# PassengerId => 乘客ID Pclass => 乘客等级(1/2/3等舱位) Name => 乘客姓名 Sex => 性别 Age => 年龄 SibSp => 堂兄弟/妹个数 Parch => 父母与小孩个数 Ticket => 船票信息 Fare => 票价 Cabin => 客舱 Embarked => 登船港口

# Introduction

1. draw graph
2. deal with missing value, mainly about age and cabin, the Embarked will be solved by factorization.
3. factorization, Scaling
4. building the LR model use train dataset, do the same thing to test dataset, the missing attribute is age, fare, cabin, embarked, we need fil the only missed fare value with 0
5. after prediction, analyze the coef.
6. Use cross\_validation to test first model
7. Extract all bad\_cases
8. Draw learning curve
9. Bagging, model ensemble

# Data Exploration

## Visualization

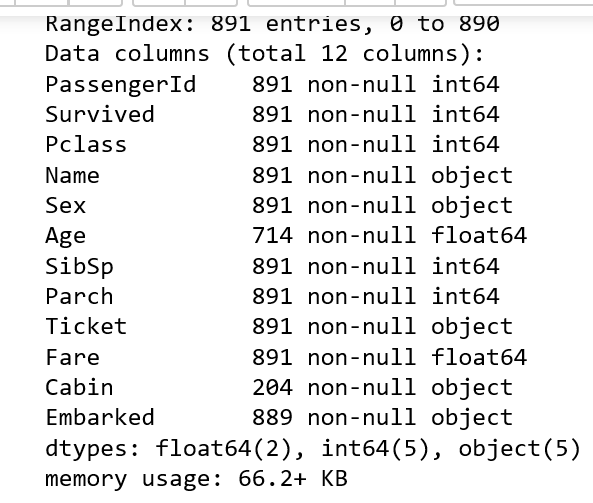
What we do at this stage is called **EDA (Exploratory Data Analysis)**, which means analytically exploring data in order to provide some insights for subsequent processing and modeling.

## Statistical Tests

# Data Preprocessing

* Deal with [**missing data**](https://en.wikipedia.org/wiki/Missing_data).
* Deal with [**outliers**](https://en.wikipedia.org/wiki/Outlier).
* Encode [**categorical variables**](https://en.wikipedia.org/wiki/Categorical_variable)

1. deal with missing data



From this table, the attribute ‘Age’ and ‘Cabin’ has missing value, so we need to handle these first.

* 如果缺值的样本占总数比例极高，我们可能就直接舍弃了，作为特征加入的话，可能反倒带入noise，影响最后的结果了
* 如果缺值的样本适中，而该属性非连续值特征属性(比如说类目属性)，那就把NaN作为一个新类别，加到类别特征中
* 如果缺值的样本适中，而该属性为连续值特征属性，有时候我们会考虑给定一个step(比如这里的age，我们可以考虑每隔2/3岁为一个步长)，然后把它离散化，之后把NaN作为一个type加到属性类目中。
* 有些情况下，缺失的值个数并不是特别多，那我们也可以试着根据已有的值，拟合一下数据，补充上。

Missing value method:

这里用scikit-learn中的RandomForest来拟合一下缺失的年龄数据(注：RandomForest是一个用在原始数据中做不同采样，建立多颗DecisionTree，再进行average等等来降低过拟合现象，提高结果的机器学习算法

# Feature Engineering

## Feature Selection

## Feature Encoding

# Model Selection

## Model Training

Here use logistic regression to build the model, because though regression is not to do classification , but logistic is a binary question could help to do this, the test dataset’s Survival attribute is binary with 0 and 1 , so bingo!

首先特征因子化：

逻辑回归建模时，需要输入的特征都是数值型特征，我们通常会先对类目型的特征因子化。   
什么叫做因子化呢？举个例子：

以Cabin为例，原本一个属性维度，因为其取值可以是[‘yes’,’no’]，而将其平展开为’Cabin\_yes’,’Cabin\_no’两个属性，成为两列，全部数值化

* 原本Cabin取值为yes的，在此处的”Cabin\_yes”下取值为1，在”Cabin\_no”下取值为0
* 原本Cabin取值为no的，在此处的”Cabin\_yes”下取值为0，在”Cabin\_no”下取值为1

## Cross Validation

# Ensemble Generation

## Stacking

**\*Pipeline**

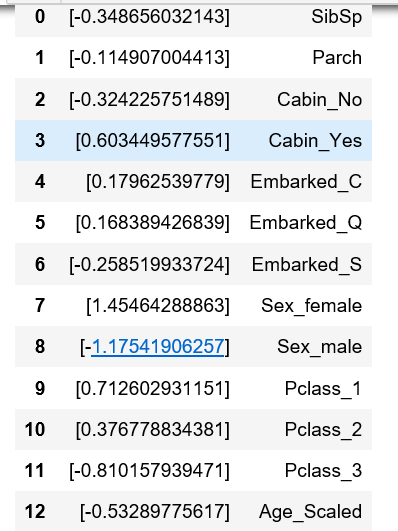
0.decribe data

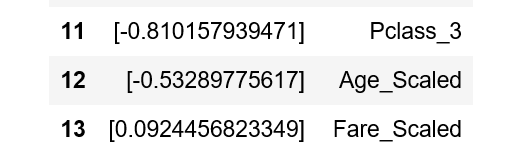
* 1. Missing value – randomforestregressor
  2. Discretization – pd.get\_dummies & pd.concat([],axis=1) df.drop([],axis=1,inplace=True)
  3. Scaling variance values (age and Fare) -- preprocessing.StandarScaler()
  4. Extract feature into model , build the model

According to the LR model’s coefficient , we could get some informations

* Sex属性，如果是female会极大提高最后获救的概率，而male会很大程度拉低这个概率。
* Pclass属性，1等舱乘客最后获救的概率会上升，而乘客等级为3会极大地拉低这个概率。
* 有Cabin值会很大程度拉升最后获救概率(**这里似乎能看到了一点端倪，事实上从最上面的有无Cabin记录的Survived分布图上看出，即使有Cabin记录的乘客也有一部分遇难了，估计这个属性上我们挖掘还不够**)
* Age是一个负相关，意味着在我们的模型里，年龄越小，越有获救的优先权(**还得回原数据看看这个是否合理**）
* 有一个登船港口S会很大程度拉低获救的概率，另外俩港口压根就没啥作用(**这个实际上非常奇怪，因为我们从之前的统计图上并没有看到S港口的获救率非常低，所以也许可以考虑把登船港口这个feature去掉试试**)。
* 船票Fare有小幅度的正相关(**并不意味着这个feature作用不大，有可能是我们细化的程度还不够，举个例子，说不定我们得对它离散化，再分至各个乘客等级上？**)

Out[60]:

 compare with bad\_cases to figure out



Question:

1. About the age, when we have all values( use the average of different people), the scaling all value between (-1,1) treat as continuous value or do discretization, follow some rules to discretized into several part, which is better ? what rules ?
2. About the age, Is that the discrete is better than continuous values ?
3. About the scaling, should I minimize all attributes into the same level ? coz the Parch (0,6) and the SibSp(0,8) , others is binary attributes. The scaling is different.
4. What’s the normal analytic process, or step by step.

Following problem:

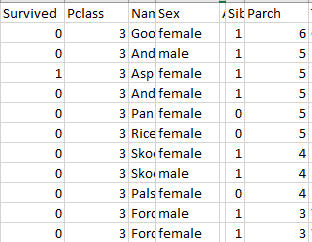
1. In the first model, I cut off name and ticket attributes, ticket I have no idea. But there are Miss, Mrs. Mr. in the Name , should be created a new feature.
2. About the Age, It’s weird to predict the age just by other numeric attribute by using rfr, could fill by the average age of the Mr. Mrs and Miss.
3. Then, about the age can be discretized, with some rules, use dummy to become the binary attribute, or not. Keep continuous or discrete, which better?
4. About age, about child, continuous age cannot tell anything, should separate it, to find old and child, below 12 or upper 75..
5. About the mother, find the female and Parch >1 , mother have better chance.

High frequency trading

Google scholar

Cited >100

Nyc dataset

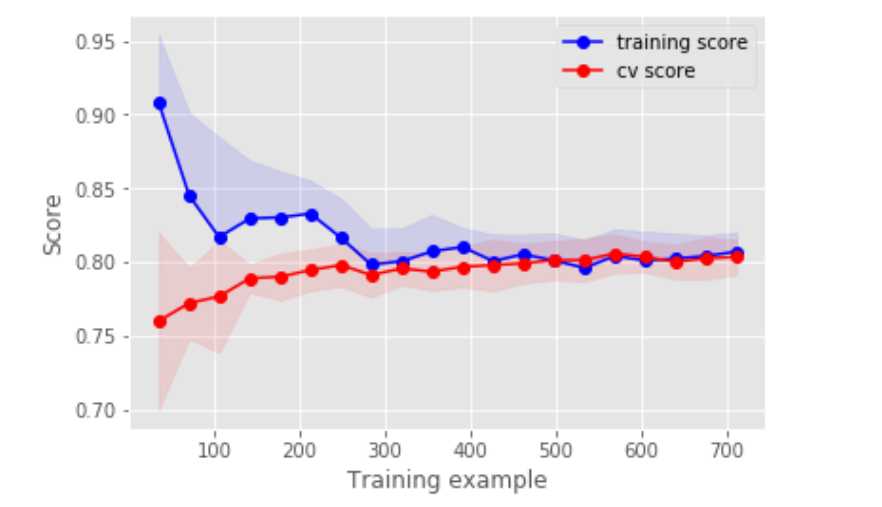


But I find a problem, the upper class 1 and 2 , most female have no child, they alive. But in the 3 class, most of female have children, they die. Maybe mother is not a good feature in this particular dataset? From raw train dataset. Maybe class is domination.

1. About single dog, parch=0 and age= youth, they died first. Poorly.
2. About family, from the name, find the family, they may move together, (should add themselves too) also can find family\_size, create a new feature, maybe big and powerful family have impact on decision( the single strong youth may alive from powerful family.
3. About Cabin, just treat that as Yes or No is ridiculous, coz the Cabin is like ‘C33’, maybe should separate them, ‘C’ maybe the layer of ship.
4. About the Pclass and Sex, I binarized Pclass into 6 columns, Sex into 2 columns, maybe they two can bind together to find something. Create new features and binarize them.
5. About stowaway, robbers.
6. I just use the LR model , use the train dataset to build the model and use the test to check. How to deal with overfitting?

From Cross\_Validation, only knows the precision is [ 0.81111111 0.81111111 0.78651685 0. 84269663 0.80898876 0.76404494

0.78651685 0.80898876 0.80898876 0.81818182]

1. 
2. What if the test score is better than training, should I cut off dataset?

Use the model ensemble ?

Without other meathod

Only LR use bagging?