Predicting Titanic Survival Rates with a Multi-Layer Perceptron

Q320 Final Project

Josh Isaacson

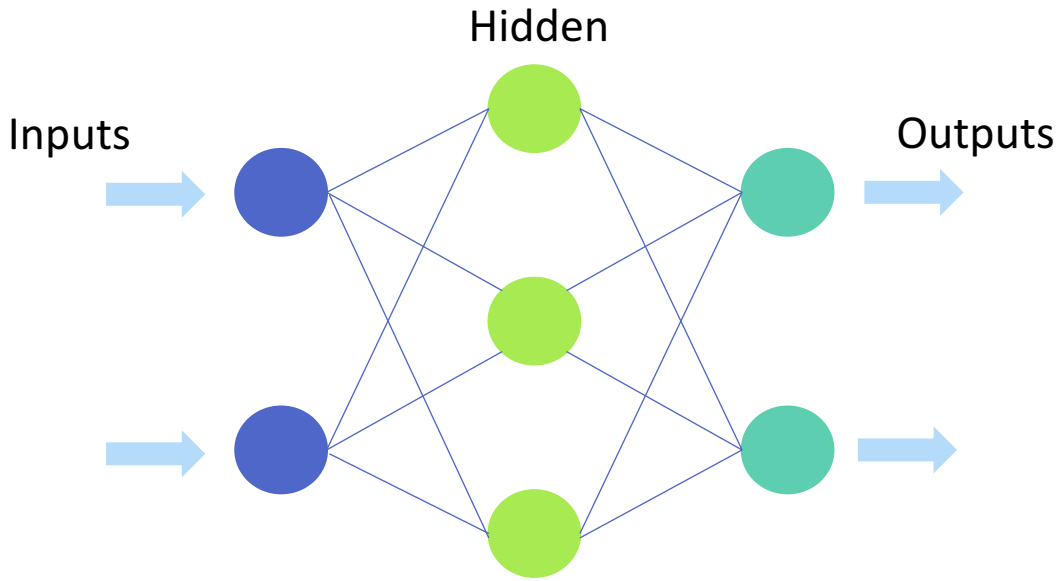
Broad Motivation

- Goal: learn how to implement a neural network on a relatively large dataset and validate its performance
- Dataset: Titanic: Machine Learning from Disaster
- Predict if a passenger survived the Titanic or not

kaggle™

Multi-Layer Perceptron

- Supervised learning model
- Model makes a guess, then evaluates the error, then makes changes to minimize it
  - Backpropagation
- Classification
  - Problem of IDing where a new instance belongs
  - Basis of a training set of data with known membership



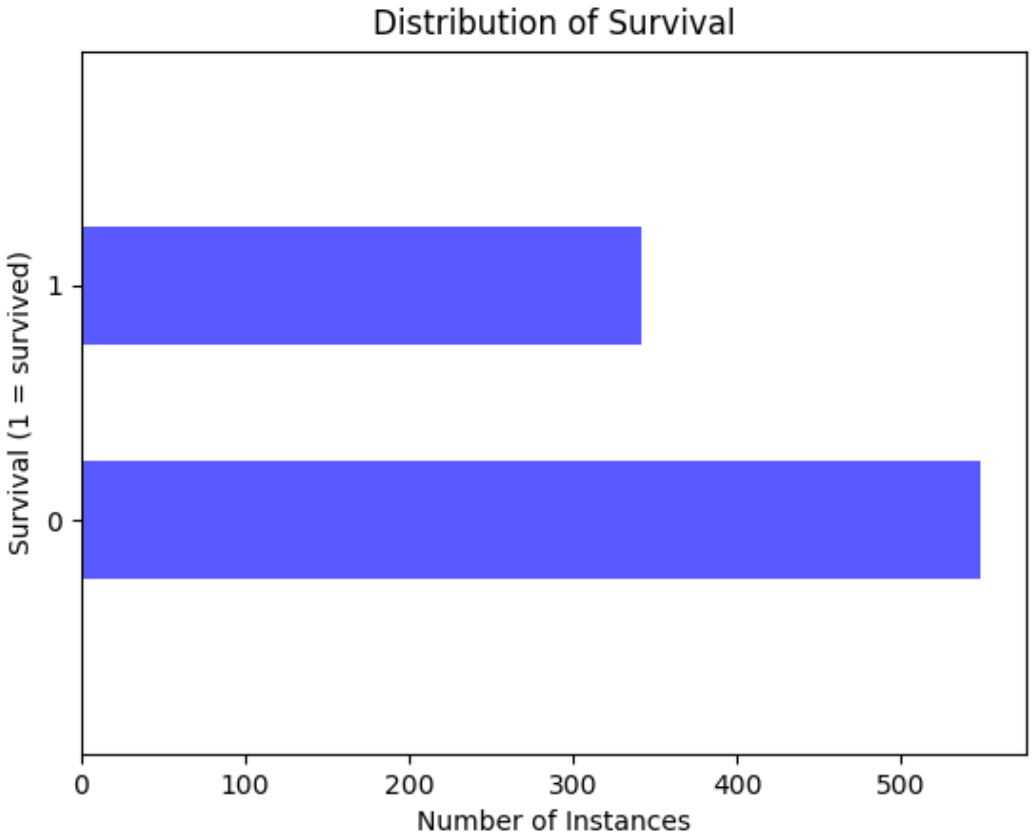
Data Exploration

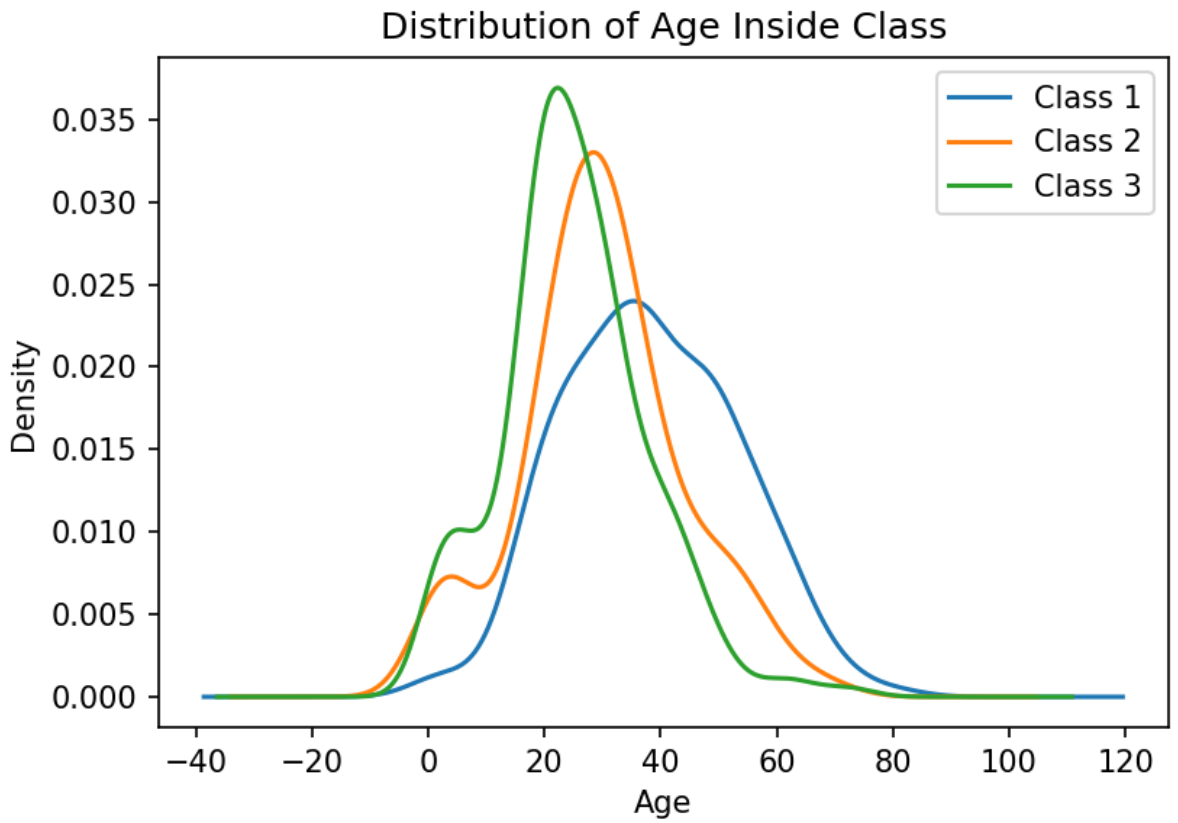
Raw Data (First 20 Lines)

	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	NaN	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	NaN	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	NaN	S

- Data is split into:
  - Train.csv
    - Trains the model and is used to choose features
  - Test.csv
    - Validates the model's performance on new data
      - Predict whether or not each passenger survives the Titanic

Data Exploration – Cont.





Pre-Processing

- Dropped PassengerId, Name, Ticket, and Cabin Columns
  - Not useful to analysis
- Set 0 = female, 1 = male in Sex Column
- Set variables to the 3 types of Class
- Filled NaN spaces in Ages by:
  - Created random generator based on avg, standard dev, and sum of null instances
- Filled NaN space in Fare

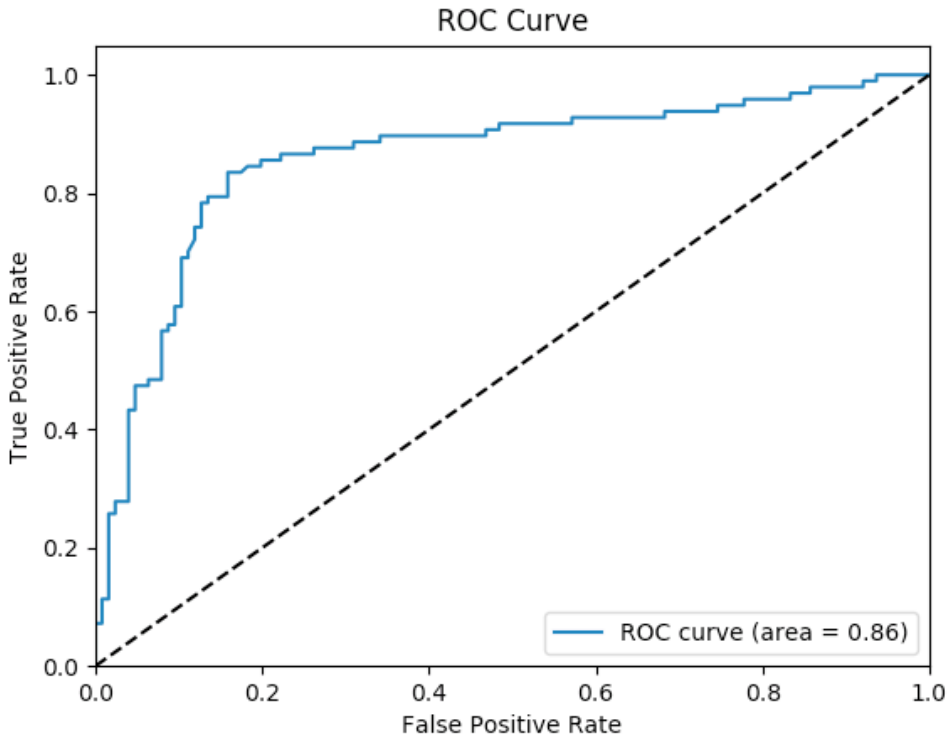
Classifier Implementation

- features used to train:
  - Pclass
  - Fare
  - Sex -> male
  - Age
- label used to train:
  - Survived

```
#  --- training and validation sets ---
# features used to train: Pclass, Fare, male or not, and Age
X_train = train[['Pclass','Fare','male','Age']]

# label used to train: Survived
Y_train = train[["Survived"]]

# features used to test: Pclass, Fare, male or not, and Age
X_test  = test[['Pclass','Fare','male','Age']]

#  --- Multi-Layer Perceptron (MLP) ---
mlp = MLPClassifier(solver='lbfgs',
                    alpha=1e-6,
                    hidden_layer_sizes=(100),
                    random_state=numpy.random.randint(0,10000),
                    learning_rate_init=0.001,
                    max_iter=10000,
                    early_stopping=False)

mlp.fit(X_train, Y_train.values.ravel())

Y_pred = mlp.predict(X_test)
score = (mlp.score(X_train, Y_train))
print("Accuracy of Multi-Layer Perceptron Predictions on the data was: {0}".format(score))
```

Interpretation of Results

- ROC Curve (Receiver Operating Characteristic)
  - Accuracy measured by area under the curve
    - (Greater area is better)
- The area really measures discrimination
  - Ability of the test to correctly classify those who survived and those who didn't




```
Accuracy of Multi-Layer Perceptron Predictions on the data was: 0.8439955106621774
ROC AUC: 0.86
```

Sources

- https://www.kaggle.com/c/titanic
- http://scikit-learn.org/stable/modules/neural_networks_supervised.html
- http://www.dataschool.io/roc-curves-and-auc-explained/

GitHub

https://github.com/jsisaacs/Q320-Final-Project