Katherine Stanton
Student ID: W210361913

For the Module 2 Lab Assignment, I worked with the Titanic dataset from Kaggle. For guidance on set-up, I refered to Medium's article by Sanjay Dutta; "Tackling the Titanic Dataset with Machine Learning (Kaggle Challenge!)"

Environment Description: I began by creating a new environment in my Anaconda Navigator, installing packages using Terminal. I then launched a Jupyter Notebook with the new environment, which was loaded with scikit-learn, pandas, numpy, mglearn, matplotlib, and seaborn. Next, I downloaded the two Titanic dataset files from Kaggle to my computer; train.csv and test.csv. Then using pd.read_csv I loaded the data to my notebook.

```
# Loading the training and test datasets
train_df = pd.read_csv('/Users/ktgraze/Documents/titanic/train.csv')
test_df = pd.read_csv('/Users/ktgraze/Documents/titanic/test.csv')
```

Next, I explored the data using various visualization methods with matplotlib.pyplot and seaborn. I cleaned the data by filling in missing values for 'Age', 'Embarked', and 'Fare'. Then, I converted categorical features to numerical and dropped irrelevant features ('Name', 'Ticket', and 'Cabin') that would not lend insight towards predicting Survival amongst passengers.

My next step was feature engineering by combining 'SibSp' and 'Parch' into a new feature 'FamilySize'.

Feature Engineering

```
# Create a new feature "FamilySize' from 'SibSp' and 'Parch'
train_df['FamilySize'] = train_df['SibSp'] + train_df['Parch'] + 1
test_df['FamilySize'] = test_df['SibSp'] + test_df['Parch'] + 1

train_df['IsAlone'] = np.where(train_df['FamilySize'] > 1, 0, 1)
test_df['IsAlone'] = np.where(test_df['FamilySize'] > 1, 0, 1)
```

Next I used *train_test_split* on the train_df dataframe and deployed the first supervised algorithm. The two supervised learning algorithms I chose from were *RandomForestClassifier* and the Gradient Boosting Machine: *XGBoost*. Metrics used to evaluate the models were accuracy and cross-validation.

For the *RandomForestClassifier* I split the data into training and validation sets. This resulted in an 82% accuracy on the validation set. I next used *GridSearchCV* to find the best parameters and to then make predictions using those parameters.

```
Best parameters found by grid search: {'criterion': 'gini', 'max_depth': 4, 'max_features': 'sqr
t', 'n_estimators': 200}
Validation set accuracy after tuning: 0.80
```

I next chose to use *XGBoost* a Gradient Boosting Machine and converted the data to DMatrix format. After training the model and making predictions on the validation set it returned an accuracy of 81.56%. I then performed StratifiedKFold to cross-validate the scores, with a mean accuracy of 83.27%, shown below.

```python
# Performing cross-validation
from sklearn.model_selection import cross_val_score, StratifiedKFold
from xgboost import XGBClassifier

# Initialize XGBClassifier with the parameters
model = XGBClassifier(**params)

# Define cross-validation strategy
cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

# Perform cross-validation
cv_scores = cross_val_score(model, X, y, cv=cv, scoring='accuracy')

# Print cross-validation scores
print(f'Cross-validation scores: {cv_scores}')
print(f'Mean accuracy: {np.mean(cv_scores):.4f} +/- {np.std(cv_scores):.4f}')
```

```
Cross-validation scores: [0.86592179 0.83146067 0.80337079 0.81460674 0.84831461]
Mean accuracy: 0.8327 +/- 0.0225
```

The cross-validation of XGBoost was the most successful metric in predicting 'Survived' for the dataset.

Learning Experience: This is only my third attempt at coding, using machine learning methods and initializing different supervised learning algorithms. This was my first time working with two different dataframes at once, which was more challenging than I'm accustomed to at this point in my learning. I was introduced to XGBoost, which returned the most accurate scores and I would like to try to fine-tune the parameters of the model more in the future to increase its accuracy.

Key Takeaways: Initially exploring the Titanic dataset, based on my knowledge of the historical event, the data was sadly not shocking to see that the relationship between Passenger Class and Survival rates were expected (the wealthier you were the higher chance at survival). As I am still a beginner, the visualizations helped the most in explaining the data and choosing my next step based on feature importance. I would like to revisit this notebook in the future to compare my learning journey and hope that it seems rudimentary in comparison to future skill.