```




```
Done
```

Finally we train the model and get a decent acuracy of 81.68%.

Conclusion

Steps:

1. Data Loading & Exploration: I loaded the Titanic dataset, performed EDA, and visualized key relationships between variables like age, sex, and survival.
2. Managing Missing Values: I handled missing age values by filling them with the mean or searching in internet.
3. Encoding Categorical Variables: I used OneHotEncoder to encode categorical variables like sex and embarked.
4. Feature Scaling: I applied both StandardScaler to standardize numerical features for better model performance.
5. Data Splitting: The data was split into training, validation, and test sets: 80, 10, 10.
6. Addressing Class Imbalance: SMOTE was used to oversample the minority class and address class imbalance.
7. Feature Selection: I removed low variance and highly correlated features, such as important_title and fare, to improve model performance.
8. Model Training: Logistic Regression model was trained on the processed data.

Observations

- Data leakage is hard, specially managing the order of the pipeline
- Creating new features doesnt always mean better performance

LOGs

Run 1: 0.82

- Variable: pclass, age, sibsp, parch, sex_male, embarked_S
- Variance threshold: 0.2
- Correlation threshold: 0.5
- ADASYN + Standarization

Run 2: 0.8015

- Variable: pclass, age, sibsp, parch, sex_male, embarked_S, alone, fare
- Variance threshold: 0.1
- Correlation threshold: 0.7
- ADASYN + Standarization

Run 3: 0.8321

- Variable: pclass, sibsp, sex_male, alone
- Variance threshold: 0.1
- Correlation threshold: 0.7
- ADASYN + Standarization