Practical Advice on Machine Learning Projects

Not included in (most) textbooks

MF1633055
杨润琦

General Process

- Problem Setup
- Data Exploration
- Data cleaning
- Feature Construction
- Feature Transformation
- Feature selection
- Architecture Design
- Model Tuning
- Error Analysis

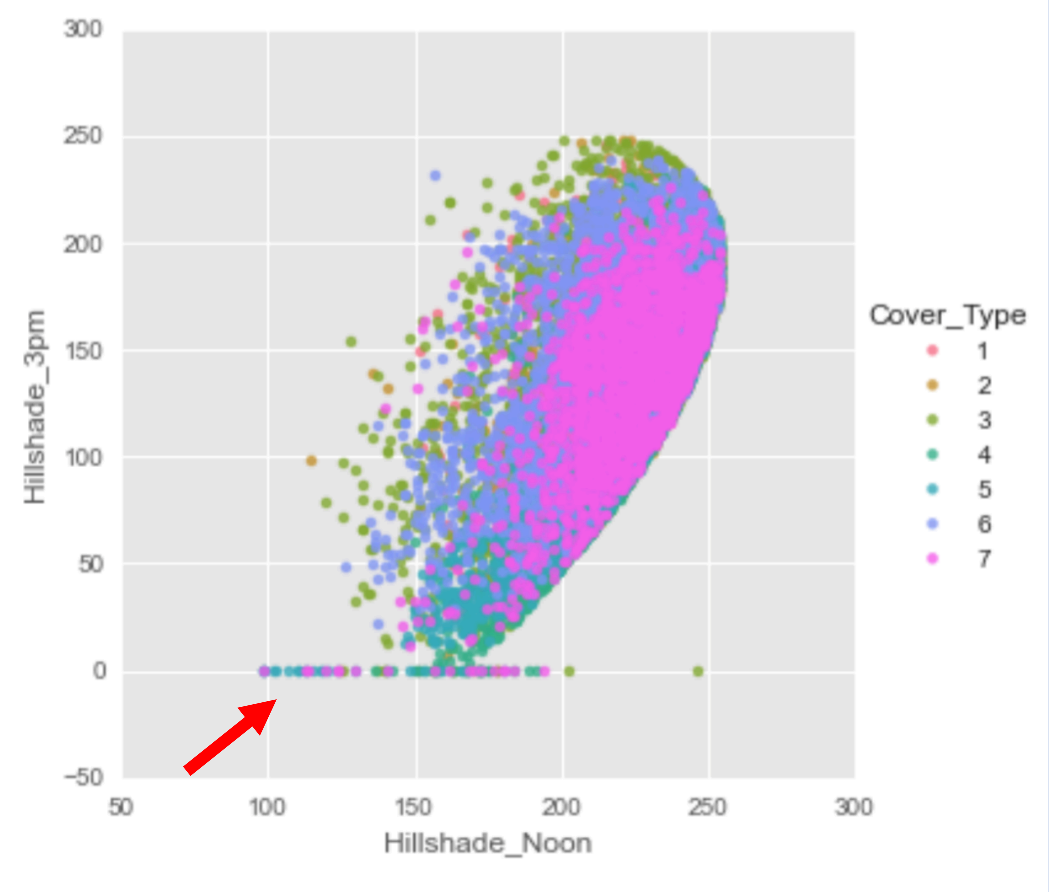
> “ Any intelligent fool can make things bigger and more complex.
> It takes a touch of genius, and a lot of courage, to move in the opposite direction. ”

Before the competition begins….

- Identify the problem
- Collect the appropriate data
- split the data into training and test datasets

- **Problem Setup**
- Data Exploration
- Data cleaning
- Feature Construction
- Data Transformation
- Feature selection
- Architecture Design
- Model Tuning
- Error Analysis

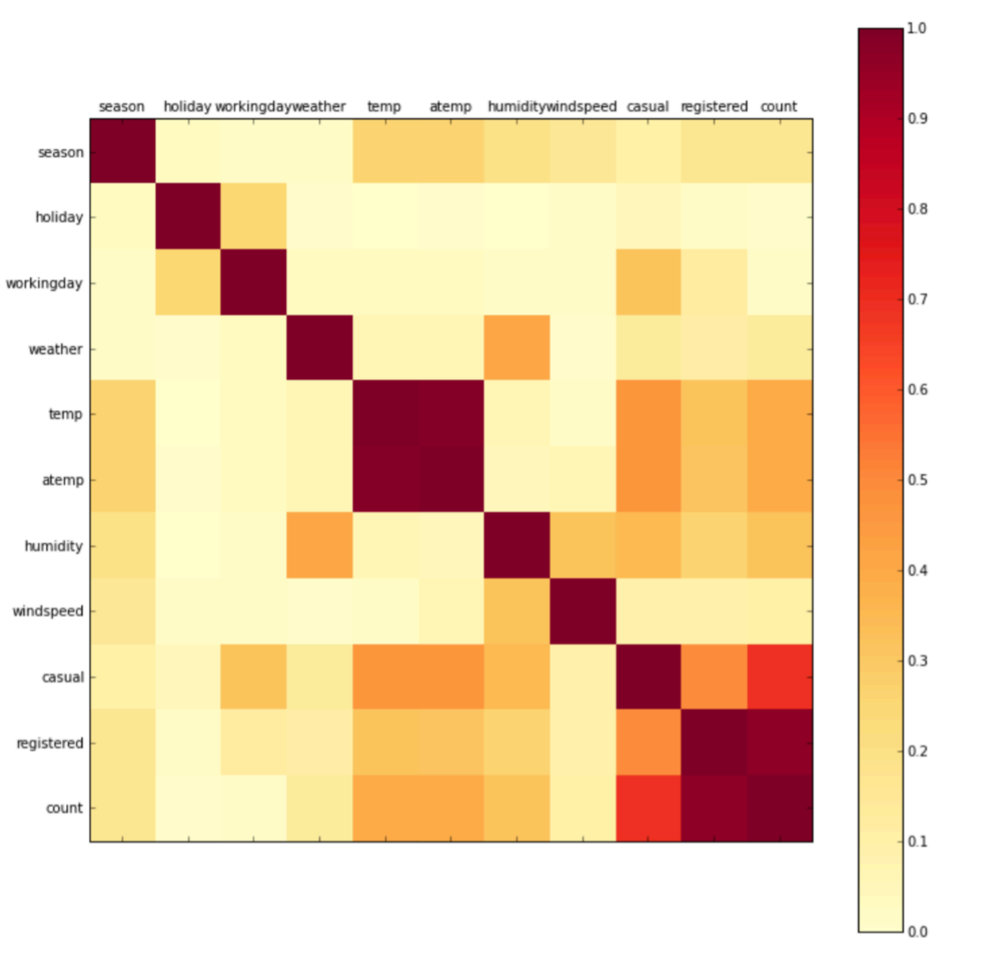
No standard way, but…



```
sns.jointplot(x='attr1', y='attr2', data=X,
kind='scatter', hue='label')
```

- Problem Setup
- **Data Exploration**
- Data cleaning
- Feature Construction
- Data Transformation
- Feature selection
- Architecture Design
- Model Tuning
- Error Analysis

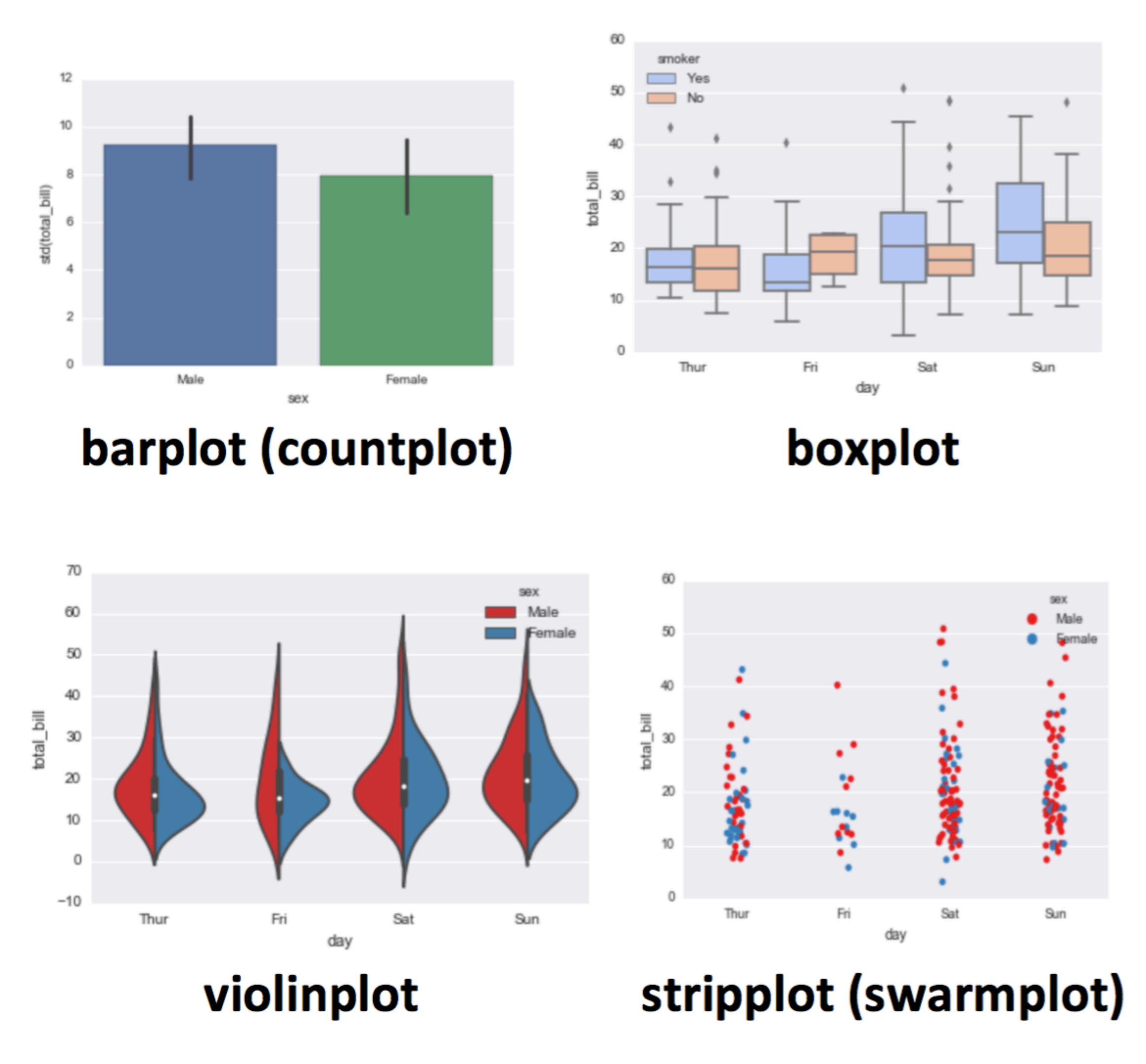
No standard way, but…



```
sns.heatmap(data.corr())
```

- Problem Setup
- **Data Exploration**
- Data cleaning
- Feature Construction
- Data Transformation
- Feature selection
- Architecture Design
- Model Tuning
- Error Analysis

No standard way, but…



barplot (countplot)

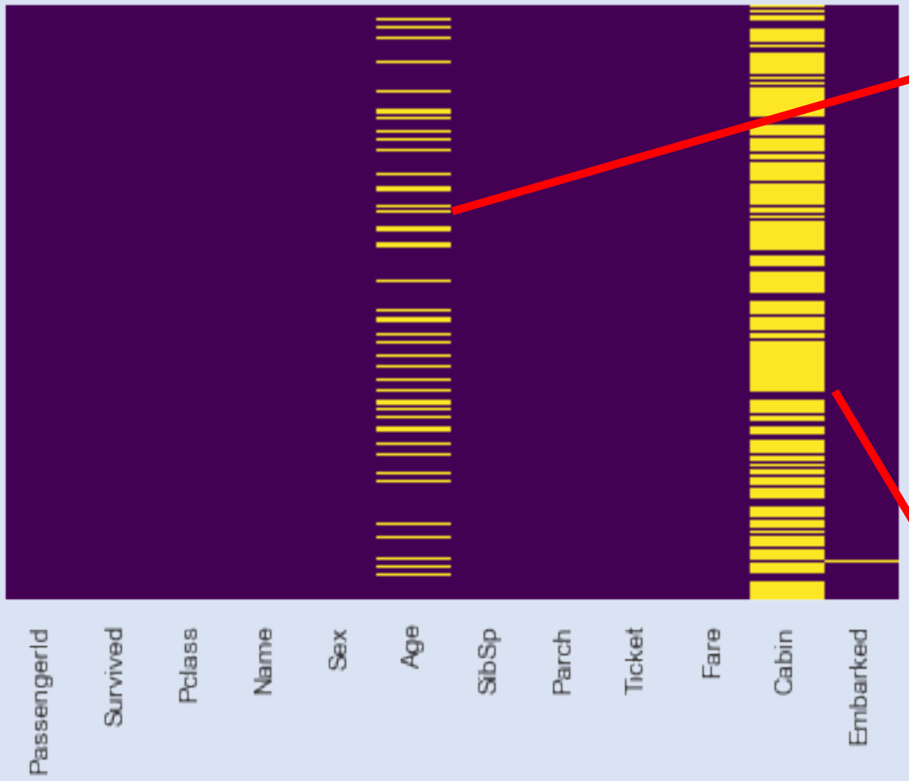
boxplot

violinplot

stripplot (swarmplot)

- Problem Setup
- **Data Exploration**
- Data cleaning
- Feature Construction
- Data Transformation
- Feature selection
- Architecture Design
- Model Tuning
- Error Analysis

Fill in missing values



```
sns.heatmap(X.isnull())
```

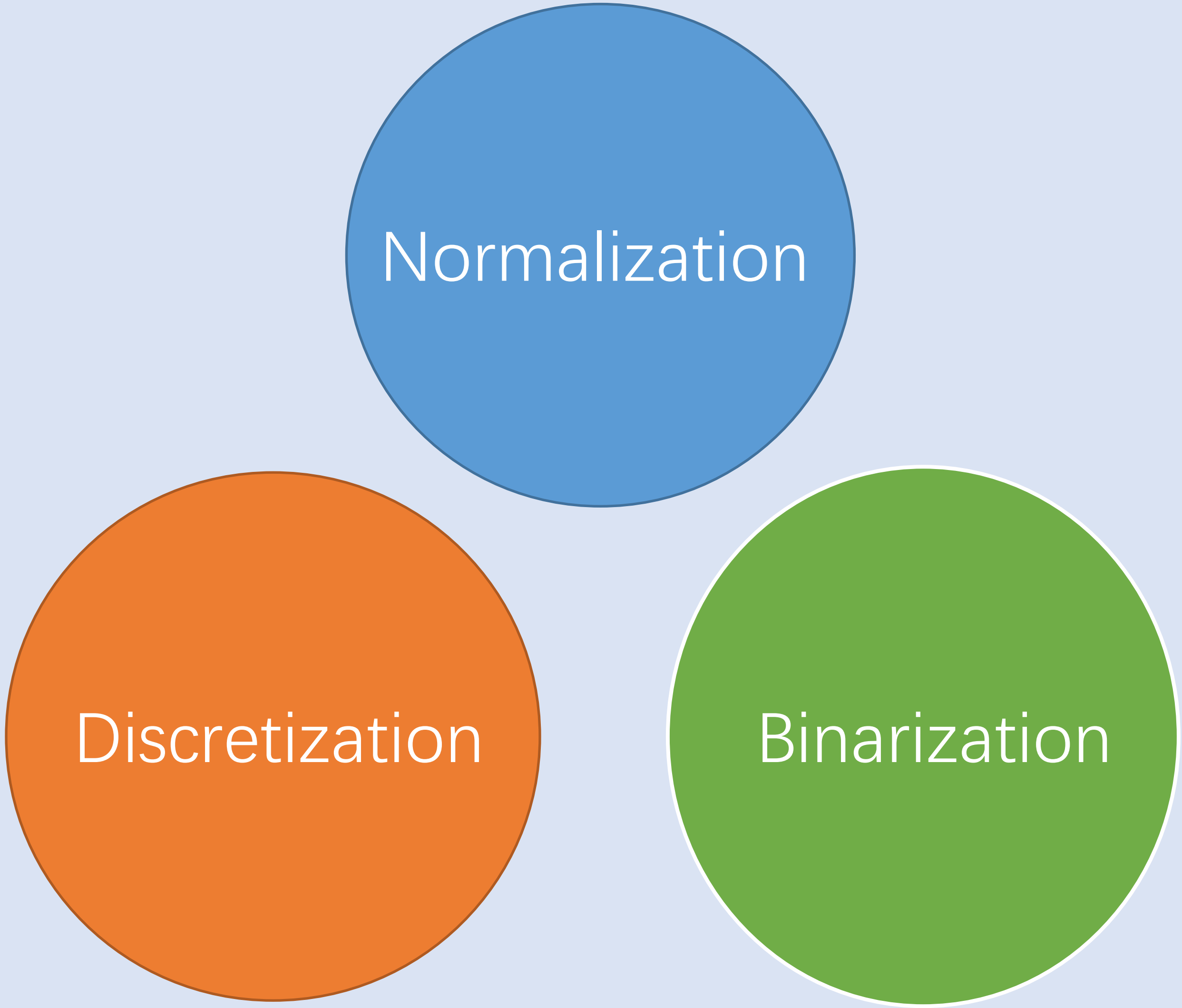
- Use attribute mean, tiered average…
- Build a model
  - test/dev data can be included

- Drop the column
- change to another feature: known/unknown

- Problem Setup
- Data Exploration
- **Data cleaning**
- Feature Construction
- Data Transformation
- Feature selection
- Architecture Design
- Model Tuning
- Error Analysis

Once you spot a trend/pattern in data exploration, try to convert it to a feature.

- Problem Setup
- Data Exploration
- Data cleaning
- **Feature Construction**
- Data Transformation
- Feature selection
- Architecture Design
- Model Tuning
- Error Analysis



- Problem Setup
- Data Exploration
- Data cleaning
- Feature Construction
- **Data Transformation**
- Feature selection
- Architecture Design
- Model Tuning
- Error Analysis

做法
腹黑
dnf手游
减肥
破解版
炮灰
宠文
手工
重生
女生
装备
洪荒
资格证
by
孕
苍穹
耽美
组词
斗鱼
最强
宠妃
显卡
主宰
都市
英语
守望
孕妇
宠
微微
韩剧
王者
np
lol
护肤品
吃
暖
书包
cf
超级
郑爽
老公
火线
微微一笑
火影忍者
王爷
口红
很
穿
灰姑娘
传说
总裁
男主
幼儿
适合
王妃
宫
会计
白带
女配
化妆品
电视剧
怀孕
宠妻
s6
云画
儿童
豪门
先锋
海贼王
番外
高手
联盟
手抄报
月经
妃
梦见
简笔画
教师
教案
步步
文
英雄
惊心
宝宝
微盘
三国
荣耀
女主
武器
动漫
美人
面膜
酷跑
幼儿园
胎儿
孩子
摩托车
公主

Generated using data from Sogou User Grouping Contest, 2016

- Problem Setup
- Data Exploration
- Data cleaning
- Feature Construction
- Data Transformation
- **Feature selection**
- Architecture Design
- Model Tuning
- Error Analysis

Ensemble

- voting
- stacking

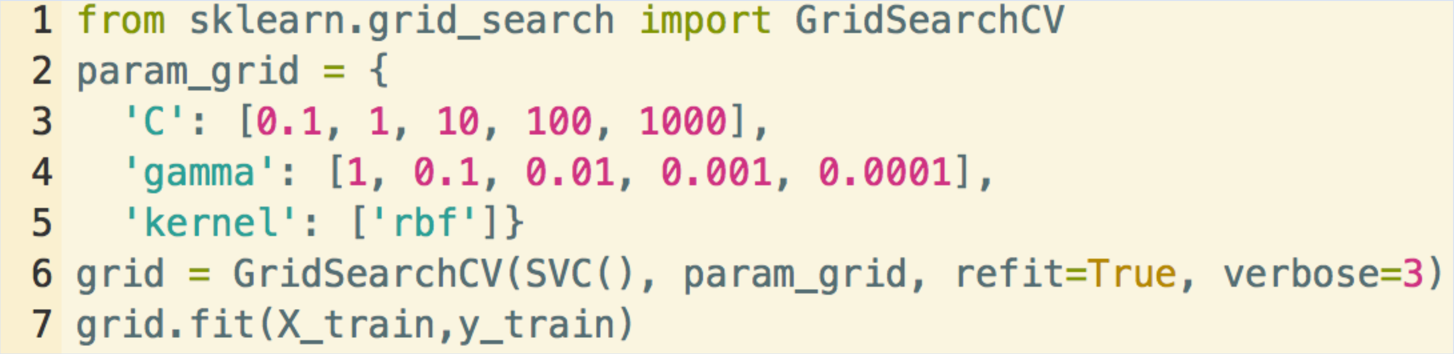
Clustering

- train a model in each cluster
- use cluster label as a new feature

Much more…

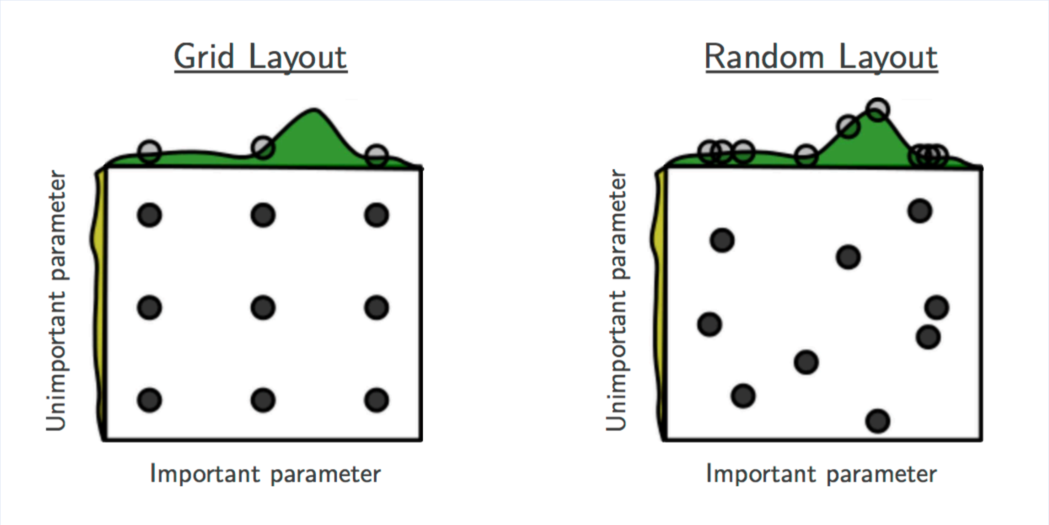
- Problem Setup
- Data Exploration
- Data cleaning
- Feature Construction
- Data Transformation
- Feature selection
- **Architecture Design**
- Model Tuning
- Error Analysis

Grid Search



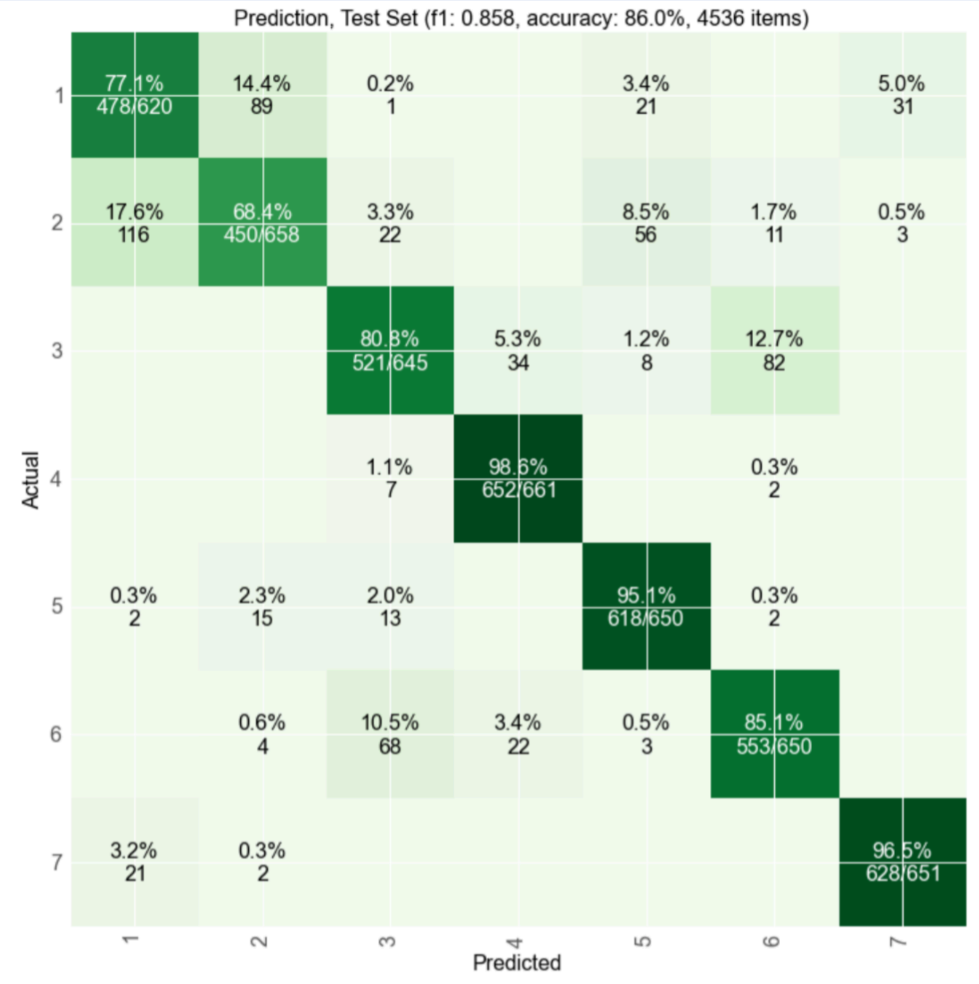
```python
from sklearn.grid_search import GridSearchCV
param_grid = {
  'C': [0.1, 1, 10, 100, 1000],
  'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
  'kernel': ['rbf']}
grid = GridSearchCV(SVC(), param_grid, refit=True, verbose=3)
grid.fit(X_train,y_train)
```

Random Search



- Problem Setup
- Data Exploration
- Data cleaning
- Feature Construction
- Data Transformation
- Feature selection
- Architecture Design
- **Model Tuning**
- Error Analysis

Classification
Matrix Plot of Confusion Matrix

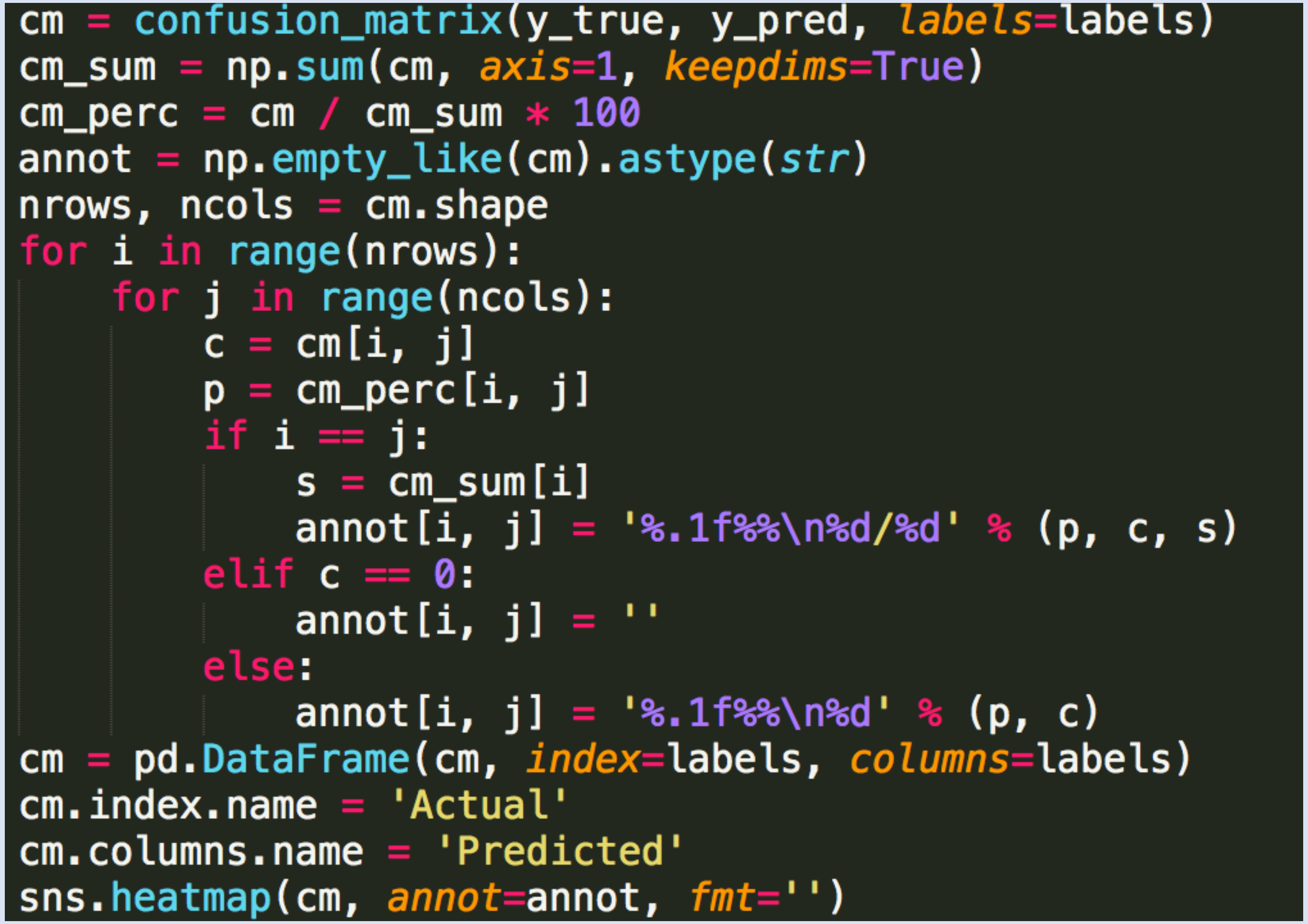


Prediction, Test Set (f1: 0.858, accuracy: 86.0%, 4536 items)

	1	2	3	4	5	6	7
1	77.1% 478/620	14.4% 89	0.2% 1		3.4% 21		5.0% 31
2	17.6% 116	68.4% 450/658	3.3% 22		8.5% 56	1.7% 11	0.5% 3
3			80.8% 521/645	5.3% 34	1.2% 8	12.7% 82	
4			1.1% 7	98.6% 652/661		0.3% 2	
5	0.3% 2	2.3% 15	2.0% 13		95.1% 618/650	0.3% 2	
6		0.6% 4	10.5% 68	3.4% 22	0.5% 3	85.1% 553/650	
7	3.2% 21	0.3% 2					96.5% 628/651

- Problem Setup
- Data Exploration
- Data cleaning
- Feature Construction
- Data Transformation
- Feature selection
- Architecture Design
- Model Tuning
- **Error Analysis**

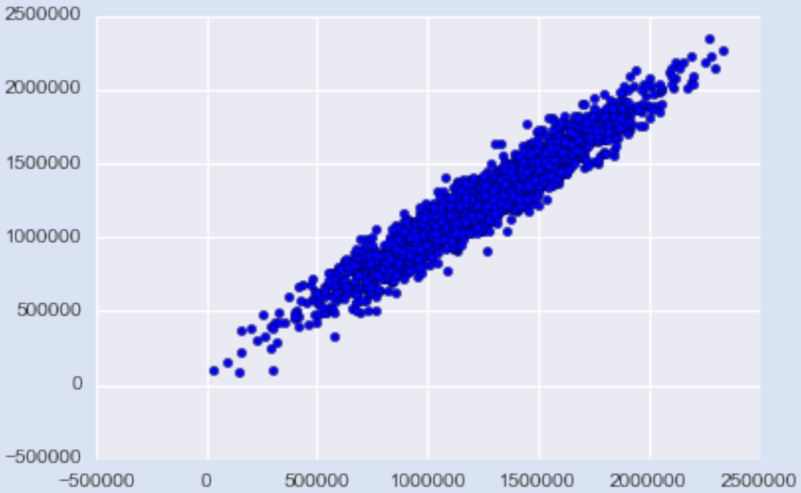
Matrix Plot of Confusion Matrix (Code)



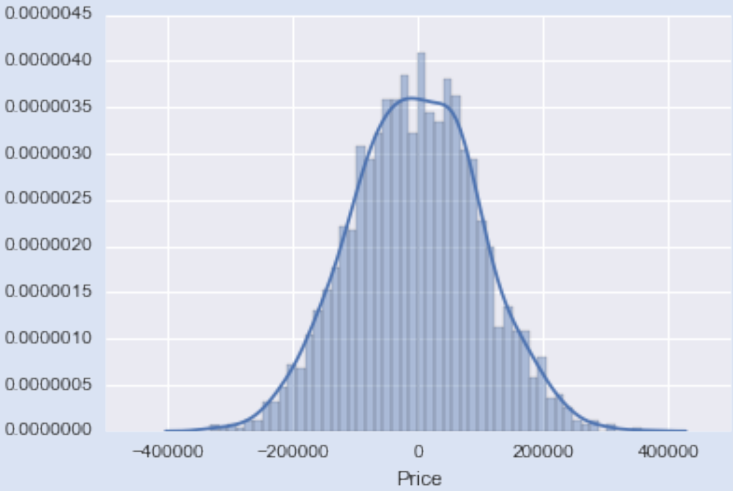
```
cm = confusion_matrix(y_true, y_pred, labels=labels)
cm_sum = np.sum(cm, axis=1, keepdims=True)
cm_perc = cm / cm_sum * 100
annot = np.empty_like(cm).astype(str)
nrows, ncols = cm.shape
for i in range(nrows):
    for j in range(ncols):
        c = cm[i, j]
        p = cm_perc[i, j]
        if i == j:
            s = cm_sum[i]
            annot[i, j] = '%.1f%%\n%d/%d' % (p, c, s)
        elif c == 0:
            annot[i, j] = ''
        else:
            annot[i, j] = '%.1f%%\n%d' % (p, c)
cm = pd.DataFrame(cm, index=labels, columns=labels)
cm.index.name = 'Actual'
cm.columns.name = 'Predicted'
sns.heatmap(cm, annot=annot, fmt='')
```

- Problem Setup
- Data Exploration
- Data cleaning
- Feature Construction
- Data Transformation
- Feature selection
- Architecture Design
- Model Tuning
- **Error Analysis**

Regression



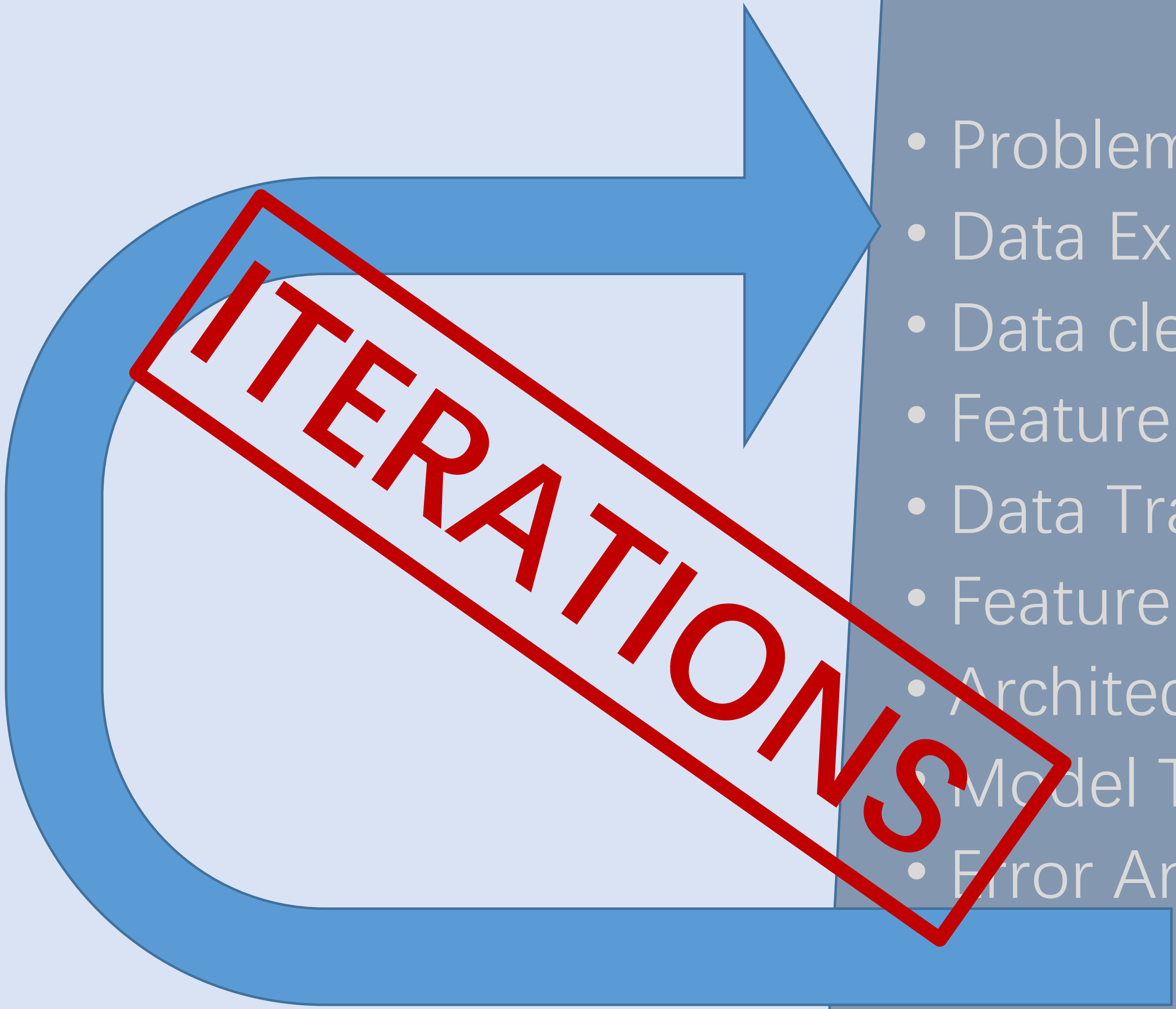
```
plt.scatter(y_dev, predictions)
```



```
sns.distplot(
  (y_dev-predictions),
  bins=50)
```

- Problem Setup
- Data Exploration
- Data cleaning
- Feature Construction
- Data Transformation
- Feature selection
- Architecture Design
- Model Tuning
- **Error Analysis**

SUCCESS LIES IN



ITERATIONS

- Problem Setup
- Data Exploration
- Data cleaning
- Feature Construction
- Data Transformation
- Feature selection
- Architecture Design
- Model Tuning
- Error Analysis

Q&A

Thanks for your time!

MF1633055
杨润琦