In [1]:

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import random
```

In [2]:

```
train_data = pd.read_csv('train.csv')
train_data.head()
train_data.Age.fillna(train_data['Age'].median())
```

Out[2]:

```
0      22.0
1      38.0
2      26.0
3      35.0
4      35.0
       ...
886    27.0
887    19.0
888    28.0
889    26.0
890    32.0
Name: Age, Length: 891, dtype: float64
```

In [3]:

```
test_data = pd.read_csv('test.csv')
test_data.head()
```

Out[3]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Emba
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	

In [4]:

```
submission_sample = pd.read_csv('gender_submission.csv')
submission_sample.head()
```

Out[4]:

	PassengerId	Survived
0	892	0
1	893	1
2	894	0
3	895	0
4	896	1

In [5]:

```
women = train_data.loc[train_data.Sex == 'female']["Survived"]
rate_women = women.mean()

print("% of women who survived:", rate_women)
```

```
% of women who survived: 0.7420382165605095
```

In [6]:

```
men = train_data.loc[train_data.Sex == 'male']["Survived"]
rate_men = men.mean()

print("% of men who survived:", rate_men)
```

```
% of men who survived: 0.18890814558058924
```

In [7]:

```
y = train_data.Survived
X = train_data.drop(['Survived','Name'], axis=1)
```

In [8]:

```
X_train_sub = test_data.drop('Name', axis=1)
```

In [9]:

```
features = ["Pclass", "Sex", "SibSp", "Parch"]
X = pd.get_dummies(X[features])
```

In [10]:

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
X_train, X_test, y_train, y_test = train_test_split(X,y)
model = DecisionTreeClassifier(max_depth = 9, min_samples_split=0.01,min_samples_leaf =4)
#model = LogisticRegression(max_iter = 1000)
model.fit(X_train,y_train)
test_score = model.score(X_test, y_test)
print('Test score is : {}'.format(test_score))

train_score = model.score(X_train, y_train)
print('Train score is : {}'.format(train_score))
```

```
Test score is : 0.7533632286995515
Train score is : 0.8233532934131736
```

In [11]:

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler



X_train, X_test, y_train, y_test = train_test_split(X,y)
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.fit_transform(X_test)

model = RandomForestClassifier(max_depth = 9, min_samples_split=0.01,min_samples_leaf =4)
#model = LogisticRegression(max_iter = 1000)
model.fit(X_train_scaled,y_train)
test_score = model.score(X_test_scaled, y_test)
print('Test score is : {}'.format(test_score))

train_score = model.score(X_train_scaled, y_train)
print('Train score is : {}'.format(train_score))
```

```
Test score is : 0.7892376681614349
Train score is : 0.812874251497006
```

In [12]:

```
features = ["Pclass", "Sex", "SibSp", "Parch"]
X_test_sub = pd.get_dummies(test_data[features])
```

In [15]:

```
pred = model.predict(scaler.fit_transform(X_test_sub))
output = pd.DataFrame({'PassengerId': test_data.PassengerId, 'Survived': pred})
output.head(10)
```

Out[15]:

	PassengerId	Survived
0	892	0
1	893	1
2	894	0
3	895	0
4	896	1
5	897	0
6	898	1
7	899	0
8	900	1
9	901	0

In [16]:

```
output.to_csv('submission3.csv', index=False)
```

In [ ]:

```

```