شش.

بعد از خواندن داده‌ها، شروع به پیش‌پردازش می‌کنیم:

```
def read_clean(file_name):
    df = pd.read_csv(file_name)
    df["SibSp"] = df["SibSp"] + df["Parch"]
    df = df.replace({"Sex": {'male': 1, 'female': 0}, "Embarked": {"C": 10, "Q": 20, "S": 30}})

    del df["Name"]
    del df["Parch"]
    del df["Ticket"]
    del df["PassengerId"]
    del df["Cabin"]

    df.fillna(df.mean().astype(int), inplace=True)

    return df
```

داده‌های مهم غیرعددی را به داده‌های عددی map می‌کنیم، داده‌های بی‌ربط را حذف می‌کنیم و مقدارهای ناموجود را با میانگین بقیه داده‌ها جای‌گزین می‌کنیم.

با معیارهای جینی و آنتروپی در عمق‌های ۲ – ۴ و ۵ تست کردم و بهترین جواب به دست آمده که با ملاک جینی و عمق ۵ بود، امتیاز ۷۷.۵ را از سایت Kaggle گرفت:

(با استفاده از RandomForest به جواب ۷۷.۷ رسیدم)

هفت.

با محاسبه‌ی Correlation ها به جدول زیر می‌رسیم که نشان می‌دهد ماکسیمم آن بین ستون‌های مختلف ۴۷ درصد است پس خیلی نمی‌توان کاهش بعد داد:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
age	1.000000	-0.090735	-0.048231	0.259415	0.223702	0.114791	-0.190592	-0.385695	0.120007	0.201417	-0.172787	0.281876	0.022636
sex	-0.090735	1.000000	-0.043990	-0.045832	-0.160064	0.126693	-0.054216	-0.066805	0.150150	0.076184	-0.017185	0.122036	0.187901
cp	-0.048231	-0.043990	1.000000	0.061022	-0.031860	0.131260	0.069756	0.269898	-0.353650	-0.074369	0.063440	-0.199414	-0.134143
trestbps	0.259415	-0.045832	0.061022	1.000000	0.110006	0.135483	-0.124200	-0.029285	0.078076	0.160412	-0.108051	0.077402	0.008513
chol	0.223702	-0.160064	-0.031860	0.110006	1.000000	-0.006783	-0.167585	0.008525	0.041301	0.038462	-0.011145	0.094042	0.089002
fbs	0.114791	0.126693	0.131260	0.135483	-0.006783	1.000000	-0.111671	0.016828	-0.006125	-0.069855	-0.080327	0.124839	-0.094289
restecg	-0.190592	-0.054216	0.069756	-0.124200	-0.167585	-0.111671	1.000000	0.094328	-0.043106	-0.033859	0.080450	-0.101146	0.019316
thalach	-0.385695	-0.066805	0.269898	-0.029285	0.008525	0.016828	0.094328	1.000000	-0.391052	-0.315468	0.393367	-0.231885	-0.077078
exang	0.120007	0.150150	-0.353650	0.078076	0.041301	-0.006125	-0.043106	-0.391052	1.000000	0.232045	-0.225611	0.112578	0.178249
oldpeak	0.201417	0.076184	-0.074369	0.160412	0.038462	-0.069855	-0.033859	-0.315468	0.232045	1.000000	-0.564095	0.220715	0.143848
slope	-0.172787	-0.017185	0.063440	-0.108051	-0.011145	-0.080327	0.080450	0.393367	-0.225611	-0.564095	1.000000	-0.095641	-0.071498
ca	0.281876	0.122036	-0.199414	0.077402	0.094042	0.124839	-0.101146	-0.231885	0.112578	0.220715	-0.095641	1.000000	0.121482
thal	0.022636	0.187901	-0.134143	0.008513	0.089002	-0.094289	0.019316	-0.077078	0.178249	0.143848	-0.071498	0.121482	1.000000

با ایتریت‌کردن روی k های مختلف برای knn به این می‌رسیم:

```
print("Best possible neighbors found for this dataset is %s", best_k)
print("Accuracy for this knn with %s neighbors is %s" % (best_k, max_acc))
```

```
Best possible neighbors found for this dataset is %s 5
Accuracy for this knn with 5 neighbors is 0.7704918032786885
```

و با انجام نایو بیز چندجمله‌ای به نتیجه‌ی زیر:

In [6]:
```
mnb = MultinomialNB()
mnb.fit(train, train_y)
```

Out[6]:
```
MultinomialNB()
```

In [7]:
```
nb_prediction = mnb.predict(test)
print("Accuracy for Naive Bayes is %s" % accuracy(nb_prediction, test_y))
```

```
Accuracy for Naive Bayes is 0.8032786885245902
```