# Improving classification performance on Titanic dataset

# Result from project 1

| | 1. AIC | 2. BIC | 3. Ridge | 4. Lasso |
|---|---|---|---|---|
| Number of parameters | 6 | 3 | 8 | 5 |
| Parameter | Pclass -0.7529<br>Sex 2.2610<br>Age 0.2535<br>Embarked 0.2845<br>Title 0.3444<br>Age*Class -0.2785 | Pclass -0.9219<br>Sex 2.2909<br>Title 0.3840 | Pclass -0.255532<br>Sex 0.397340<br>Age -0.056094<br>Fare -0.006139<br>Embarked 0.061012<br>Title 0.149729<br>IsAlone 0.016209<br>Age*Class 0.002501 | Pclass -0.119751<br>Sex 0.202281<br>Age -0.016549<br>Embarked 0.021558<br>Title 0.064091 |
| Accuracy | 0.76794 | 0.76794 | 0.74162 | 0.77033 |

# 3 Objectives

- Improving Classification Performance with discretization
- Improving Classification Performance with different model
- Improving Classification Performance with hyper-parameters tuning

# Brief desciption on data pre-processing

| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **889** | 890 | 1 | 1 | Behr, Mr. Karl Howell | male | 26.0 | 0 | 0 | 111369 | 30.0000 | C148 | C |
| **651** | 652 | 1 | 2 | Doling, Miss. Elsie | female | 18.0 | 0 | 1 | 231919 | 23.0000 | NaN | S |
| **290** | 291 | 1 | 1 | Barber, Miss. Ellen "Nellie" | female | 26.0 | 0 | 0 | 19877 | 78.8500 | NaN | S |
| **396** | 397 | 0 | 3 | Olsson, Miss. Elina | female | 31.0 | 0 | 0 | 350407 | 7.8542 | NaN | S |
| **815** | 816 | 0 | 1 | Fry, Mr. Richard | male | NaN | 0 | 0 | 112058 | 0.0000 | B102 | S |
| **87** | 88 | 0 | 3 | Slocovski, Mr. Selman Francis | male | NaN | 0 | 0 | SOTON/OQ 392086 | 8.0500 | NaN | S |
| **615** | 616 | 1 | 2 | Herman, Miss. Alice | female | 24.0 | 1 | 2 | 220845 | 65.0000 | NaN | S |
| **127** | 128 | 1 | 3 | Madsen, Mr. Fridtjof Arne | male | 24.0 | 0 | 0 | C 17369 | 7.1417 | NaN | S |
| **209** | 210 | 1 | 1 | Blank, Mr. Henry | male | 40.0 | 0 | 0 | 112277 | 31.0000 | A31 | C |
| **310** | 311 | 1 | 1 | Hays, Miss. Margaret Bechstein | female | 24.0 | 0 | 0 | 11767 | 83.1583 | C54 | C |

- Fill in age, embark, fare missing value
- Drop passenger id, cabin and ticket
- Create family size feature
- Extract title from name

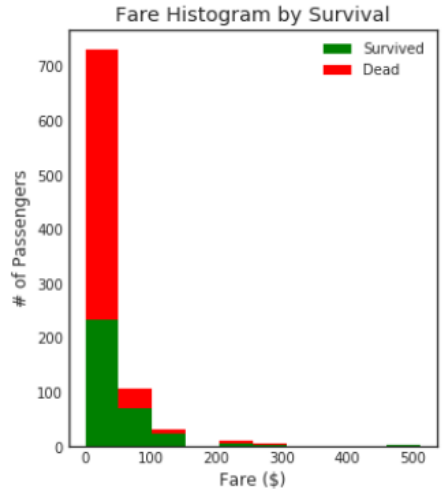
# Feature discretization/Binning

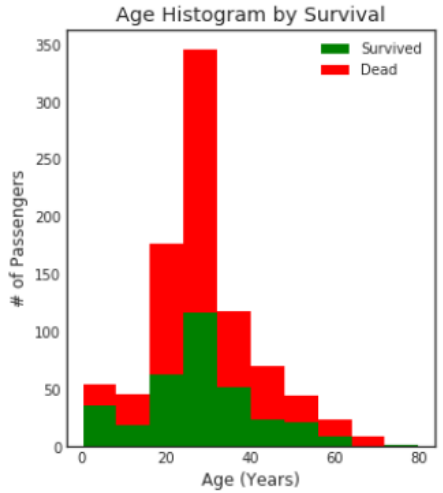
- Among all variables, only age and fare are continous variables

| | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Fare | Embarked | FamilySize | IsAlone | Title | FareBin | AgeBin |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **296** | 0 | 3 | Hanna, Mr. Mansour | male | 23.50 | 0 | 0 | 7.2292 | C | 1 | 1 | Mr | (-0.001, 7.91] | (16.0, 32.0] |
| **578** | 0 | 3 | Caram, Mrs. Joseph (Maria Elias) | female | 28.00 | 1 | 0 | 14.4583 | C | 2 | 0 | Mrs | (14.454, 31.0] | (16.0, 32.0] |
| **644** | 1 | 3 | Baclini, Miss. Eugenie | female | 0.75 | 2 | 1 | 19.2583 | C | 4 | 0 | Miss | (14.454, 31.0] | (-0.08, 16.0] |
| **339** | 0 | 1 | Blackwell, Mr. Stephen Weart | male | 45.00 | 0 | 0 | 35.5000 | S | 1 | 1 | Mr | (31.0, 512.329] | (32.0, 48.0] |
| **383** | 1 | 1 | Holverson, Mrs. Alexander Oskar (Mary Aline To... | female | 35.00 | 1 | 0 | 52.0000 | S | 2 | 0 | Mrs | (31.0, 512.329] | (32.0, 48.0] |
| **732** | 0 | 2 | Knight, Mr. Robert J | male | 28.00 | 0 | 0 | 0.0000 | S | 1 | 1 | Mr | (-0.001, 7.91] | (16.0, 32.0] |
| **185** | 0 | 1 | Rood, Mr. Hugh Roscoe | male | 28.00 | 0 | 0 | 50.0000 | S | 1 | 1 | Mr | (31.0, 512.329] | (16.0, 32.0] |
| **529** | 0 | 2 | Hocking, Mr. Richard George | male | 23.00 | 2 | 1 | 11.5000 | S | 4 | 0 | Mr | (7.91, 14.454] | (16.0, 32.0] |
| **558** | 1 | 1 | Taussig, Mrs. Emil (Tillie Mandelbaum) | female | 39.00 | 1 | 1 | 79.6500 | S | 3 | 0 | Mrs | (31.0, 512.329] | (32.0, 48.0] |
| **515** | 0 | 1 | Walker, Mr. William Anderson | male | 47.00 | 0 | 0 | 34.0208 | S | 1 | 1 | Mr | (31.0, 512.329] | (32.0, 48.0] |

# Apply feature discretization/binning

- Pandas method Qcut is applied to Fare
- Pandas method cut is applied to Age




| | |
|---|---|
| 1 | 224 |
| 0 | 223 |
| 2 | 222 |
| 3 | 222 |

| | |
|---|---|
| 1 | 525 |
| 2 | 186 |
| 0 | 100 |
| 3 | 69 |
| 4 | 11 |

# Without discretization vs with discretization

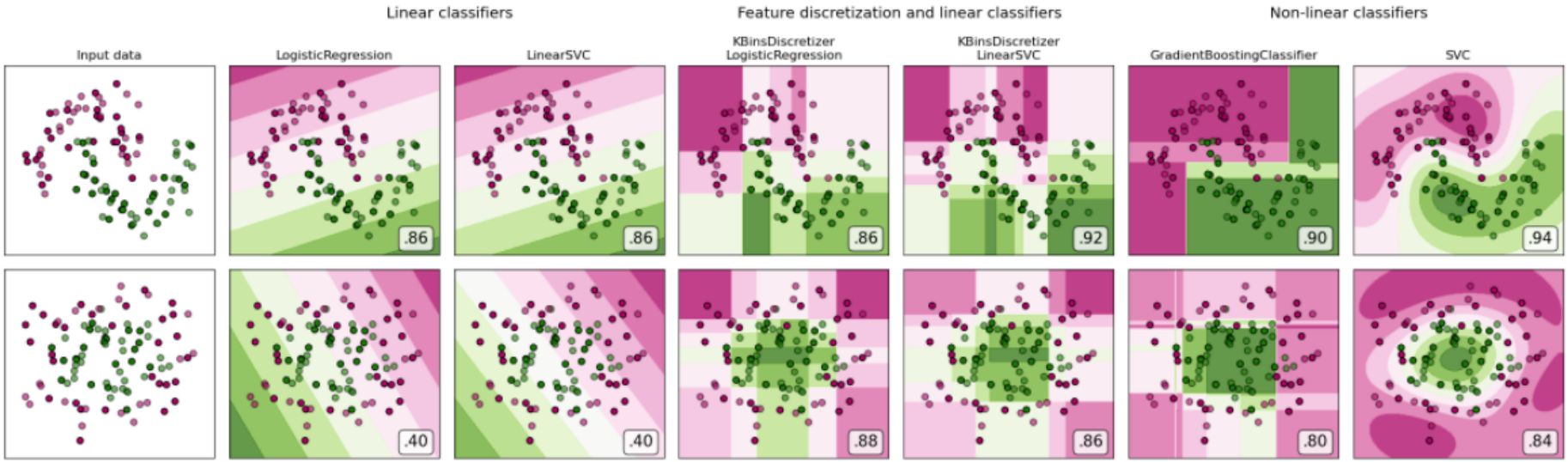
| | Sex_Code | Pclass | Embarked_Code | Title_Code | FamilySize | Age | Fare |
|---|---|---|---|---|---|---|---|
| **0** | 1 | 3 | 2 | 3 | 2 | 22.0 | 7.2500 |
| **1** | 0 | 1 | 0 | 4 | 2 | 38.0 | 71.2833 |
| **2** | 0 | 3 | 2 | 2 | 1 | 26.0 | 7.9250 |
| **3** | 0 | 1 | 2 | 4 | 2 | 35.0 | 53.1000 |
| **4** | 1 | 3 | 2 | 3 | 1 | 35.0 | 8.0500 |
| **5** | 1 | 3 | 1 | 3 | 1 | 28.0 | 8.4583 |
| **6** | 1 | 1 | 2 | 3 | 1 | 54.0 | 51.8625 |
| **7** | 1 | 3 | 2 | 0 | 5 | 2.0 | 21.0750 |
| **8** | 0 | 3 | 2 | 4 | 3 | 27.0 | 11.1333 |
| **9** | 0 | 2 | 0 | 4 | 2 | 14.0 | 30.0708 |

| | Sex_Code | Pclass | Embarked_Code | Title_Code | FamilySize | AgeBin_Code | FareBin_Code |
|---|---|---|---|---|---|---|---|
| **0** | 1 | 3 | 2 | 3 | 2 | 1 | 0 |
| **1** | 0 | 1 | 0 | 4 | 2 | 2 | 3 |
| **2** | 0 | 3 | 2 | 2 | 1 | 1 | 1 |
| **3** | 0 | 1 | 2 | 4 | 2 | 2 | 3 |
| **4** | 1 | 3 | 2 | 3 | 1 | 2 | 1 |
| **5** | 1 | 3 | 1 | 3 | 1 | 1 | 1 |
| **6** | 1 | 1 | 2 | 3 | 1 | 3 | 3 |
| **7** | 1 | 3 | 2 | 0 | 5 | 0 | 2 |
| **8** | 0 | 3 | 2 | 4 | 3 | 1 | 1 |
| **9** | 0 | 2 | 0 | 4 | 2 | 0 | 2 |

# How feature discretization/Binning affect performance?

- Intution:
  - Some information may loss because of feature discretization
- Hypothesis:
  - Performance should increase for linear model
  - Performance will be at least the same for non-linear model
  - Should have little to no effect on tree-based model
  - Less overfitting with feature discretization

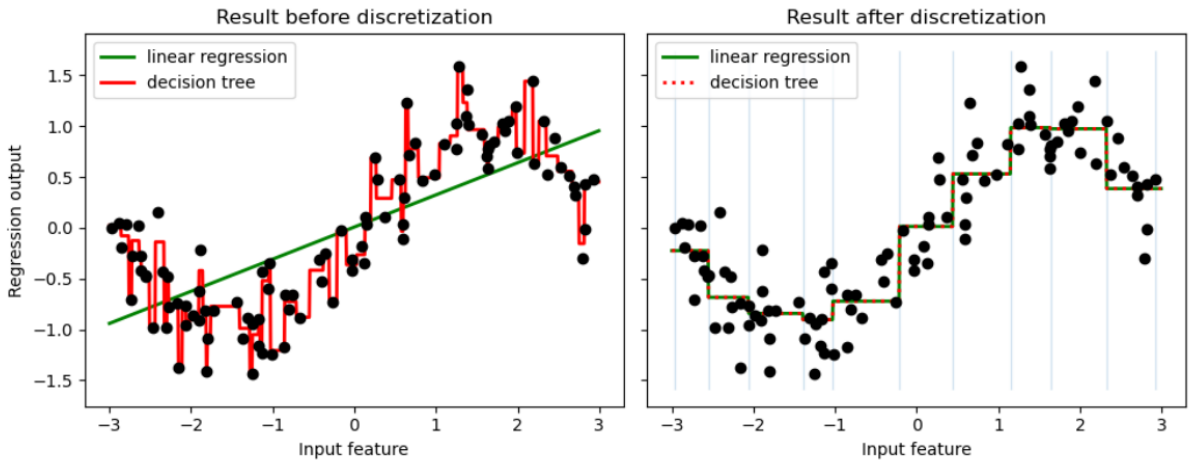
# Rationale behind the hypothesis




- With 2 features, binning can create similar effect for linear model as non-linear classifier

# Rationale of the hypothesis

- However, for non-linear model( e.g. decision tree), binning does not help
- Less overfit, because reduce noise/variance (at the cost of increasing bias)



# Result

## Without discretization

| | MLA Name | MLA Parameters | MLA Train Accuracy Mean | MLA Test Accuracy Mean | MLA Test Accuracy 3*STD | MLA Time |
|---|---|---|---|---|---|---|
| 0 | GradientBoostingClassifier | {'ccp_alpha': 0.0, 'criterion': 'friedman_mse'... | 0.922285 | 0.825373 | 0.05769 | 0.100948 |
| 1 | RandomForestClassifier | {'bootstrap': True, 'ccp_alpha': 0.0, 'class_w... | 0.98764 | 0.813433 | 0.025526 | 0.182288 |
| 2 | LogisticRegressionCV | {'Cs': 10, 'class_weight': None, 'cv': None, '... | 0.809738 | 0.791418 | 0.057831 | 0.82209 |
| 3 | DecisionTreeClassifier | {'ccp_alpha': 0.0, 'class_weight': None, 'crit... | 0.987828 | 0.784701 | 0.040376 | 0.007579 |
| 4 | SVC | {'C': 1.0, 'break_ties': False, 'cache_size': ... | 0.677341 | 0.688433 | 0.069231 | 0.085074 |

## With discretization

| | MLA Name | MLA Parameters | MLA Train Accuracy Mean | MLA Test Accuracy Mean | MLA Test Accuracy 3*STD | MLA Time |
|---|---|---|---|---|---|---|
| 4 | SVC | {'C': 1.0, 'break_ties': False, 'cache_size': ... | 0.83839 | 0.826493 | 0.059286 | 0.070706 |
| 0 | GradientBoostingClassifier | {'ccp_alpha': 0.0, 'criterion': 'friedman_mse'... | 0.872097 | 0.822388 | 0.033657 | 0.089094 |
| 1 | RandomForestClassifier | {'bootstrap': True, 'ccp_alpha': 0.0, 'class_w... | 0.899813 | 0.811194 | 0.034027 | 0.182235 |
| 3 | DecisionTreeClassifier | {'ccp_alpha': 0.0, 'class_weight': None, 'crit... | 0.899813 | 0.808955 | 0.044439 | 0.005686 |
| 2 | LogisticRegressionCV | {'Cs': 10, 'class_weight': None, 'cv': None, '... | 0.80206 | 0.778731 | 0.052277 | 0.311267 |

# Analysis

- Performance does not increase for logistic regression(Without discretization testing accuracy:0.791418, With discretization:0.778731)
- All other models' test accuracy increase
- For tree-based models, the test accuracy mostly remain the same or slightly increase
- Without discretization, some models' train accuracy goes up to >0.9

# Explaination

- Is the dataset linearly seperable?
- Is the two features with binning apply important?
- Is the model optimize?

# Improving Classification Performance with hyper-parameters tuning

Without discretization

| | MLA Name | MLA Parameters | MLA Train Accuracy Mean | MLA Test Accuracy Mean | MLA Test Accuracy 3*STD |
|---|---|---|---|---|---|
| 2 | RandomForestClassifier | {'criterion': 'entropy', 'max_depth': 8, 'n_es... | 90.472955 | 86.410815 | 5.319646 |
| 3 | LogisticRegressionCV | {'Cs': 10, 'penalty': 'l2'} | 86.063145 | 85.150016 | 5.011195 |
| 1 | GradientBoostingClassifier | {'learning_rate': 0.2, 'loss': 'deviance', 'ma... | 92.511855 | 84.879984 | 7.425558 |
| 4 | DecisionTreeClassifier | {'criterion': 'gini', 'max_depth': 4, 'random_... | 90.086761 | 84.670324 | 7.729409 |
| 0 | SVC | {'C': 100} | 74.595554 | 74.335287 | 6.483699 |

With discretization

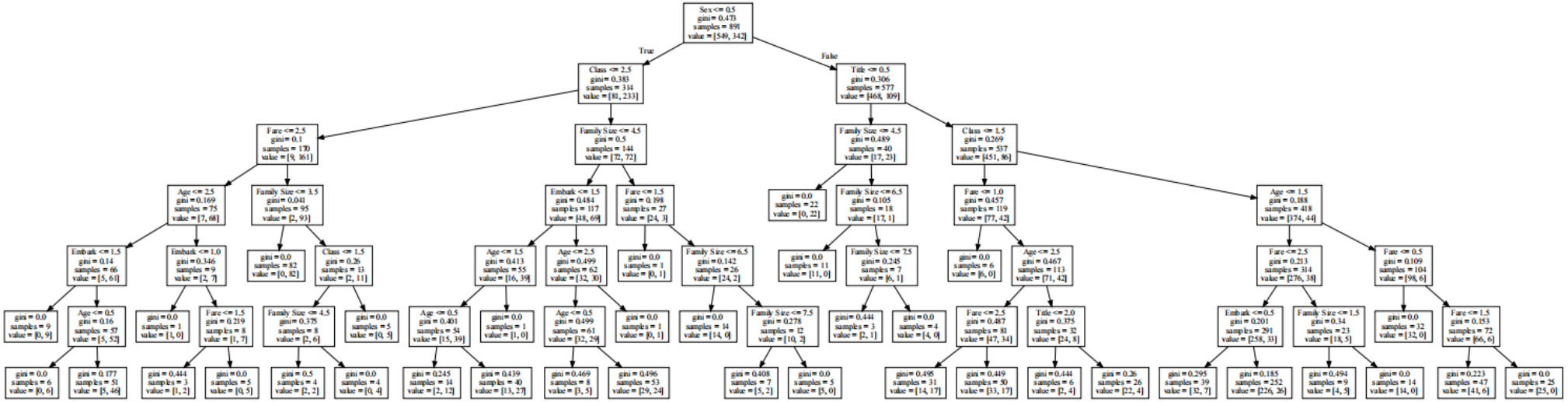
| | MLA Name | MLA Parameters | MLA Train Accuracy Mean | MLA Test Accuracy Mean | MLA Test Accuracy 3*STD |
|---|---|---|---|---|---|
| 4 | DecisionTreeClassifier | {'criterion': 'gini', 'max_depth': 4, 'random_... | 89.632818 | 86.799306 | 4.844119 |
| 2 | RandomForestClassifier | {'criterion': 'entropy', 'max_depth': 5, 'n_es... | 89.188418 | 86.599677 | 4.703276 |
| 1 | GradientBoostingClassifier | {'learning_rate': 0.15, 'loss': 'deviance', 'm... | 91.315719 | 85.893921 | 6.720044 |
| 3 | LogisticRegressionCV | {'Cs': 10, 'penalty': 'l2'} | 86.759518 | 85.665497 | 4.817815 |
| 0 | SVC | {'C': 0.1} | 88.488571 | 84.223976 | 5.274345 |

# Analysis

- All model improves after grid search
- Comparing with or without discretization:
  - Performance increases slightly for logistic regression(Without discretization testing accuracy:0.8515, With discretization:0.8566)
  - All other models' test accuracy increase with discretization
  - The second most significant increase is from decision tree model(Without discretization testing accuracy:0.84670, With discretization:0.86799)
  - The most significant increase is from SVC(Without discretization testing accuracy:0.74335, With discretization:0.84224)

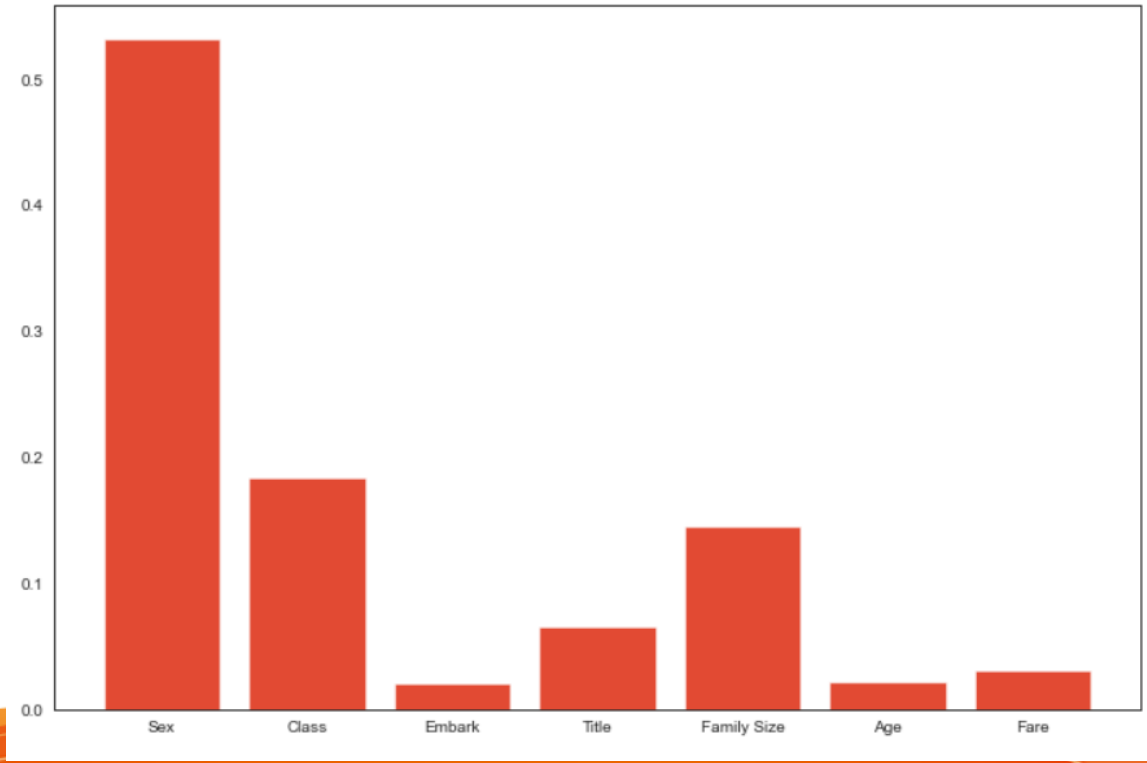
# Best model overall

- Decision Tree with discretization



# Feature importance

```
Feature: 0, Score: 0.53253
Feature: 1, Score: 0.18366
Feature: 2, Score: 0.02068
Feature: 3, Score: 0.06581
Feature: 4, Score: 0.14558
Feature: 5, Score: 0.02165
Feature: 6, Score: 0.03010
```



# Conclusion

- Compare to the logistic regression model in project 1, the performance has been improved by completing the 3 objectives.
  - Using feature discretization, most models' test accuracy improve
  - Using different kind of models, tree-based model (e.g. decision tree) works much better for this dataset
  - Using hyper-parameters tuning, all models' test accuracy improve significantly

# Reference

- Decision Tree visualization
  - https://mljar.com/blog/visualize-decision-tree/
  - https://machinelearningmastery.com/calculate-feature-importance-with-python/
- Feature discretization
  - https://www.kaggle.com/tilii7/feature-discretization-less-is-better
  - https://scikit-learn.org/stable/auto_examples/preprocessing/plot_discretization_classification.html
  - https://pbpython.com/pandas-qcut-cut.html
  - https://scikit-learn.org/stable/auto_examples/preprocessing/plot_discretization.html