PROJECT REPORT

Exploratory Data Analysis:

Data preparation

We started by loading the data from both file ‘titanic.csv’ and ‘test.csv’ for the purpose of concatenating the data for easier preprocessing. We encountered two issues, The first was a typo in the column name 'sibSp'. We fixed it by renaming the column to ‘sibsp’. The second was that the data in the test file was missing the target column ‘survived’. So, we added dummy values for the column and then removed them after preprocessing was done. We printed the number of rows from each file, So, we can accurately separate them after preprocessing.

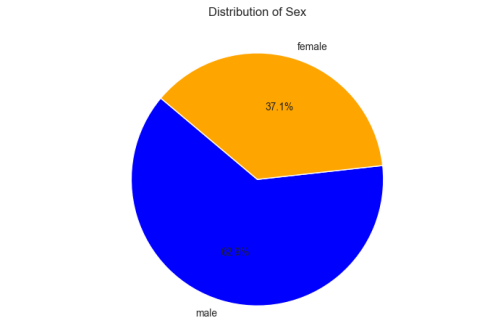
	count	mean	std	min	25%	50%	75%	max
pclass	1328.000000	2.296687	0.836753	1.000000	2.000000	3.000000	3.000000	3.000000
age	1064.000000	29.927788	14.413728	0.166700	21.000000	28.000000	39.000000	80.000000
sibsp	1328.000000	0.499247	1.036631	0.000000	0.000000	0.000000	1.000000	8.000000
parch	1328.000000	0.381024	0.860768	0.000000	0.000000	0.000000	0.000000	9.000000
fare	1327.000000	33.106063	51.464510	0.000000	7.895800	14.454200	31.137500	512.329200
survived	1328.000000	0.376506	0.484692	0.000000	0.000000	0.000000	1.000000	1.000000

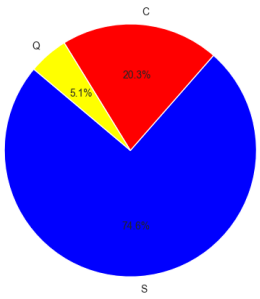
Handling missing values

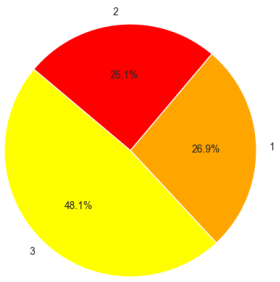
We started our analysis by printing a statistical summary of the data features. We calculated the percentage of missing values in each column. Four columns were found to have missing values, ‘age’, ‘fare’, ‘cabin’, ‘embarked’. For the ‘fare’ and ‘embarked’ columns, the percentage was less than 1%. So, we decided to handle them by removing the rows with missing values. For the ‘cabin’ column, we found that more than 75% of the data was missing. So, there was really nothing we could do about it, as trying to generate the missing data would make it biased. So, we decided to handle the issue by dropping the column. Finally, for the ‘age’ column, we found 20% of the data missing. We had to make a choice between losing 20% of the data or losing potentially valuable information. So, we decided to test our models on both scenarios, and we found that our models yielded higher performance with the ‘age’ column than without it.

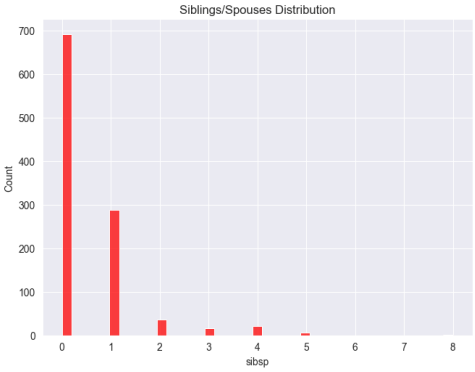
Feature visualization

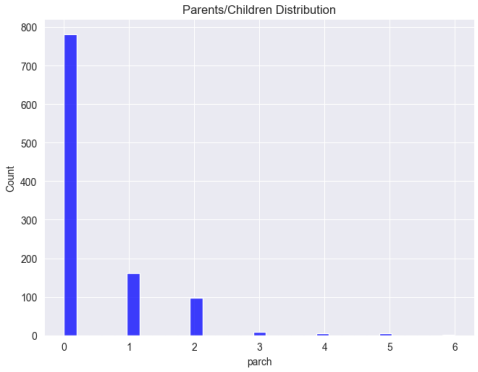
We moved on to visualizing features to better understand them. For features with discrete values (e.g. ‘sex’, ‘embarked’, ‘pclass’), we used pie charts. For features with continuous values (e.g. ‘sibsp’, ‘parch’, ‘fare’), we used histograms.

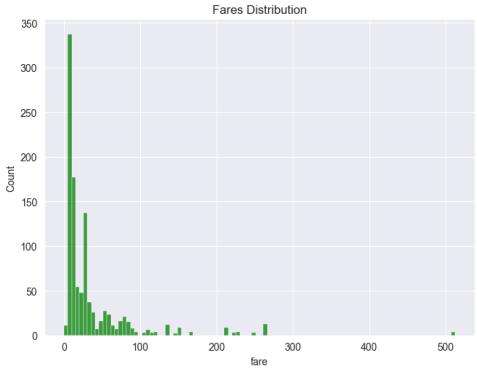






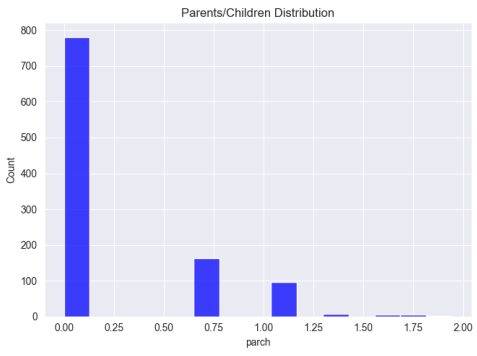


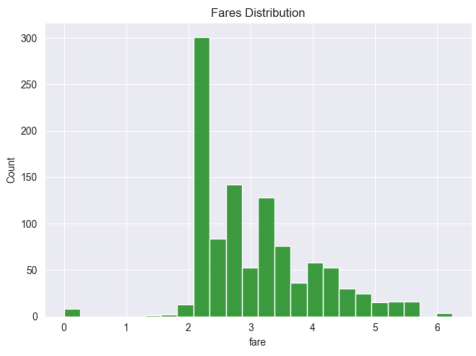




Feature Transformation

After visualizing the features, we proceeded to encode features with discrete values using one-hot encoding. We noticed that some of the features with continuous values were skewed. So, we applied log transformation to these features to make them more normalized. Finally, we used Min-Max technique to scale our features to simplify the training process.





Handling Outliers

We checked for outliers in the data set and handled them by calculating the inter-quartile range and eliminating datapoints outside that range improving performance.

Model Training

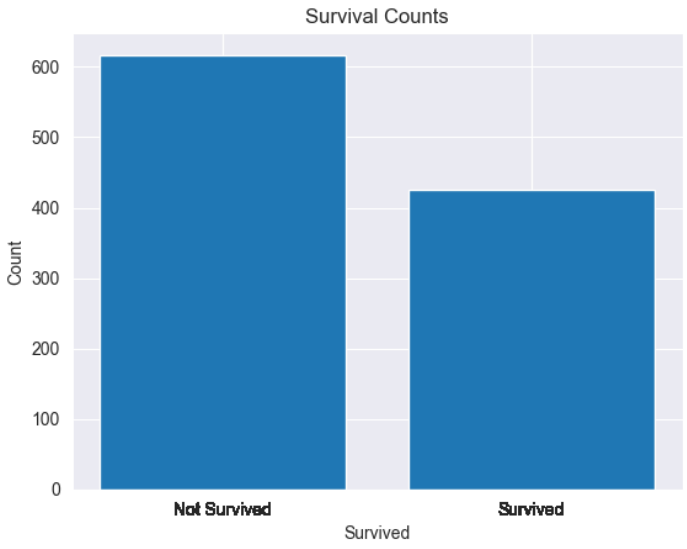
We separated the data from the 'test.csv' file from the rest of the data and saved it in a new dataframe 'test_set'. We used the 'train_test_split' function to split the dataframe into 'X_train', 'X_test', 'y_train', and 'y_test'.

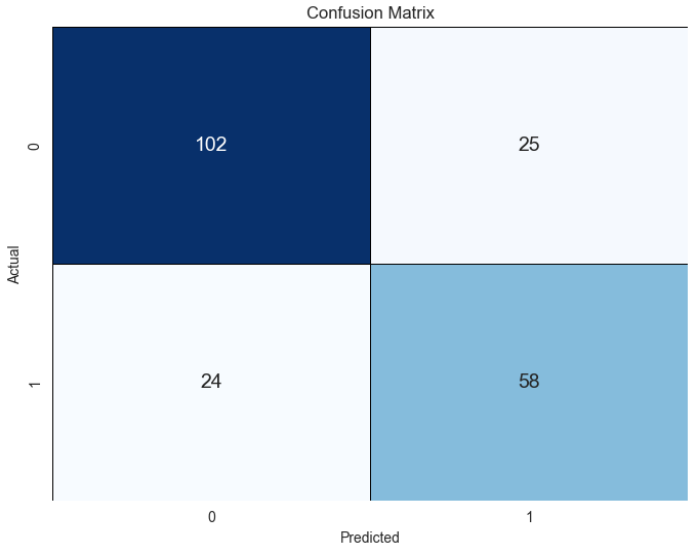
	pclass	name	age	sibsp	parch	ticket	fare	survived	sex_female	sex_male	embarked_C	embarked_Q	embarked_S
0	1	Allen, Miss. Elisabeth Walton	29.000000	0	0	24160	211.337500	1.000000	True	False	False	False	True
1	1	Allison, Master. Hudson Trevor	0.916700	1	2	113781	151.550000	1.000000	False	True	False	False	True
2	1	Allison, Miss. Helen Loraine	2.000000	1	2	113781	151.550000	0.000000	True	False	False	False	True
3	1	Allison, Mr. Hudson Joshua Creighton	30.000000	1	2	113781	151.550000	0.000000	False	True	False	False	True
4	1	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	25.000000	1	2	113781	151.550000	0.000000	True	False	False	False	True

Naïve Bayes

We used a bar plot to visualize the ratio between the passengers who survived and those who didn't. This allowed us to check whether the classes were balanced or not.

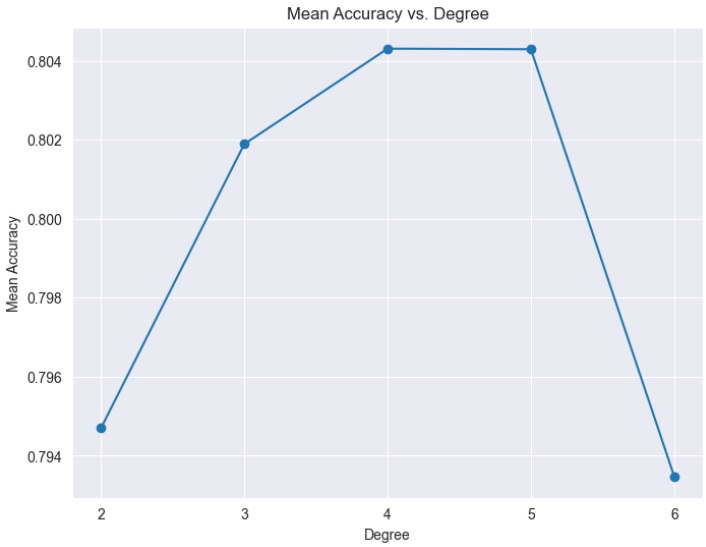
We then proceeded to train a Gaussian Naïve Bayes model on the training set. We used cross-validation to calculate model accuracy. We plotted the confusion matrix and printed a classification report with summary of different metrics (e.g. precision, recall, F1-Score). Finally, we printed the passengers' names along with their survival status predicted by the model.

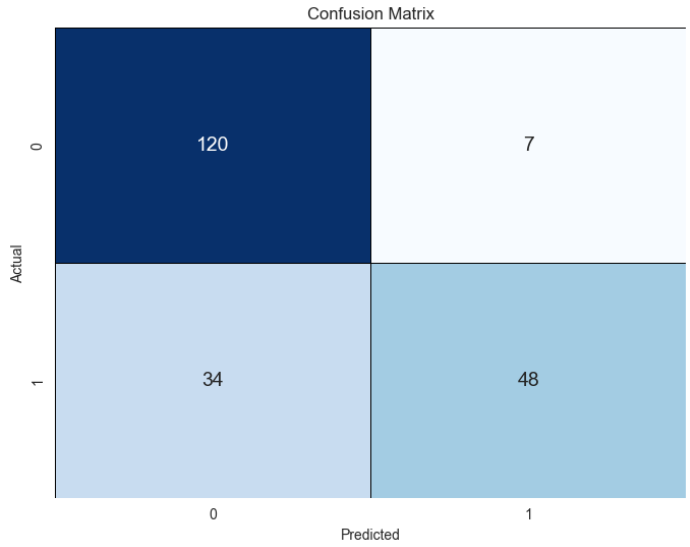




Support Vector Machine (SVM)

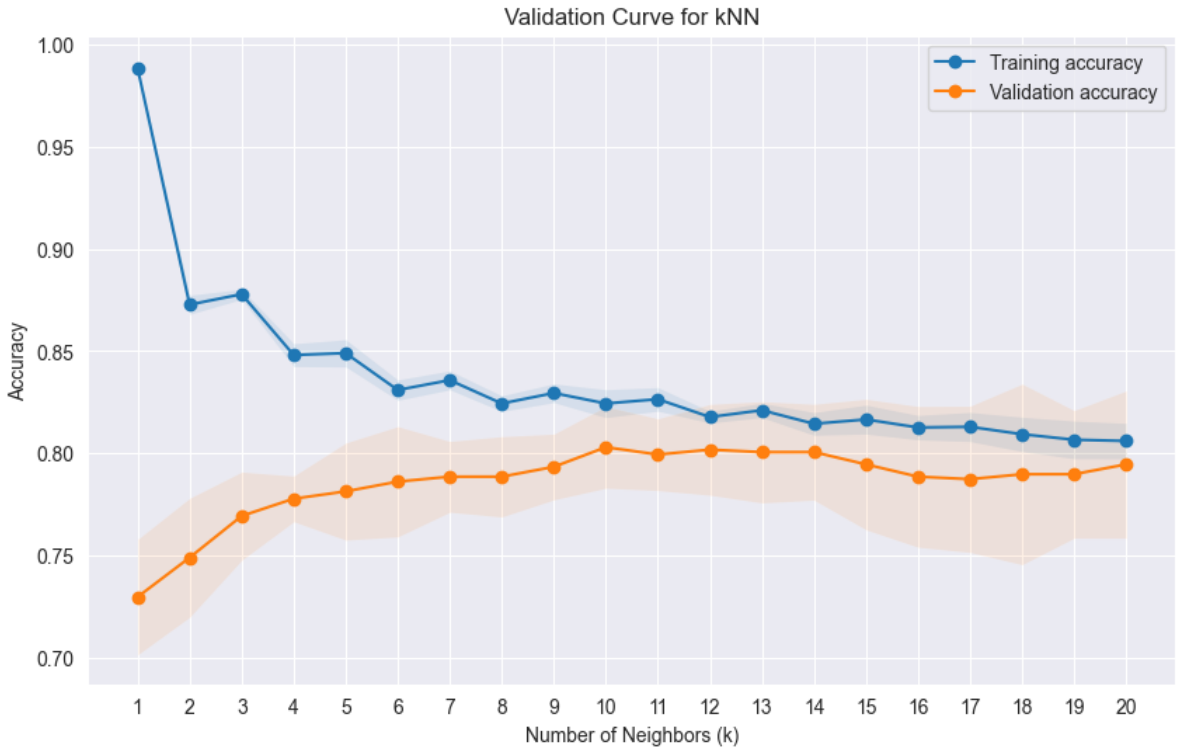
We used an SVM model with polynomial kernel to fit our data. We used a grid search to find the optimal model degree. We plotted a curve to visualize the relationship between model complexity and accuracy. We passed the optimal degree to the model and fitted it to the training data. We plotted the confusion matrix and printed the classification report. Finally, we printed the passengers' names along with their survival status predicted by both the first model and the new one.



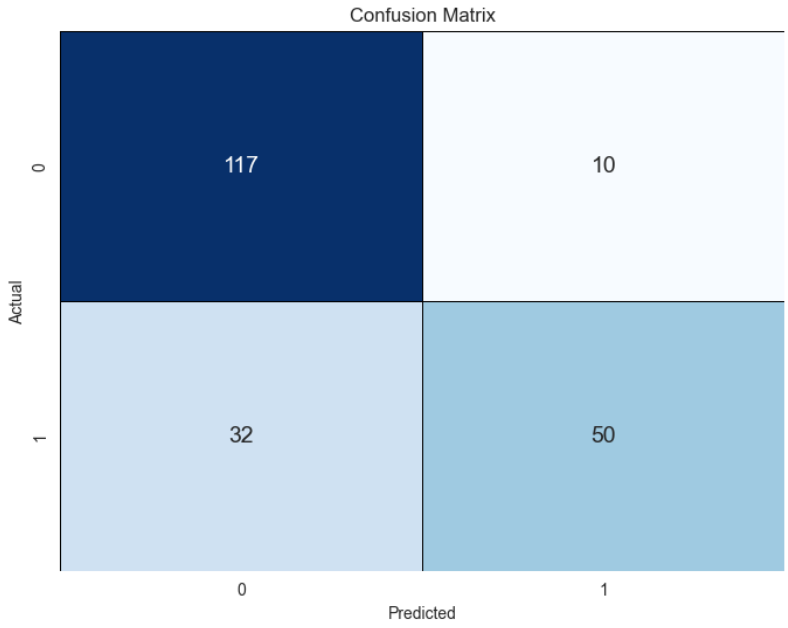


K-Nearest Neighbor (KNN)

We started by defining a range of values for K to try. We used a validation curve to visualize the relationship between the value of K and the model's accuracy. We fitted the model to the training data using the optimal value of K.



We performed cross-validation to check the model performance. We plotted the confusion matrix and printed the classification report. Finally, we printed the passengers' names along with their survival status predicted by the new model along with the predictions from previous models.



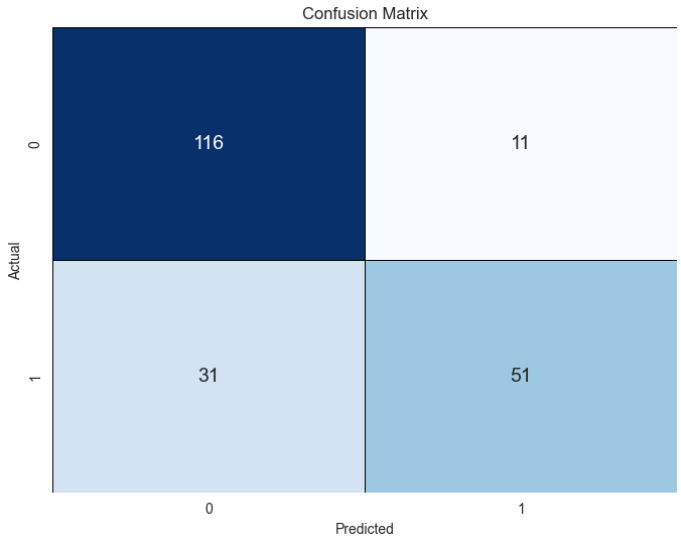
Artificial Neural Network (ANN)

We chose to use ‘tensorflow’ library, as it is the most advanced NN library worldwide nowadays. We started by converting our data to tensor format, which is the data type associated with tensorflow library. We then divided our data into batches to perform cross-validation. We defined the network architecture. We used a three-layer network with two hidden layers and an output layer with one unit. We used RELU activation for the hidden layers and Sigmoid activation for the output layer. As for the optimization algorithm, we chose to use ‘Adam optimizer’ instead of the usual ‘Gradient descend’ as it yields faster convergence. We used the binary cross-entropy loss function as it is usually associated with Sigmoid function for binary classification tasks.

After defining our network, we proceeded to fit our model to the training data and calculated the model’s accuracy.

```
Epoch 93/100
27/27 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8234 - loss: 0.4008 - val_accuracy: 0.7943 - val_loss: 0.5126
Epoch 94/100
27/27 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8273 - loss: 0.3850 - val_accuracy: 0.8086 - val_loss: 0.5251
Epoch 95/100
27/27 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8264 - loss: 0.3878 - val_accuracy: 0.7990 - val_loss: 0.5148
Epoch 96/100
27/27 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8233 - loss: 0.4139 - val_accuracy: 0.7990 - val_loss: 0.5148
Epoch 97/100
27/27 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8245 - loss: 0.4134 - val_accuracy: 0.8134 - val_loss: 0.5181
Epoch 98/100
27/27 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8162 - loss: 0.4202 - val_accuracy: 0.8038 - val_loss: 0.5108
Epoch 99/100
27/27 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8349 - loss: 0.3925 - val_accuracy: 0.8086 - val_loss: 0.5205
Epoch 100/100
27/27 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8435 - loss: 0.3723 - val_accuracy: 0.7990 - val_loss: 0.5162
7/7 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8208 - loss: 0.4691
Test Accuracy: 0.7990430593490601
7/7 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step
```

We then plotted the confusion matrix and printed the classification report. Finally, we printed the passengers' names along with their survival status predicted by the new model along with the predictions from previous models.



This is the final chart with passenger names and the predicted survival status from the 4 models.

	name	nb_survived	svm_survived	knn_survived	ann_survived
1308	Zimmerman, Mr. Leo	0	0	0	0
1309	Kelly, Mr. James	0	0	0	0
1310	Wilkes, Mrs. James (Ellen Needs)	1	0	0	0
1311	Myles, Mr. Thomas Francis	0	0	0	0
1312	Wirz, Mr. Albert	0	0	0	0
1313	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	1	0	1	0
1314	Svensson, Mr. Johan Cervin	0	0	0	0
1315	Connolly, Miss. Kate	0	0	0	0
1316	Caldwell, Mr. Albert Francis	0	0	0	0
1317	Abrahim, Mrs. Joseph (Sophie Halaut Easu)	1	1	1	1
1318	Davies, Mr. John Samuel	0	0	0	0
1320	Jones, Mr. Charles Cresson	0	0	0	0
1321	Snyder, Mrs. John Pillsbury (Nelle Stevenson)	1	1	1	1
1322	Howard, Mr. Benjamin	0	0	0	0
1323	Chaffee, Mrs. Herbert Fuller (Carrie Constance Toogood)	1	1	1	1
1324	del Carlo, Mrs. Sebastiano (Argenia Genovesi)	1	1	1	1
1325	Keane, Mr. Daniel	0	0	0	0
1326	Assaf, Mr. Gerios	0	0	0	0
1327	Ilmakangas, Miss. Ida Livija	1	0	0	0

Last but not least, we created a chart to summarize the performance metrics results of all models for easier comparison.

	Model	Accuracy	Precision	Recall	F1-score
0	Naive Bayes	75.790000	67.610000	67.610000	67.610000
1	SVM	80.530000	80.360000	63.380000	70.870000
2	KNN	82.110000	83.640000	64.790000	73.020000
3	ANN	82.630000	86.540000	63.380000	73.170000

Also worth mentioning that throughout the entire project we relied on version control systems and platforms like Git and GitHub to organize our work and keep track of tasks and issues that needed to be addressed.

CREDITS:

Fares Mohamed Salah **ID:** 22011614

Kareem Ashraf Ahmed **ID:** 22060171

Mahmoud Ahmed Bahig **ID:** 2203165

Abdelrahman Ahmed El-Motawakel **ID:** 22010456

Our project repo can be found at: https://github.com/faresmohamed260/School-Work/tree/main/Machine%20Learning/Final%20Project

THANKS 😊