Data Mining
Lab 5 - Kaggle Competition
Team 'The One' - Gideon Chia, Melissa Chien, Rohit Raghavan, Peter Rowland, Ji Wei Yoon

Introduction

For this assignment, we created a machine learning model to predict the survival of passengers on the Titanic. The training dataset consists of 915 passengers and 11 features, and a target label to describe the survival of the passenger (1 is survived, 0 is the contrary). The features are categorical, numerical, ordinal in nature.

Data Exploration

For any machine learning exercise, it is important to have an understanding of the information represented in the dataset to generate initial intuitions about the usefulness and deficiencies of the features. To that end, we generated various summary statistics.

Pclass	**Survived**
1	0.657895
2	0.412371
3	0.251521

Table 1: Survival rates in various Pclass.

Sex	**Survived**
female	0.750000
male	0.187817

Table 2: Survival rates for each gender.

SibSp	**Survived**
1	0.528302
2	0.400000
0	0.353407
3	0.307692
4	0.176471
5	0.000000
8	0.000000

Table 3: Survival rates for different number of siblings.

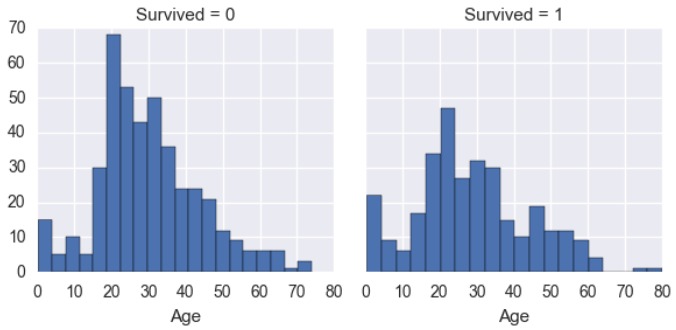

Figure 1: Survival and death distributions with respect to age of passenger.

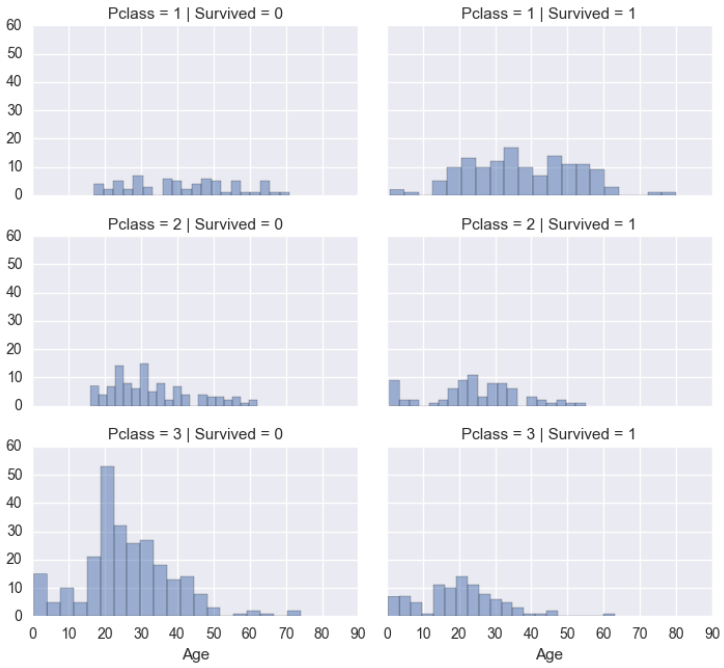

Figure 2: Survival and death distributions with respect to age of cabin class.

We discover various trends in the dataset:

1. The percentage of passengers in first class who survived was more than twice that of the third class cabin.
2. Females were four times more likely than men to have survived.
3. The less siblings a passenger has, the more likely he(she) survived.

Extra care has to be taken when one is interpreting figure 1 and 2 as the y-axis of the histograms represents frequencies and not percentage figures. At the very least, these two figures give us a summary of the distributional characteristics of the training dataset.

Data Pre-processing and Imputation

We started by investigating the data and doing pre-processing of the data to get it into a form that could be analyzed by the predictors. We picked the minmaxscaler in the preprocessing toolkit in sklearn to do feature normalization. We replaced blank values in age feature by the mean value of ages. Categorical variables are transformed into binary variables using the DictVectorizer() function, as recommended by various machine learning 'cookbooks' [1].

Feature Generation and Selection

In every machine learning exercise, there is a need to select feature that are predictive with respect to the algorithm used. Any leftover features are removed to prevent the model from tending to overfit to a large feature set - this is called dimension reduction. In our case, we did a pre-training selection of features with a validation set. The original variables - Pclass, Sex, Age, SubSp, parCh - were deemed to be predictive while the addition of features Fare, Embarked and Ticket were not.

We also added novel features that we think are intuitive for the machine learning objective: Family Size, Title, Age Bands and No Family. The enlarged feature set did result in some accuracy improvement.

Feature Importance

To choose the features, we looked at what data was available and hand-picked a couple of features that looked like they would be highly correlated with survival. We observed that there were some features that does not correlate with survival rate, like the name of the passenger. Thus, we did not use all the data provided in our classifiers. In our research on methods for feature engineering, we found several ways to create better features by the data in this dataset. Based on this research and our analysis of the correlations between certain features and survival, we focused on these features:

Family size
People with larger family sizes often ended up dying, probably because they were trying to save their family.

Sex
Women and children were given priority in the lifeboats, so the genders had a different survival rate.

Pclass
People with different PClasses were located in different parts of the ship, which may have led to different survival rates.

Age
Again, children were given priority. People who were elderly did not have good survival rates because they received no priority and were probably not in the best physical shape to escape from a quickly sinking ship.

Training and Hyperparameter Tuning

We then began training models on the data using the different techniques we have learned in the class so far to create a variety of predictors and get an idea of their initial accuracy. We used 5-fold cross validation to train our model to as to avoid overfitting.

We trained an SVM model, and did a coarse grid search of the hyperparameter space for C and gamma to find the optimal parameters. We then did a fine grid search of the hyperparameter space of C and gamma around +/- 2 of the best hyperparameter values we found using the coarse grid search. Our initial submission of the predictions of the tuned SVM model put us on the leaderboard and became the baseline of accuracy we then tried to improve upon.

We performed a similar search of the hyperparameter space of activation, solver, learning_rate and hidden_layer_sizes to find the best settings for our neural network.

We took a more manual approach to training a decision tree using different parameter values for max depth and leaf node parameters. We were able to get a high level of accuracy using this method and some feature engineering on the training set, but this model did not improve on the baseline prediction accuracy we got with the SVM.

In total, we tried a full range of machine learning models. The models and their accuracy on the test-set are listed in Table 4.

Model	**Accuracy (%)**
Random Forest	95.74
Decision Tree	95.74
KNN	86.34
Support Vector Machines	85.68
Logistic Regression	80.55
Naive Bayes	79.56
Linear SVC	79.56
Perceptron	72.79
Stochastic Gradient Descent	61.31

Table 4: Machine learning models and test-set accuracy

Ensemble Methods

We started combining this set of classifiers together to see if we could improve on our single predictor results. Our first attempt was to create a voting classifier and run it with a selection of the models that we created, using 'hard' voting or majority vote from each classifier. We used 5-fold cross validation and ran each classifier individually and then combined to see if the ensemble performed better than any of the classifiers individually. This ensemble performed slightly better than our most accurate SVM classifier, and it improved on our highest submission score by 2 percentage points.

To add more variety to our predictors, we tried some ensembling predictors like boosting (Adaboost and Stochastic Gradient Boosting), bagging, random forests. The main hyperparameter we varied with these ensemble predictors was n_estimators. These methods individually performed with about the same accuracy as our first set of predictors on the training set. We created another ensemble classifier that combined all of the models created so far to see if adding more classifier diversity resulted in better predictions. Unfortunately, it did not improve our baseline accuracy.

Bibliography

1. "Support Vector Machines for Classification." *Support Vector Machines Information Science and Statistics*: 285-329. Print.