

# 用python参加Kaggle的些许经验总结



JxKing (/u/e4ef312d540c) [+ 关注](#)

2016.05.07 23:29\* 字数 2014 阅读 19895 评论 7 喜欢 84

(/u/e4ef312d540c)

最近挤出时间，用python在kaggle上试了几个project，有点体会，记录下。

## Step1: Exploratory Data Analysis

EDA，也就是对数据进行探索性的分析，一般就用到pandas和matplotlib就够了。EDA一般包括：

1. 每个feature的意义，feature的类型,比较有用的代码如下

```
df.describe()
```

```
df['Category'].unique()
```

2. 看是否存在missing value

```
df.loc[df.Dates.isnull(),'Dates']
```

3. 每个特征下的数据分布，可以用boxplot或者hist来看

```
%matplotlib inline
```

```
import matplotlib.pyplot as plt
```

```
df.boxplot(column='Fare', by = 'Pclass')
```

```
plt.hist(df['Fare'], bins = 10, range =(df['Fare'].min(),df['Fare'].max()))
```

```
plt.title('Fare >distribution')
```

```
plt.xlabel('Fare')
```

```
plt.ylabel('Count of Passengers')
```



#如果变量是categorical的，想看distribution，则可以：

```
df.PdDistrict.value_counts().plot(kind='bar', figsize=(8,10))
```

4. 如果想看几个feature之间的联立情况，则可以用pandas的groupby,

```
temp = pd.crosstab([df.Pclass, df.Sex], df.Survived.astype(bool))
```

```
temp.plot(kind='bar', stacked=True, color=['red','blue'], grid=False)
```

(/apps/download?  
utm\_source=sbc)

在这步完成之后，要对以下几点有大致了解

- 理解每个特征的意义
- 要知道哪些特征是有用的，这些特征哪些是直接可以用的，哪些需要经过变换才能用，为之后的特征工程做准备

## Step2: Data Preprocessing

数据预处理，就是将数据处理下，为模型输入做准备，其中包括：

- 处理missing value：这里学问有点深，如果各位有好的经验可以跟我交流下。以我浅薄的经验来说我一般会分情况处理
  1. 如果missing value占总体的比例非常小，那么直接填入平均值或者众数
  2. 如果missing value所占比例不算小也不算大，那么可以考虑它跟其他特征的关系，如果关系明显，那么直接根据其他特征填入；也可以建立简单的模型，比如线性回归，随机森林等。
  3. 如果missing value所占比例大，那么直接将miss value当做一种特殊的情况，另取一个值填入
- 处理Outlier：这个就是之前EDA的作用了，通过画图，找出异常值
- 处理categorical feature：一般就是通过dummy variable的方式解决，也叫one hot encode，可以通过pandas.get\_dummies()或者 sklearn中 preprocessing.OneHotEncoder(), 我个人倾向于用pandas的get\_dummies()



看个例子吧，

```
In [43]: df.ix[:3, ['Month']]
```

```
Out[43]:
```

	Month
0	5
1	5
2	5
3	5

```
In [45]: month=pd.get_dummies(df.Month)
month.head(4)
```

```
Out[45]:
```

	1	10	11	12	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0	1	0	0	0	0
2	0	0	0	0	0	0	0	1	0	0	0	0
3	0	0	0	0	0	0	0	1	0	0	0	0

dummy variable

将一列的month数据展开为了12列，用0、1代表类别。

另外在处理categorical feature有两点值得注意：

1. 如果特征中包含大量需要做dummy variable处理的，那么很可能导致得到一个**稀疏**的dataframe，这时候最好用下**PCA**做降维处理。
2. 如果某个特征有好几个取值，那么用dummy variable就并不现实了，这时候可以用**Count-Based Learning** (<https://link.jianshu.com?t=https://msdn.microsoft.com/en-us/library/azure/dn913056.aspx>).

(/apps/download?  
utm\_source=sbc)

3. (更新) 近期在kaggle成功的案例中发现, 对于类别特征, 在模型中加入tf-idf总是有效果的。
4. 还有个方法叫“Leave-one-out” encoding, 也可以处理类别特征种类过多的问题, 实测效果不错。

(/apps/download?  
utm\_source=sbc)

## Step 3: Feature Engineering

理论上来说, 特征工程应该也归属于上一步, 但是它太重要了, 所以将它单独拿出来。kaggle社区对特征工程的重要性已经达成了共识, 可以说最后结果的好坏, 大部分就是由**特征工程**决定的, 剩下部分应该是**调参**和**Ensemble**决定。特征工程的好坏主要是由**domain knowledge**决定的, 但是大部分人可能并不具备这种知识, 那么只能尽可能多的根据原来feature生成新的feature, 然后让模型选择其中重要的feature。这里就又涉及到**feature selection**,

有很多方法, 比如backward, forward selection等等。我个人倾向于用**random forest**的**feature importance**, 这里 (<https://link.jianshu.com?t=https://hal.archives-ouvertes.fr/hal-00755489/file/PRLv4.pdf>)有论文介绍了这种方法。

## Step 4: Model Selection and Training

- 最常用的模型是**Ensemble Model**, 比如 **Random Forest, Gradient Boosting**。当然在开始的时候, 可以用点简单的模型, 一方面是可以作为底线threshold, 另一方面也可以在最后作为Ensemble Model。

当然还有大名鼎鼎的**xgboost** (<https://link.jianshu.com?t=https://github.com/dmlc/xgboost>), 这个我还没有深入的研究, 只是简单的用python调用了下, 接下来如果有时间, 要好好深入研究下。

- 选择完模型之后, 就是要训练模型, 主要就是调参, 每种模型都有自己最关键的几个参数, sklearn中**GridSearchCV** ([https://link.jianshu.com?t=http://scikit-learn.org/stable/modules/generated/sklearn.grid\\_search.GridSearchCV.html](https://link.jianshu.com?t=http://scikit-learn.org/stable/modules/generated/sklearn.grid_search.GridSearchCV.html))可以设置需要比较的几种参数组合, 然后用**cross validation**来选出最优秀的参数组合。大概用法为:

```
from sklearn.grid_search import GridSearchCV
```

```
from pprint import pprint
clf=RandomForestClassifier(random_state=seed)
parameters = {'n_estimators': [300, 500], 'max_features':[4,5,'auto']}
grid_search = GridSearchCV(estimator=clf,param_grid=parameters, cv=10,
scoring='accuracy')
print("parameters:")
pprint(parameters)
grid_search.fit(train_x,train_y)
print("Best score: %0.3f" % grid_search.best_score_)
print("Best parameters set:")
best_parameters=grid_search.best_estimator_.get_params()
for param_name in sorted(parameters.keys()):
    print("\t%s: %r" % (param_name, best_parameters[param_name]))
```

(/apps/download?  
utm\_source=sbc)

## Step 5: Model Ensemble

Model Ensemble有**Bagging**,**Boosting**,**Stacking**,其中Bagging和Boosting都算是**Bootstrapping**的应用。**Bootstrapping**的概念是对样本每次有放回的抽样，抽样K个，一共抽N次。

- **Bagging**:每次从总体样本中随机抽取K个样本来训练模型，重复N次，得到N个模型，然后将各个模型结果合并，分类问题投票方式结合，回归则是取平均值,e.g.Random Forest。
- **Boosting**:一开始给每个样本取同样的权重，然后迭代训练，每次对训练失败的样本调高其权重。最后对多个模型用加权平均来结合,e.g. GBDT。
- **Bagging与Boosting的比较**：在深入理解Bagging和Boosting后发现，bagging其实是用相同的模型来训练随机抽样的数据，这样的结果是各个模型之间的bias差不多，variance也差不多，通过平均，使得variance降低（由算平均方差的公式可知），从而提高ensemble model的表现。而Boosting其实是一种贪心算法，不断降低bias。



- **Stacking**: 训练一个模型来组合其他各个模型。首先先训练多个不同的模型，然后再以之前训练的各个模型的输出为输入来训练一个模型，以得到一个最终的输出。使用过stacking之后，发现其实stacking很像神经网络，通过很多模型的输出，构建中间层，最后用逻辑回归讲中间层训练得到最后的结果。这里贴一个例子供参考。

(/apps/download?  
utm\_source=sbc)

```
def single_model_stacking(clf):
    skf = list(StratifiedKFold(y, 10))
    dataset_blend_train = np.zeros((Xtrain.shape[0], len(set(y.tolist()))))
    # dataset_blend_test = np.zeros((Xtest.shape[0], len(set(y.tolist()))))
    dataset_blend_test_list=[]
    loglossList=[]
    for i, (train, test) in enumerate(skf):
        # dataset_blend_test_j = []
        X_train = Xtrain[train]
        y_train = dummy_y[train]
        X_val = Xtrain[test]
        y_val = dummy_y[test]
        if clf=='NN_fit':
            fold_pred, pred=NN_fit(X_train, y_train, X_val, y_val)
        if clf=='xgb_fit':
            fold_pred, pred=xgb_fit(X_train, y_train, X_val, y_val)
        if clf=='lr_fit':
            fold_pred, pred=lr_fit(X_train, y_train, X_val, y_val)
        print('Fold %d, logloss:%f'%(i, log_loss(y_val, fold_pred)))
        dataset_blend_train[test, :] = fold_pred
        dataset_blend_test_list.append( pred )
        loglossList.append(log_loss(y_val, fold_pred))
    dataset_blend_test = np.mean(dataset_blend_test_list, axis=0)
    print('average log loss is :', np.mean(log_loss(y_val, fold_pred)))
    print ("Blending.")
    clf = LogisticRegression(multi_class='multinomial', solver='lbfgs')
    clf.fit(dataset_blend_train, np.argmax(dummy_y, axis=1))
    pred = clf.predict_proba(dataset_blend_test)
    return pred
```

## Step 6: Two Little Tips

最后是我的两点心得吧

- 设置random seed，使得你的模型reproduce，以Random Foreset举例：  
seed=0

```
clf=RandomForestClassifier(random_state=seed)
```

- 每个project组织好文件层次和布局，既方便与其他人交流，也方便自己。比如在一个project下，分设3个文件夹，一个是input，放训练数据、测试数据，一个model，放模型文件，最后一个submission文件，放你生成要提交的结果文件。

具体的可以参考这里 ([https://link.jianshu.com?](https://link.jianshu.com?t=https://www.kaggle.com/wiki/ModelSubmissionBestPractices)

[t=https://www.kaggle.com/wiki/ModelSubmissionBestPractices](https://www.kaggle.com/wiki/ModelSubmissionBestPractices))

(/apps/download?  
utm\_source=sbc)

## 最后的回顾和展望

这篇文章是参加kaggle之后的第一次总结，描述了下kaggle的步骤，通用的知识点和技巧。希望在未来一个月中，能把xgboost和stacking研究应用下，然后再来update。希望大家有什么想法都能跟我交流下~~

update: 更新了关于类别特征的处理方式以及Boosting和Bagging的看法，还有stacking的内容。

📖 学习体会 (/nb/3748710)

举报文章 © 著作权归作者所有



JxKing (/u/e4ef312d540c)

写了 13140 字，被 171 人关注，获得了 141 个喜欢

(/u/e4ef312d540c)

+ 关注

数据挖掘

♥ 喜欢 (/sign\_in?utm\_source=desktop&utm\_medium=not-signed-in-like-button) | 84



nshu.io/notes/images/3725095/weibo/image\_e  
(/apps/download?  
utm\_source=sbc)

(/apps/download?utm\_source=nbc)

被以下专题收入，发现更多相似内容



首页投稿 (/c/bDHhpK?utm\_source=desktop&utm\_medium=notes-included-collection)



程序员 (/c/NEt52a?utm\_source=desktop&utm\_medium=notes-included-collection)



我是程序员；您... (/c/abe194e18e78?

utm\_source=desktop&utm\_medium=notes-included-collection)



生活不易 我用... (/c/8c01bfa7b98a?

utm\_source=desktop&utm\_medium=notes-included-collection)



数据挖掘 (/c/aedd722949ae?utm\_source=desktop&utm\_medium=notes-included-collection)



机器学习 (/c/dfc68565d16f?utm\_source=desktop&utm\_medium=notes-included-collection)



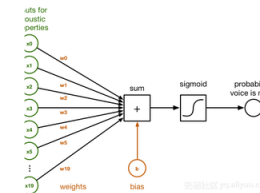
收藏-Pyth... (/c/fd81ec2767d6?utm\_source=desktop&utm\_medium=notes-included-collection)

展开更多 ∨





(/p/b370ac791613?




(/apps/download?  
utm\_source=sbc)

utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

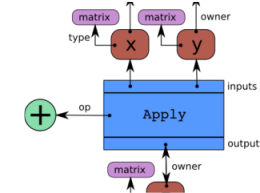
## IOS平台TensorFlow实践 (/p/b370ac791613?utm\_campaign=maleskine...

1 天前 作者简介：MATTHIJS HOLLEMANS 荷兰人，独立开发者，专注于底层编码，GPU优化和算法研究。目前研究方向为IOS上的深度学习及其在APP上的应用。推特地址：https://twitter.com/mhollemans ...

 阿里云云栖社区 (/u/12532d36e4da?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)


(/p/769f47377a3?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

## 李理：Theano tutorial和卷积神经网络的Theano实现 Part1 (/p/769f47377...


本系列文章面向深度学习研发者，希望通过Image Caption Generation，一个有意思的具体任务，深入浅出地介绍深度学习的知识。本系列文章涉及到很多深度学习流行的模型，如CNN，RNN/LSTM，Attention等...

 IMGeek (/u/cf291f021d93?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

## 结合Scikit-learn介绍几种常用的特征选择方法 (/p/bbcdcb983ab3?utm\_ca...

结合Scikit-learn介绍几种常用的特征选择方法 作者：Edwin Jarvis 特征选择(排序)对于数据科学家、机器学习从业者来说非常重要。好的特征选择能够提升模型的性能，更能帮助我们理解数据的特点、底层结构，...

 阿甘run (/u/c8be94c66af1?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/abc4e084bdbd?

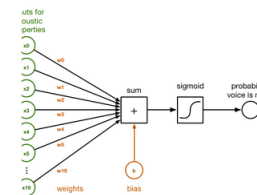
utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

## 深度学习指南：在iOS平台上使用TensorFlow (/p/abc...

在利用深度学习网络进行预测性分析之前，我们首先需要对其加以训练。目前市面上存在着大量能够用于神经网络训练的工具，但TensorFlow无疑是其中极...



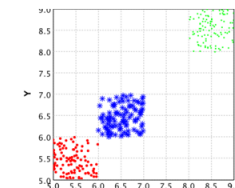
BURIBURI\_ZAEMON (/u/00d1ed2b53ae?)



(/apps/download?  
utm\_source=sbc)

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/1194b6f80240?)



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

## 面向开发人员的机器学习指南 (/p/1194b6f80240?utm\_campaign=maleski...

首页 资讯 文章 资源 小组 相亲 登录 注册 首页 最新文章 IT 职场 前端 后端 移动端 数据库 运维 其他技术 - 导航条 - 首页最新文章IT 职场前端- JavaScript- HTML5- CSS后端- Python- Java- C/C++- PHP- .NE...



Helen\_Cat (/u/58977be48437?)

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

## 约定 (/p/bf3edf592c1b?utm\_campaign=maleskine&utm\_content=note&...

跨过山川 越过河流 我和冬天有个约定 从雨季走向另一个雨季 横跨四个四季 你来了 我也到了 我不问你来自何方 你也不要问我的归处 就让我们一起 颠覆整个苍穹



柏浅歌 (/u/2b1927ff94d9?)

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

## kali升级系统之后遇到的错误 (/p/0c6f3f878b42?utm\_campaign=maleskin...

0x01 N: Ignoring file '50unattended-upgrades.ucf-dist' in directory '/etc/apt/apt.conf.d/' as it has an invalid filename extension 在Debia...



g0 (/u/eea6acda4b25?)

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

## 雅思官方范文【教育类词汇&短语】01 (/p/24525aa915ba?utm\_campaign=...

警告！微信公众号虽然大多为“快销型”，其目的在于提供信息。但本号不同，其目的在于增进理解（也就是学习）。所以，这个系列当中的每一篇文章，对于一个写作有志于突破6分的同学而言，都值得在上面投入1...



Philip\_Dai (/u/127516da0b49?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation) (/apps/download?utm\_source=sbc)

(/p/3a00bfee1ddd?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

## 2016金丹若国际微电影艺术节网络影视创新论坛暨合作伙伴大会圆满落幕 (/...

2016年12月15-16日，由中国电视艺术家协会、中央新影集团、陕西文化产业投资基金、金丹若（北京）品牌运营管理有限公司、西安大奥影视传媒股份有限公司主办的金丹若国际微电影艺术节网络影视论坛暨合...



影视潜规则 (/u/9e1244b1647e?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/3e7ed9affd77?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

## 一个人的好天气 (/p/3e7ed9affd77?utm\_campaign=maleskine&utm\_cont...

和一个人在一起，如果他给你的能量是让你每天都能高兴的起床，每夜都能安心的入睡，做每一件事都充满了动力，对未来充满期待，那你就没有爱错人。真正的爱，永远都不是以爱的名义互相折磨，而是彼此陪...



麻花辫儿 (/u/1bcd01ef772b?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

