

A dramatic, dark blue-toned image of the Titanic shipwreck. The ship is partially submerged, with its bow and upper decks visible above the water. A bright light emanates from the ship's interior, and a jagged lightning bolt strikes the dark, stormy sky above. The water is dark and choppy, reflecting the light from the ship and the sky.

Titanic Survival Prediction

An Lu, Jose Vela, Jim Lao

Tools

- SQL
- Postgres
- Pandas
- Python
- PGAdmin
- Streamlit

Loading in the Data

	passengerid integer	survived integer	pclass integer	name character varying (100)
1	1	0	3	Braund, Mr. Owen Harris
2	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)
3	3	1	3	Heikkinen, Miss. Laina
4	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)
5	5	0	3	Allen, Mr. William Henry
6	6	0	3	Moran, Mr. James
7	7	0	1	McCarthy, Mr. Timothy J
8	8	0	3	Palsson, Master. Gosta Leonard
9	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)
10	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)
11	11	1	3	Sandstrom, Miss. Marguerite Rut
12	12	1	1	Bonnell, Miss. Elizabeth
13	13	0	3	Saunderscock, Mr. William Henry
14	14	0	3	Andersson, Mr. Anders Johan
15	15	0	3	Vestrom, Miss. Hulda Amanda Adolfina
16	16	1	2	Hewlett, Mrs. (Mary D Kingcome)
17	17	0	3	Rice, Master. Eugene
18	18	1	2	Williams, Mr. Charles Eugene
19	19	0	3	Vander Planke, Mrs. Julius (Emelia Maria Vandemoortele)
20	20	1	3	Masselmani, Mrs. Fatima
21	21	0	2	Fynney, Mr. Joseph J
22	22	1	2	Beesley, Mr. Lawrence

```
DROP TABLE IF EXISTS titanic_Test;  
DROP TABLE IF EXISTS titanic_Train;
```

```
CREATE TABLE titanic_Train(  
  PassengerId int,  
  Survived int,  
  Pclass int,  
  Name VARCHAR (100),  
  Sex VARCHAR (100),  
  Age float,  
  SibSp int,  
  Parch int,  
  Ticket Varchar (100),  
  Fare float,  
  Cabin varchar (100),  
  Embarked varchar (100));
```

```
CREATE TABLE titanic_Test(  
  PassengerId int,  
  Pclass int,  
  Name VARCHAR (100),  
  Sex VARCHAR (100),  
  Age float,  
  SibSp int,  
  Parch int,  
  Ticket Varchar (100),  
  Fare float,  
  Cabin varchar (100),  
  Embarked varchar (100))
```

	passengerid integer	pclass integer	name character varying (100)	sex character varying (10)
1	892	3	Kelly, Mr. James	male
2	893	3	Wilkes, Mrs. James (Ellen Needs)	female
3	894	2	Myles, Mr. Thomas Francis	male
4	895	3	Wirz, Mr. Albert	male
5	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female
6	897	3	Svensson, Mr. Johan Cervin	male
7	898	3	Connolly, Miss. Kate	female
8	899	2	Caldwell, Mr. Albert Francis	male
9	900	3	Abraham, Mrs. Joseph (Sophie Halaut Easu)	female
10	901	3	Davies, Mr. John Samuel	male
11	902	3	Ilieff, Mr. Ylio	male
12	903	1	Jones, Mr. Charles Cresson	male
13	904	1	Snyder, Mrs. John Pillsbury (Nelle Stevenson)	female
14	905	2	Howard, Mr. Benjamin	male
15	906	1	Chaffee, Mrs. Herbert Fuller (Carrie Constance Toogood)	female
16	907	2	del Carlo, Mrs. Sebastiano (Argenia Genovesi)	female
17	908	2	Keane, Mr. Daniel	male
18	909	3	Assaf, Mr. Gerios	male
19	910	3	Ilmakangas, Miss. Ida Livija	female
20	911	3	Assaf Khalil, Mrs. Mariana (Miriam)	female
21	912	1	Rothschild, Mr. Martin	male
22	913	3	Olsen, Master. Artur Karl	male

Loading in the Data

```
[3]: # Define SQL query for full table
query = 'SELECT * FROM titanic_train'
# Read the data into a pandas DataFrame
titanic_train_df = pd.read_sql(query, engine)
# Display the DataFrame
titanic_train_df.head()
```

```
[3]:
```

	passengerid	survived	pclass	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	None	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	None	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	None	S

Data Cleanup

[3]:

	passengerid	survived	pclass	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	None	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	None	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	None	S

[4]:

titanic_train_df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
Column Non-Null Count Dtype
--- -
0 passengerid 891 non-null int64
1 survived 891 non-null int64
2 pclass 891 non-null int64
3 name 891 non-null object
4 sex 891 non-null object
5 age 714 non-null float64
6 sibsp 891 non-null int64
7 parch 891 non-null int64
8 ticket 891 non-null object
9 fare 891 non-null float64
10 cabin 204 non-null object
11 embarked 889 non-null object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB

[5]:

titanic_train_df.loc[pd.isna(titanic_train_df['age']), 'age']

=39.2

[6]:

titanic_train_df.head(6)

[6]:

	passengerid	survived	pclass	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	None	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	None	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	None	S
5	6	0	3	Moran, Mr. James	male	39.2	0	0	330877	8.4583	None	Q

Data Cleanup

```
[7]: titanic_train_df.drop('cabin', axis=1, inplace=True)
titanic_train_df.head()
```

```
[7]:
```

	passengerid	survived	pclass	name	sex	age	sibsp	parch	ticket	fare	embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	S

```
[8]: # Define SQL query for full table
query = 'SELECT * FROM titanic_test'
# Read the data into a pandas DataFrame
titanic_test_df = pd.read_sql(query, engine)
# Display the DataFrame
titanic_test_df.head()
```

```
[8]:
```

	passengerid	pclass	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	None	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	None	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	None	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	None	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	None	S

```
[9]: titanic_test_df.loc[pd.isna(titanic_test_df['age']), 'age'] = 39.2
```

```
[10]: titanic_test_df.head(6)
```

```
[10]:
```

	passengerid	pclass	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	None	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	None	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	None	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	None	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	None	S

Deep Learning Model

Define the deep learning model

```
nn_model = tf.keras.models.Sequential([  
    tf.keras.layers.Dense(units=16, activation="relu", input_shape=(X_train_scaled.shape[1],)),  
    tf.keras.layers.Dense(units=1, activation="sigmoid")])
```

Compile the Sequential model together and customize metrics

```
nn_model.compile(loss="binary_crossentropy", optimizer="adam", metrics=["accuracy"])
```

Train the model

```
fit_model = nn_model.fit(X_train_scaled, y_train, epochs=100)
```

Evaluate the model using the test data

```
model_loss, model_accuracy = nn_model.evaluate(X_test_scaled, y_test, verbose=2)  
print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")
```


Deep Learning Model

21/21 ————— 0s 6ms/step - accuracy: 0.7460 - loss: 0.5214
7/7 - 0s - 29ms/step - accuracy: 0.6906 - loss: 0.5974
Loss: 0.5974016189575195, Accuracy: 0.6905829310417175

```
In [17]: # Evaluate the model using the test data
model_loss, model_accuracy = nn_model.evaluate(X_train_scaled, y_train, verbose=2)
print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")
```

21/21 - 0s - 3ms/step - accuracy: 0.7575 - loss: 0.5202
Loss: 0.5202223658561707, Accuracy: 0.757485032081604

```
In [18]: nn_model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 16)	112
dense_1 (Dense)	(None, 1)	17

Total params: 389 (1.52 KB)

Trainable params: 129 (516.00 B)

Non-trainable params: 0 (0.00 B)

Optimizer params: 260 (1.02 KB)

Deep Learning Models

```
[16]: # Evaluate the model using the test data
model_loss, model_accuracy = nn_model.evaluate(X_train_scaled, y_train, verbose=2)
print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")

21/21 - 0s - 2ms/step - accuracy: 0.7440 - loss: 0.5296
Loss: 0.5295883417129517, Accuracy: 0.7440119981765747
```

```
[17]: nn_model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 16)	112
dense_1 (Dense)	(None, 1)	17

Total params: 389 (1.52 KB)

Trainable params: 129 (516.00 B)

Non-trainable params: 0 (0.00 B)

Optimizer params: 260 (1.02 KB)

```
[21]: #Random Forrest
from sklearn.ensemble import RandomForestClassifier

randomforest = RandomForestClassifier()
randomforest.fit(X_train_scaled, y_train)
y_pred = randomforest.predict(X_test)
acc_randomforest = round(accuracy_score(y_pred, y_test) * 100, 2)
print(acc_randomforest)
```

Deep Learning Models

- We used a Random Forest model

```
[21]: #Random Forrest
      from sklearn.ensemble import RandomForestClassifier

      randomforest = RandomForestClassifier()
      randomforest.fit(X_train_scaled, y_train)
      y_pred = randomforest.predict(X_test)
      acc_randomforest = round(accuracy_score(y_pred, y_test) * 100, 2)
      print(acc_randomforest)
```

64.57

```
[26]: # Move the decimal place two spots to the right for model_accuracy
      adjusted_model_accuracy = model_accuracy * 100

      models = pd.DataFrame({
          'Model': ['Neural Network Model', 'Random Forest'],
          'Score': [adjusted_model_accuracy, acc_randomforest]
      })

      sorted_models = models.sort_values(by='Score', ascending=False)
      sorted_models
```

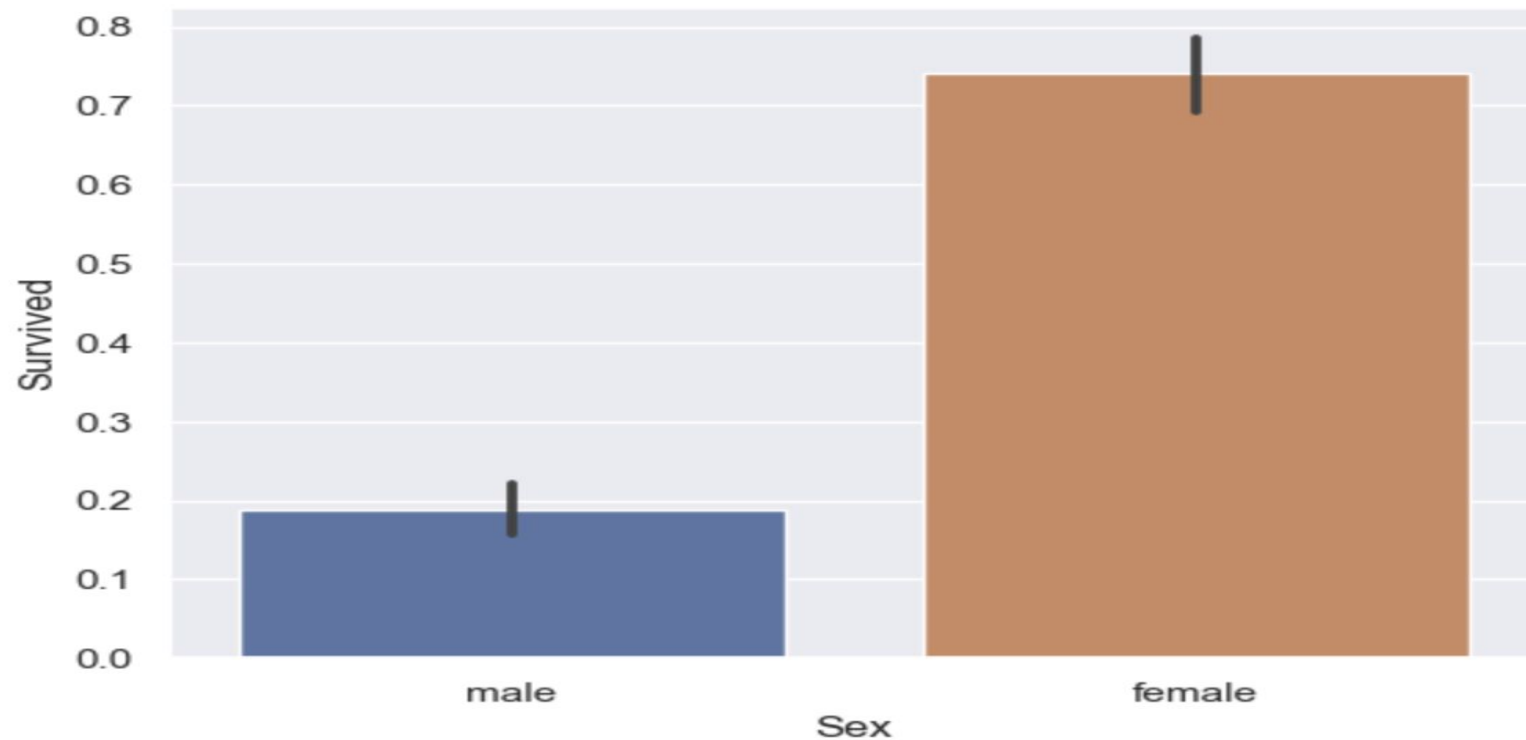
```
[26]:
```

	Model	Score
0	Neural Network Model	74.4012
1	Random Forest	64.5700

Predictions

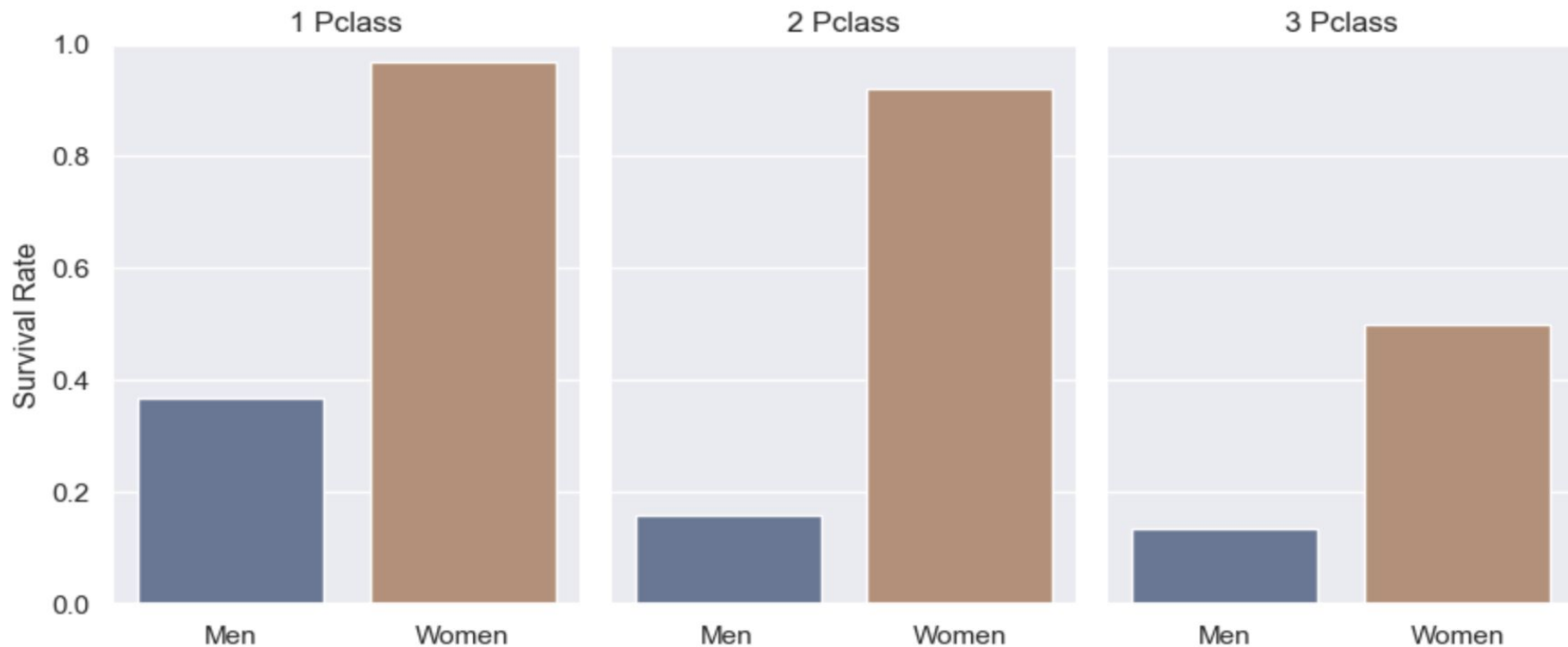
Percentage of females who survived: 74.03846153846155

Percentage of males who survived: 18.890814558058924



Predictions

How many Men and Women Survived by Passenger Class



Interface Predictions



Resources

<https://www.kaggle.com/competitions/titanic>