PREDICTING TITANIC SURVIVORS

June 23, 2021

Francisco Manuel Pires Gonçalves

Departamento de Ciência de Computadores
Faculdade de Ciências de Universidade do Porto

Contents

Introduction

For this practical assignment of the second part of the course of Advanced Topics in Artificial Intelligence, I decided to combine previous experience in the area of Data Mining with the newly acquired knowledge in the area of Artificial Intelligence to base this project on accurately predicting the survivors from a dataset containing the passengers of the Titanic. Such a topic came to me when looking through different ideas and topics through which machine learning techniques could be applied. This was the chosen dataset as it seemed like quite an interesting topic to develop.

The initial work of finding a suitable dataset was cut short as the dataset from where I got this idea already provided a link to a Kaggle competition called *Titanic - Machine Learning from Disaster* which contained an initial dataset as well as instructions of its contents.

Then, the remaining work consisted of pre-processing tasks such as removing unnecessary variables from the dataset and finding solutions for any NaN values. Afterwards, the different prediction models are applied to a machine learning model in order to figure out which model obtains more accurate predictions.

Considering this dataset also comes from an open Kaggle competition I have also decided to submit an entry using the best prediction model on the test dataset which is also provided.

Further discussion on each prediction model has been included in this report.

Materials and Methods

With the dataset chosen and objectives defined, analysing the dataset became the first priority. The training dataset contains the record of 891 passengers and 12 different variables that describe each of them. SOme of these had no impact on the predictions and were later removed. Here is some basic information regarding each feature:

- **PassengerId:** Integer to distinguish each passenger
- **Survived:** Integer that distinguishes if the passenger survived (1) or not (0).
- **Pclass:** Class of the passenger divided into integers. 1 is the wealthiest class and 3 is the poorest.
- **Name:** Full name of the passenger.
- **Sex:** Gender of the passenger (male or female).
- **Age:** Float of the age of the passenger.
- **SibSp:** Number of siblings/spouses on board for each passenger.
- **Parch:** Number of parents/children on board for each passenger.
- **Ticket:** Ticket number.
- **Fare:** Passenger fare.
- **Cabin:** Number of cabin where passenger stayed.
- **Embarked:** Port where passenger embarked (C for Cherbourg, Q for Queenstown, S for Southampton)

A lot of interesting data exploration steps could be performed with this data to answer quite pertinent questions regarding the passengers aboard the Titanic. However, for the task at hand this step was skipped as it wouldn't contribute much towards the final goal of this project.

To better understand which features are important and which can be dismissed, a correlation heatmap was created. Those that show higher values of correlation with the 'Survived' variable are the ones that will influence the prediction models the most when it comes to decide who would live and who would die.

Figure 1: Correlation Plot between survivability and other features

Although the variable 'Sex' appears to show the most outstanding correlation value from the bunch, that is due to the fact that there are only two possible options (male or female) which doesn't make up much of a difference when trying to predict who would survive. Obviously, in this scenario of the Titanic sinking, women and children were the priority which justifies the value shown here. However, the next most significant value is the one that really matters and that is 'Fare' which related to the price of the ticket for each passenger. Comparatively, the 'Pclass' feature would show similar results but with less detail which would result in less accuracy when using the prediction models and that is due to the fact that it could only differentiate between three different classes which will never be as precise. On the other hand, the fare of a passenger would directly relate to their wealth, which would imply higher survival chances. To reach this conclusion one would only need to watch the 1997 movie.

As for the remaining variables shown in the plot, they will also be kept as they will never hurt the prediction model's accuracy despite having very little impact. Irrelevant features such as 'Name', 'PassengerId', 'Cabin' and 'Ticket' can be removed as they would only impact the predictions in a negative matter.

After eliminating some unnecessary columns from the dataset, there were still some pesky NA values that needed to be take care of in the most suitable way. How they were handled varied according to each variable but no row was removed to preserve the original state of the dataset.

Afterwards, the last pre-processing step was to convert categorical variables to numerical, which was the case for the 'Sex' and the 'Embarked' features.

At last, the dataset was ready to be used.

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	3	0	22.0	1	0	7.2500	0
1	1	1	1	38.0	1	0	71.2833	1
2	1	3	1	26.0	0	0	7.9250	0
3	1	1	1	35.0	1	0	53.1000	0
4	0	3	0	35.0	0	0	8.0500	0
...	...	...	...	...	...	...	...	...
886	0	2	0	27.0	0	0	13.0000	0
887	1	1	1	19.0	0	0	30.0000	0
888	0	3	1	43.0	1	2	23.4500	0
889	1	1	0	26.0	0	0	30.0000	1
890	0	3	0	32.0	0	0	7.7500	2

891 rows × 8 columns

Figure 2: Training dataset after pre-processing steps

Results and Discussion

Continuing from the previous steps, building a machine learning model is the next step and an essential one as it defines how every prediction model used will work. An accurate and common split is 80% of the dataset for training and 20% for testing which is what was used for this assignment.

Following this procedure, the different prediction models are applied to the training set and then make predictions on the test set, obtaining the respective accuracy results.

Random Forests:

The accuracy obtained using the Random Forests Classification Model was 82.12%.

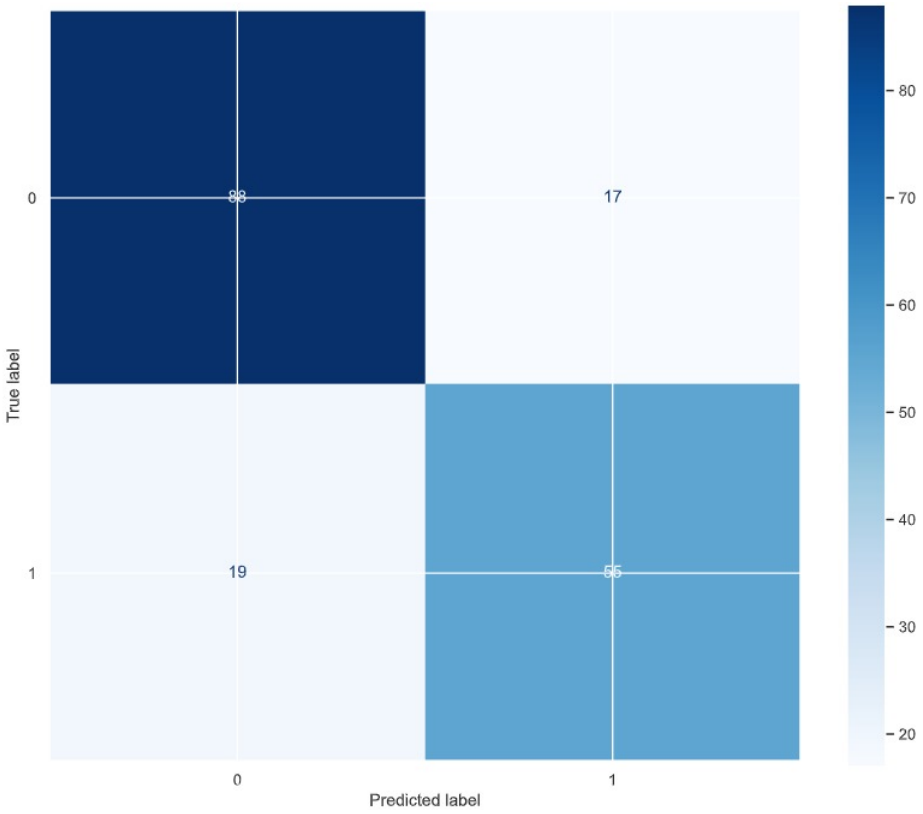

Figure 3: Confusion Matrix for Random Forests

Decision Trees:

The accuracy obtained using the Decision Trees Classifier was 79.33%.

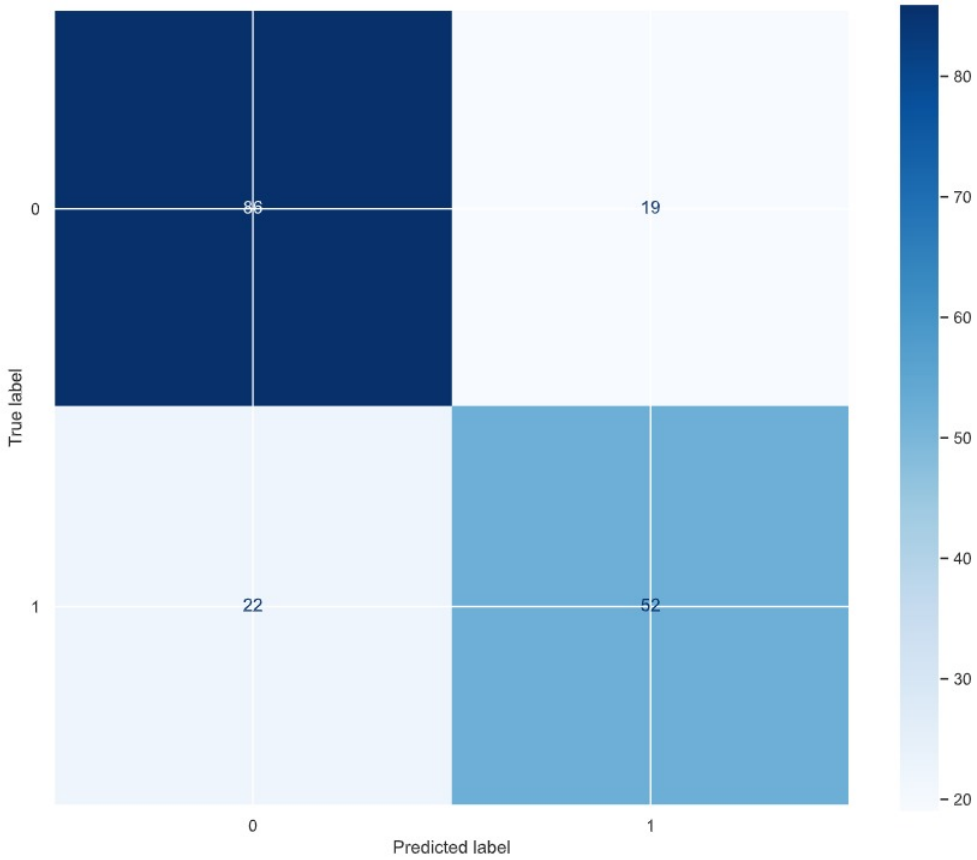

Figure 4: Confusion Matrix for Decision Trees

Logistic Regression:

The accuracy obtained using the Logistic Regression Classifier was 78.21% using the "lbfgs" solver.

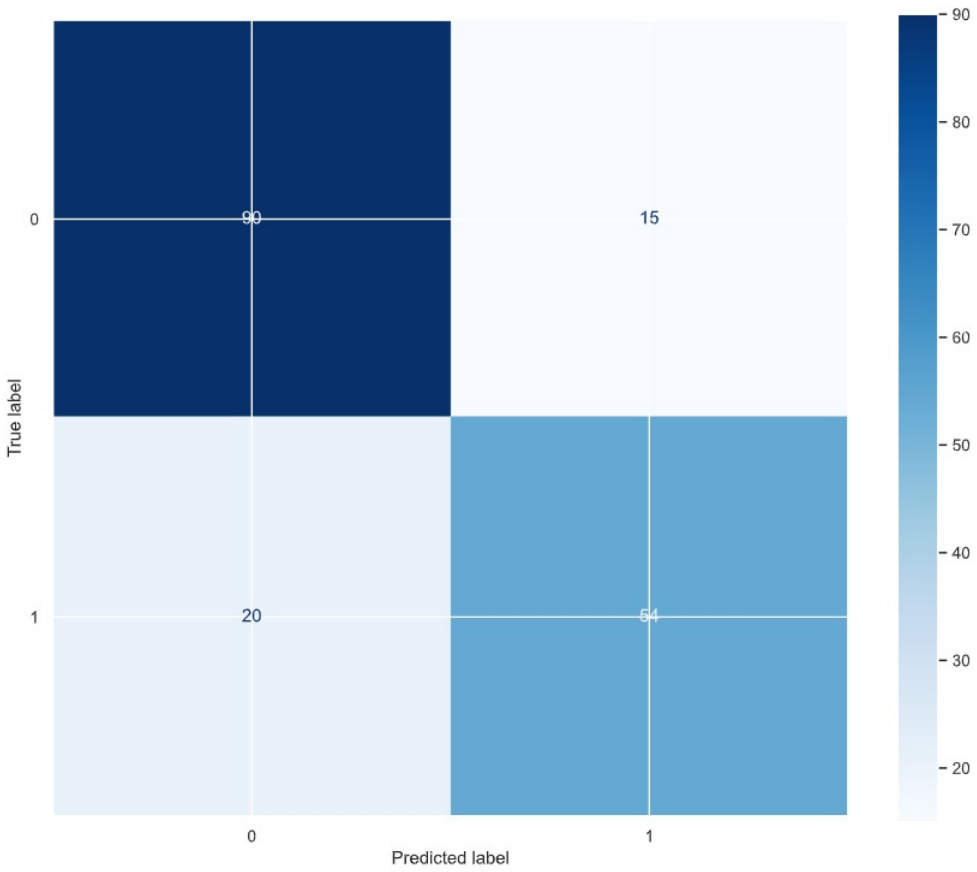

Figure 5: Confusion Matrix for Logistic Regression

K-Nearest Neighbors:

The accuracy obtained using the K-Nearest Neighbors Classification Model was 70.95% with a k=5.

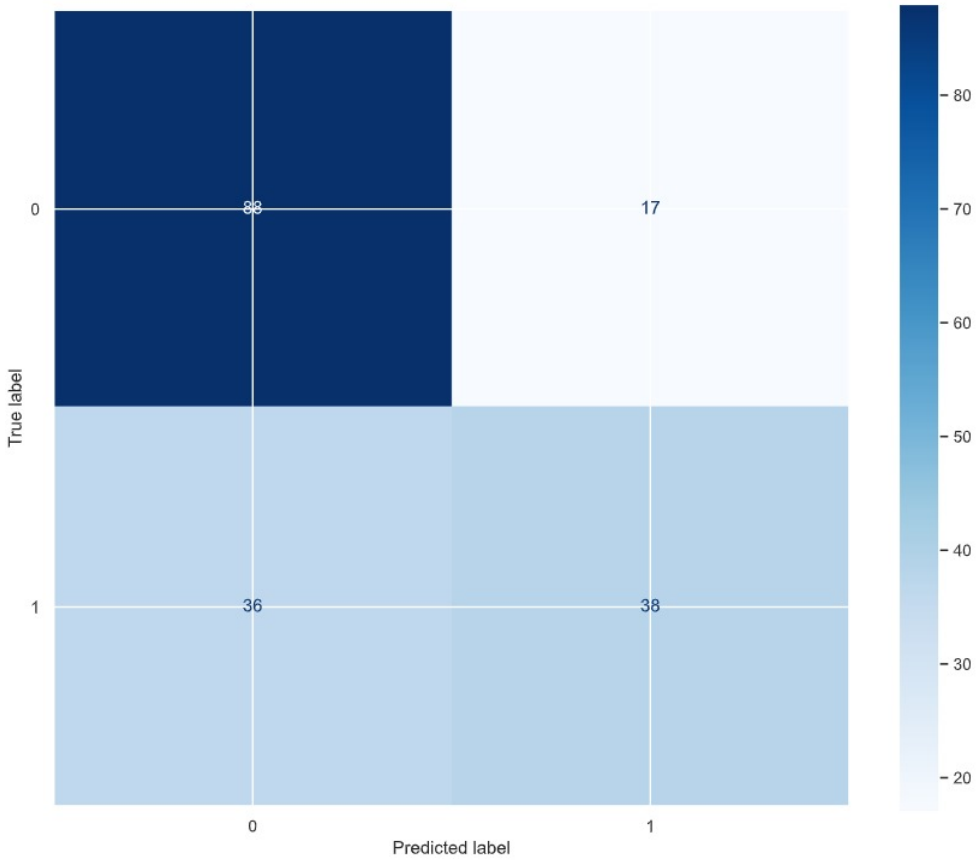

Figure 6: Confusion Matrix for KNN

Support Vector Machine:

The accuracy obtained using the Random Forests Classification Model was 65.36%.

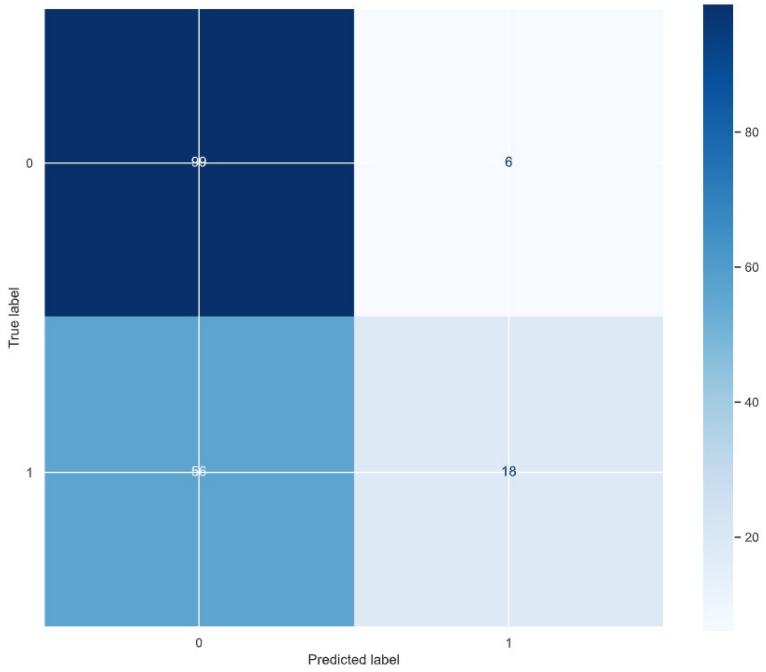

Figure 7: Confusion Matrix for SVM

Comparing the accuracy results obtained for each prediction model used, the Random Forests algorithm shows a minimal advantage in relation to the other models. The confusion matrices of each one confirm that assumption. It is worth noting that these accuracy results could be greatly increased if other techniques were applied to the dataset prior to the implementation of a prediction model. However, the treatment done at this stage suffices to show which algorithm is the most advantageous for this example.

Looking at the confusion matrices allows us to better understand where each model failed as it shows the predictions as well as the true values. In this case, 0 means that a passenger did not survive and 1 means it did.

Therefore, taking into account the least accurate model, which was the Support Vector Machine, it actually predicted the most number of passengers that perished out of all five models used. However, it could very well be a coincidence that would not be verified for all datasets as it also predicted wrongly for many passengers that did survive making him a very pessimistic classifier.

Up next is the KNN algorithm, which refers to K-Nearest Neighbors. For this iteration, k=5 proved to be the most accurate without falling into the overfitting category. This one also predicted the non-survivors quite accurately but it also turned out quite pessimistic although not as bad as the SVM. It showed a slight improvement by also predicting some survivors right,

although it didn't even reach 50% at that.

Continuing this assessment, the next two prediction models come very close in terms of accuracy. The Logistic Regression model and the Decision Trees Classifier present 78.21% and 79.33%, respectively. They both show quite a big improvement when predicting suvivors correctly. The non-survivor guesses have remained fairly accurate just as in the previous model.

Last but not least, the classifier that obtained the best predictions was indeed the Random Forests model but its optimality when compared to the Logistic Regression classifier is minimal.

Afterwards, the model that showed the best accuracy was used on the testing dataset provided by Kaggle and used to create a file of predictions for submission. That same file obtained an accuracy result of 76.07% which was expectedly far from the top scorers in the leaderboard.

Conclusions

After comparing every model used and then investigating how each of them work in order to understand why one has better results than the rest for this specific task at hand, it was clear that Linear Regression, which was not covered in this assignment was able to provide the best results. It proved to be far more accurate than even Random Forests.

Even though 80% accuracy is considered decent for this specific dataset, more techniques and different machine learning models could be applied to further increase this accuracy. For example, the Gradient Boosting Classifier or the Linear Regression model could be used to increase accuracy even further. Transforming some features like 'Age' even further to restrict conditions even more would definitely make a positive impact on predictions for any model. These models that were used, especially the Random Forests algorithm could be better tuned by entering the parameters manually and accepting only the very best parameters. Hyperparameter tuning seems like a viable method of optimizing the machine learning models. And even then, if the same level of care was to be done with the Linear Regression model, it would obtain greater accuracy results, even being able of obtaining 100% accuracy.

Taking into account the somewhat simplicity of the dataset, it isn't inexplicable that Linear Regression would obtain great results. However, it should still not be as accurate as a tree-based model like Random Forests or Decision Trees as they should be better at selecting important features in the dataset and then making better predictions based on such selections.

It is clear after finishing our work that CNN's do indeed perform much better on image recognition than the models we tested as we expected before starting the work. The best scores on Kaggle scored around 70% whereas the best performing model of our work, SVM's only scored around 52% accuracy.

KNN and Decision trees also have potential for distinguishing emotions from pictures, but they require quite a sizeable amount of data, while logistic regression is not great if we are looking for reliability in distinguishing several emotions, but if the goal is to classify 2 classes of emotions, then it has the potential to perform really well.

In the end, this assignment serves to show that different machine learning models can be used but knowing what model applies best to a certain scenario is extremely important. Fully tuning and optimizing the dataset and respective training model is also important, giving an extra boost of accuracy.