Fundamental Method of Binary Classification

Hsu An, Du Yixuan, Xiong Yi
Neptunus Team

Nov 7, 2018

任务整体评估

任务整体评估

任务类型

评价方法

任务整体评估

任务类型

Binary Classification

评价方法

F1-Score

Step 1 数据探索

Step 2 数据清洗&预处理

Step 3 模型选择

Step 4 模型优化

Step 1　数据探索

Overall Preview

命令: Dateframe.info(), Dataframe.describe()

Step 1　数据探索　**Overall Preview**

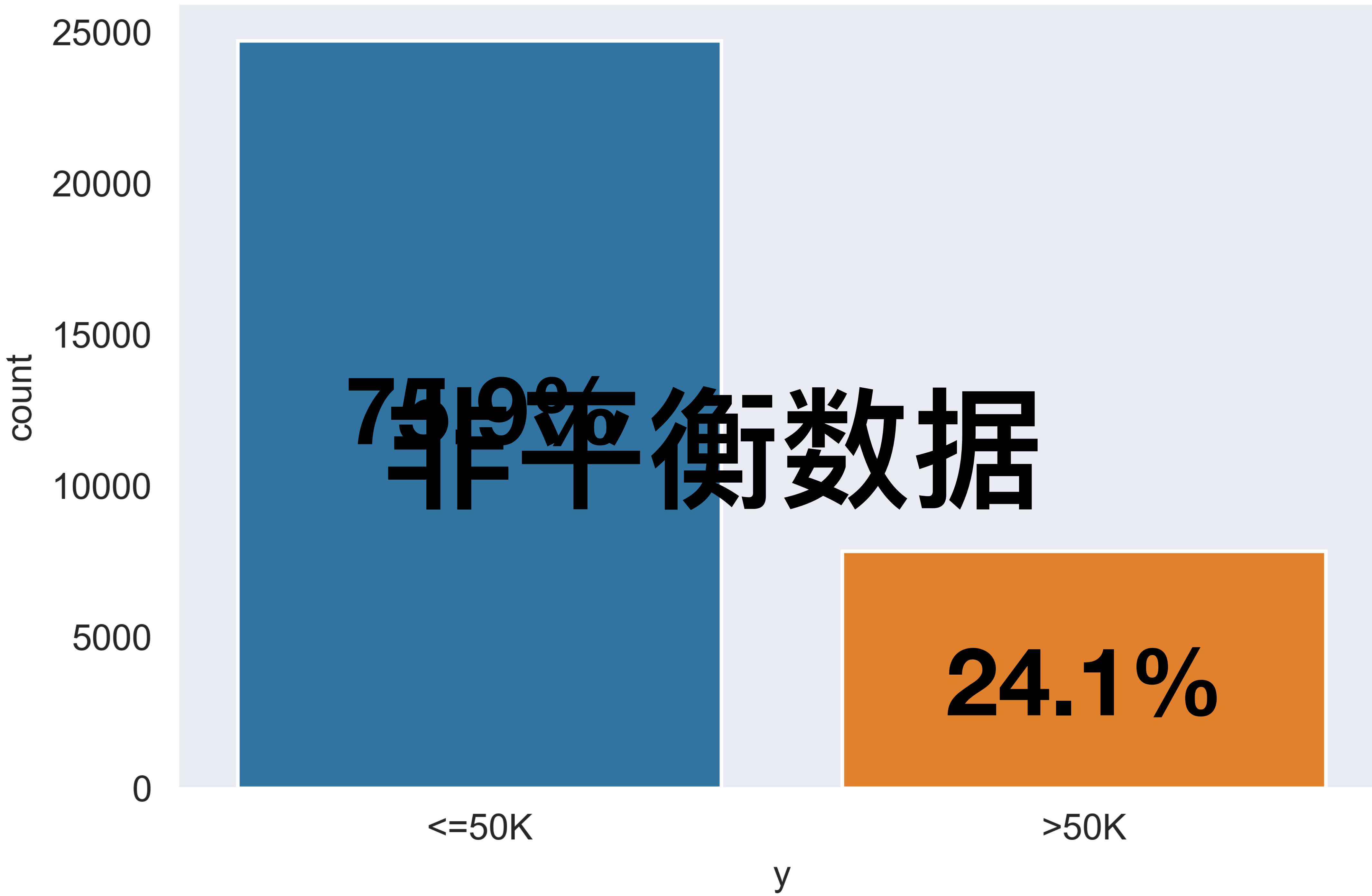
Data Columns	Data	count	mean	std	Min	25%	50%	75%	max
age	Int64	32561.0	38.6	13.6	17.0	28.0	37.0	48.0	90.0
workclass	object								
fnlwgt	Int64	32561.0	189778.4	105550.0	12285.0	117827.0	178356.0	237051.0	1484705.0
education	object								
education_num	Int64	32561.0	10.1	2.6	1.0	9.0	10.0	12.0	16.0
marital_status	object								
occupation	object								
relationship	object								
race	object								
sex	object								
capital_gain	Int64	32561.0	1077.7	7385.3	0.0	0.0	0.0	0.0	99999.0
capital_loss	Int64	32561.0	87.3	403.0	0.0	0.0	0.0	0.0	4356.0
hours_per_week	Int64	32561.0	40.4	40.4	1.0	40.0	40.0	45.0	99.0
native_country	object								
y	object								

Step 1　数据探索

Y-label overview

Step 1　数据探索

Y-label overview



Step 1 数据探索

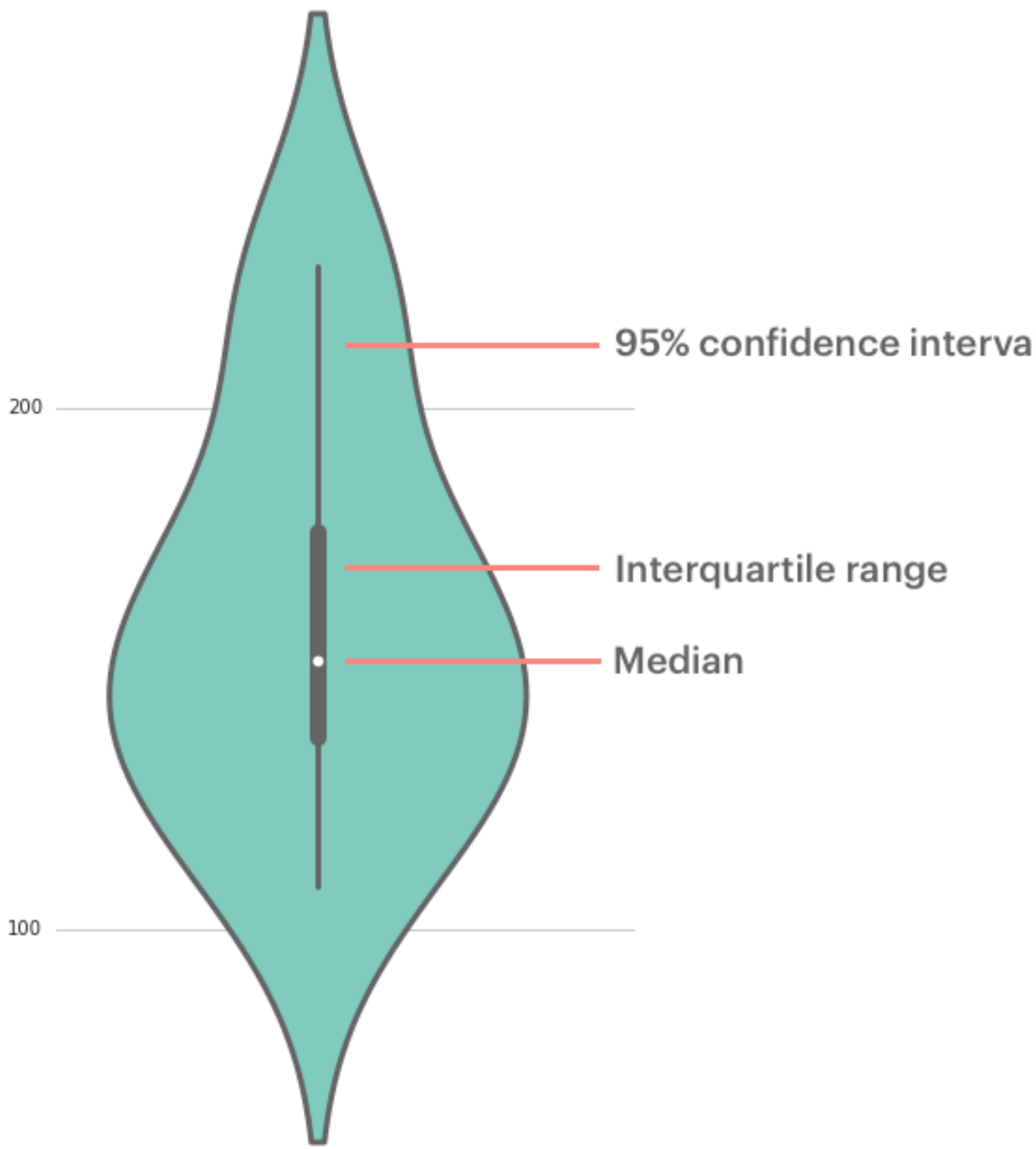
数值型连续变量

Numerical Variable

e.g. age, fnlwgt, capital_gain, capital_loss…

Step 1　数据探索

Numerical Variable

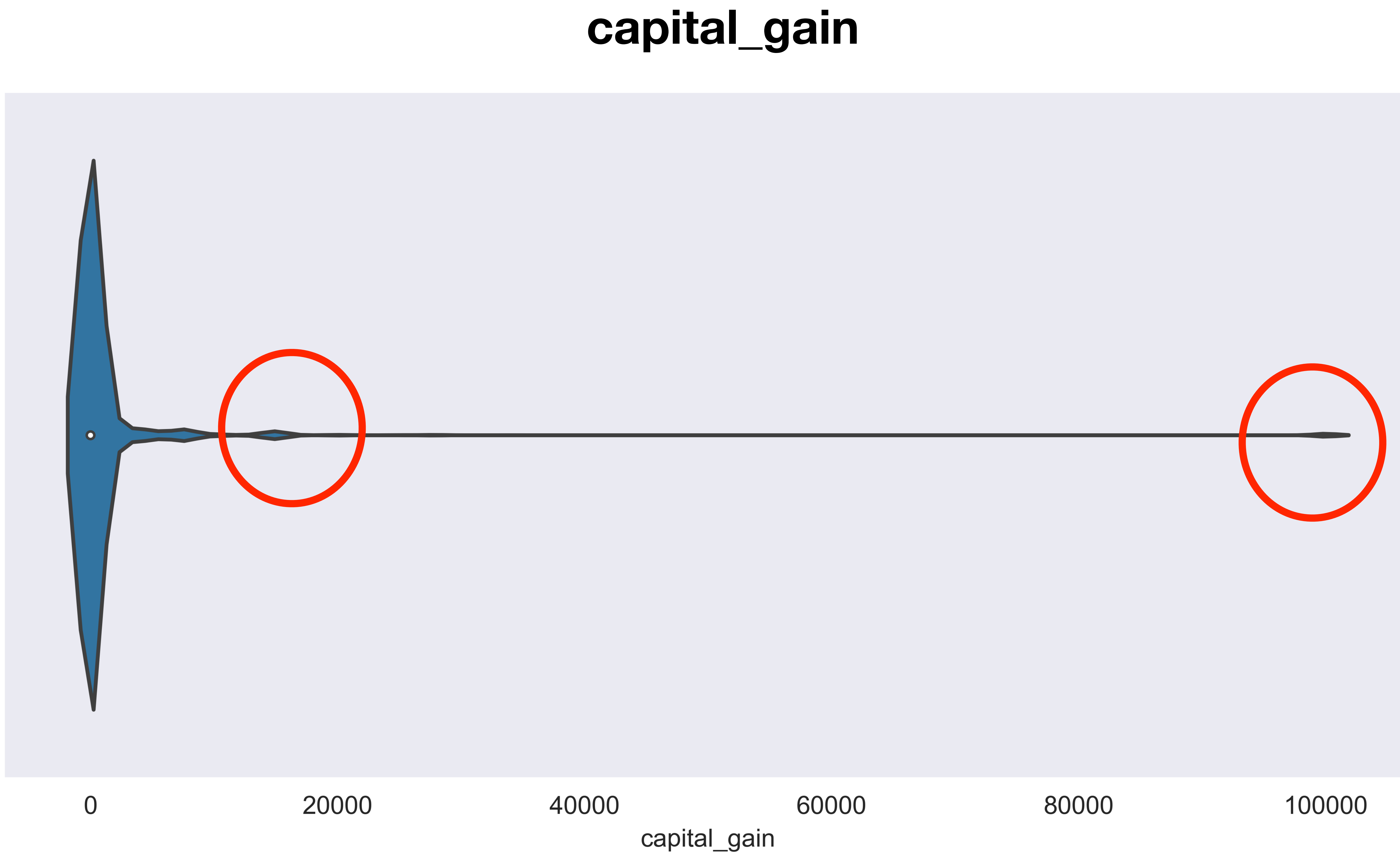


琴箱图

Violinplot

Step 1 数据探索

Numerical Variable



类别型离散变量

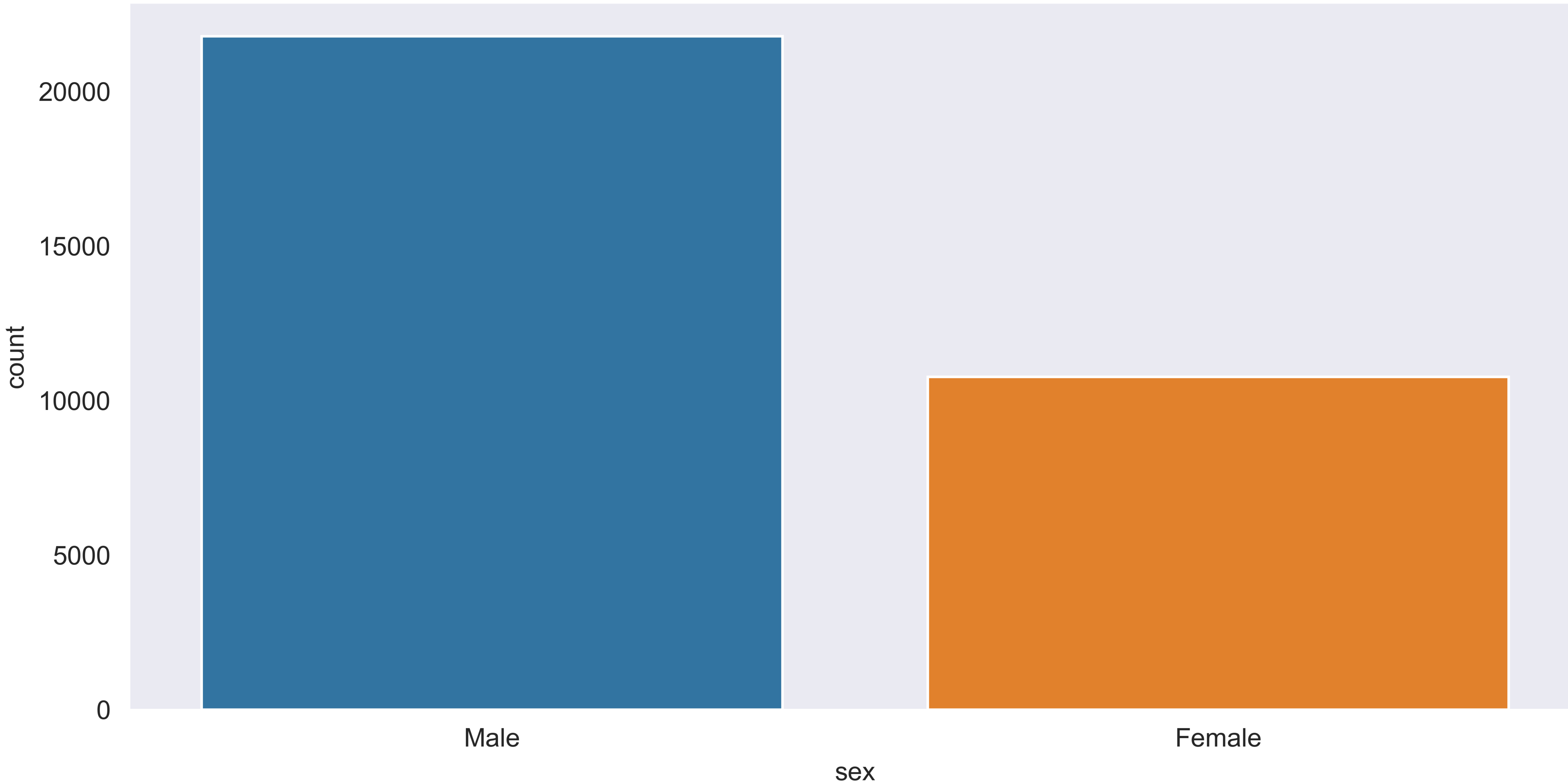
Categorical Variable

e.g. workclass, education, marital_status…

Step 1 数据探索

Categorical Variable

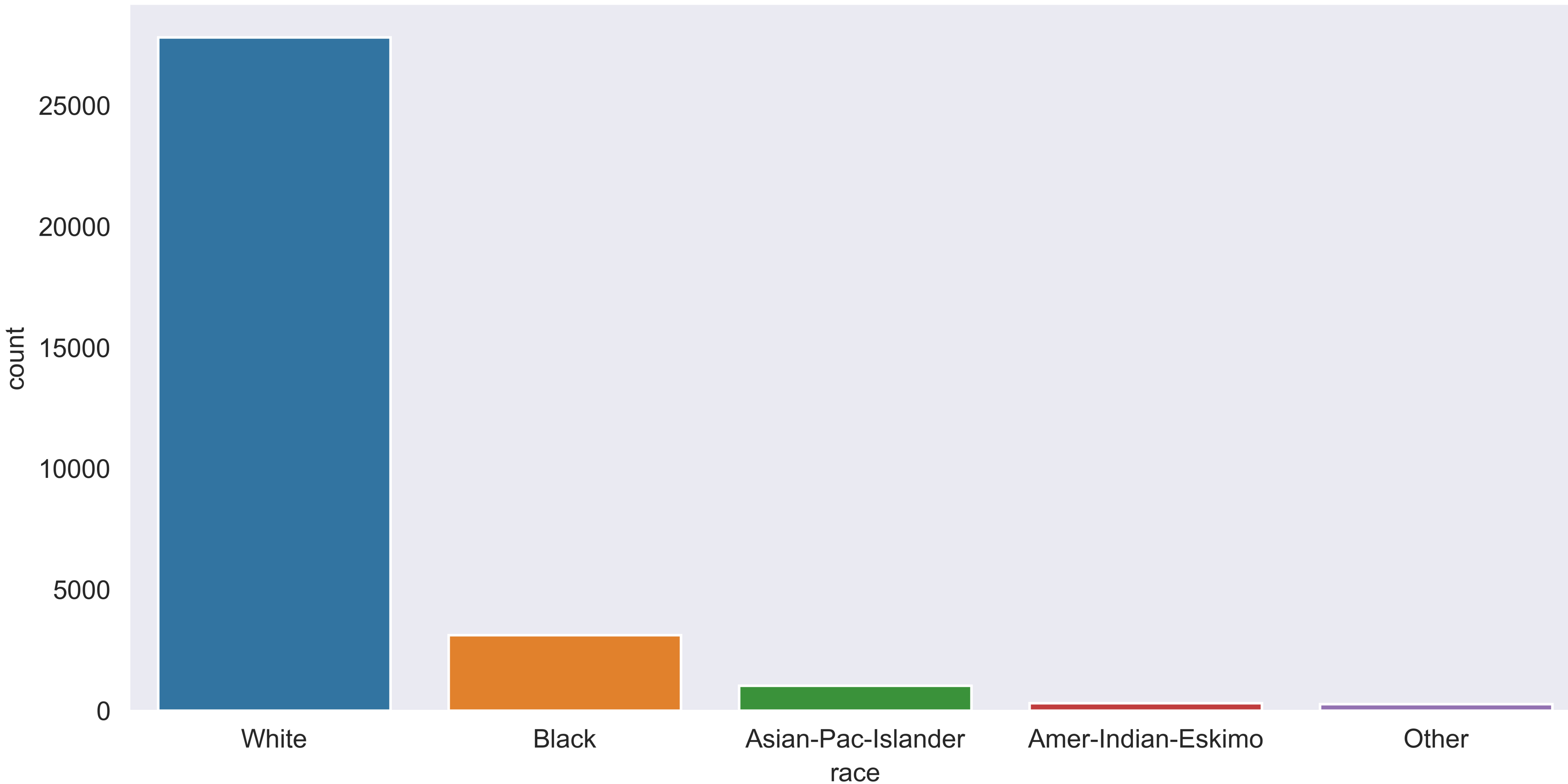
Sex (Binary-category)



Step 1 数据探索

Categorical Variable

Race (Multi-category)



相关度分析

Correlation analysis

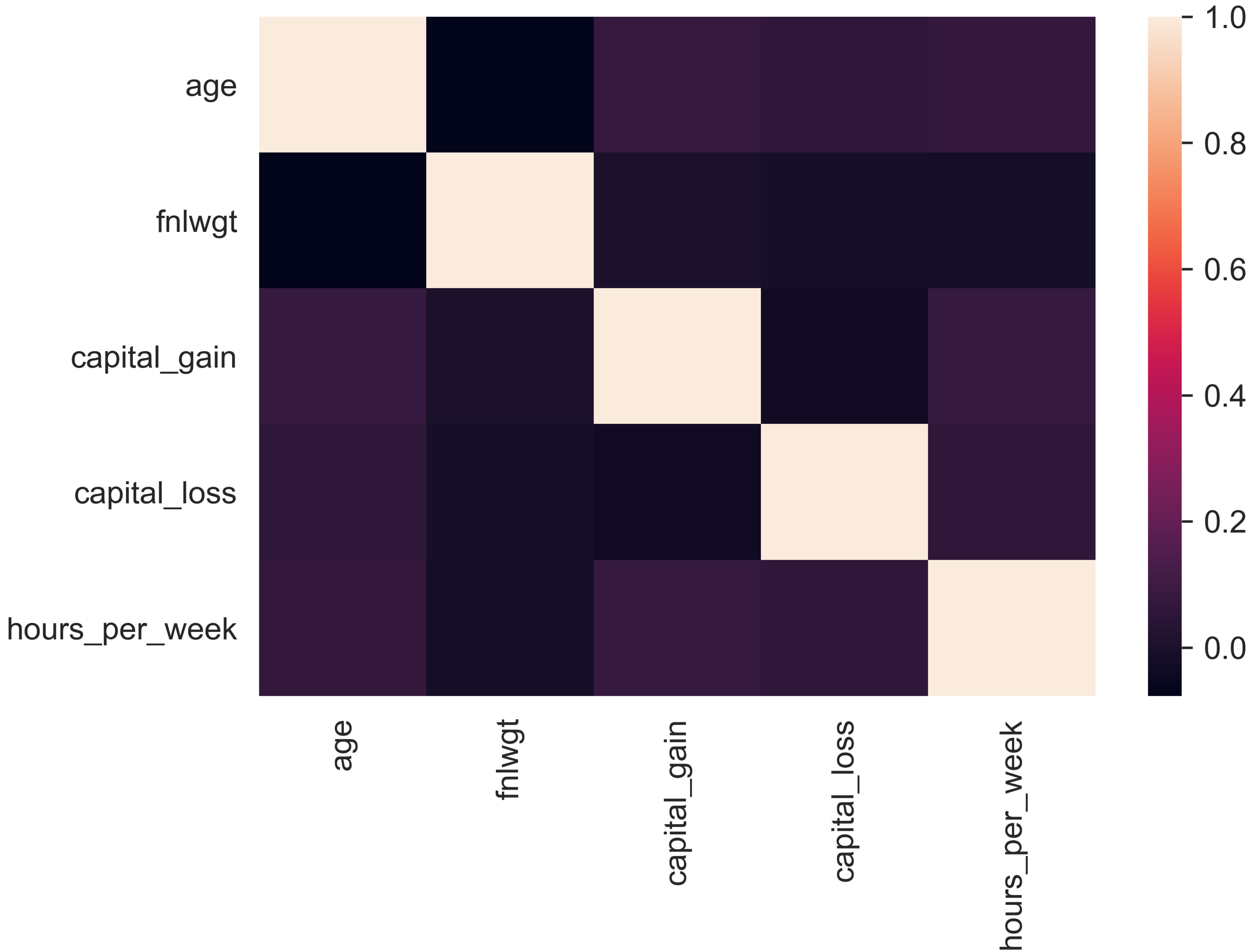
Step 1 数据探索 **Correlation analysis**

Pearson correlation coefficient

$$r = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \overline{Y})^2}}$$

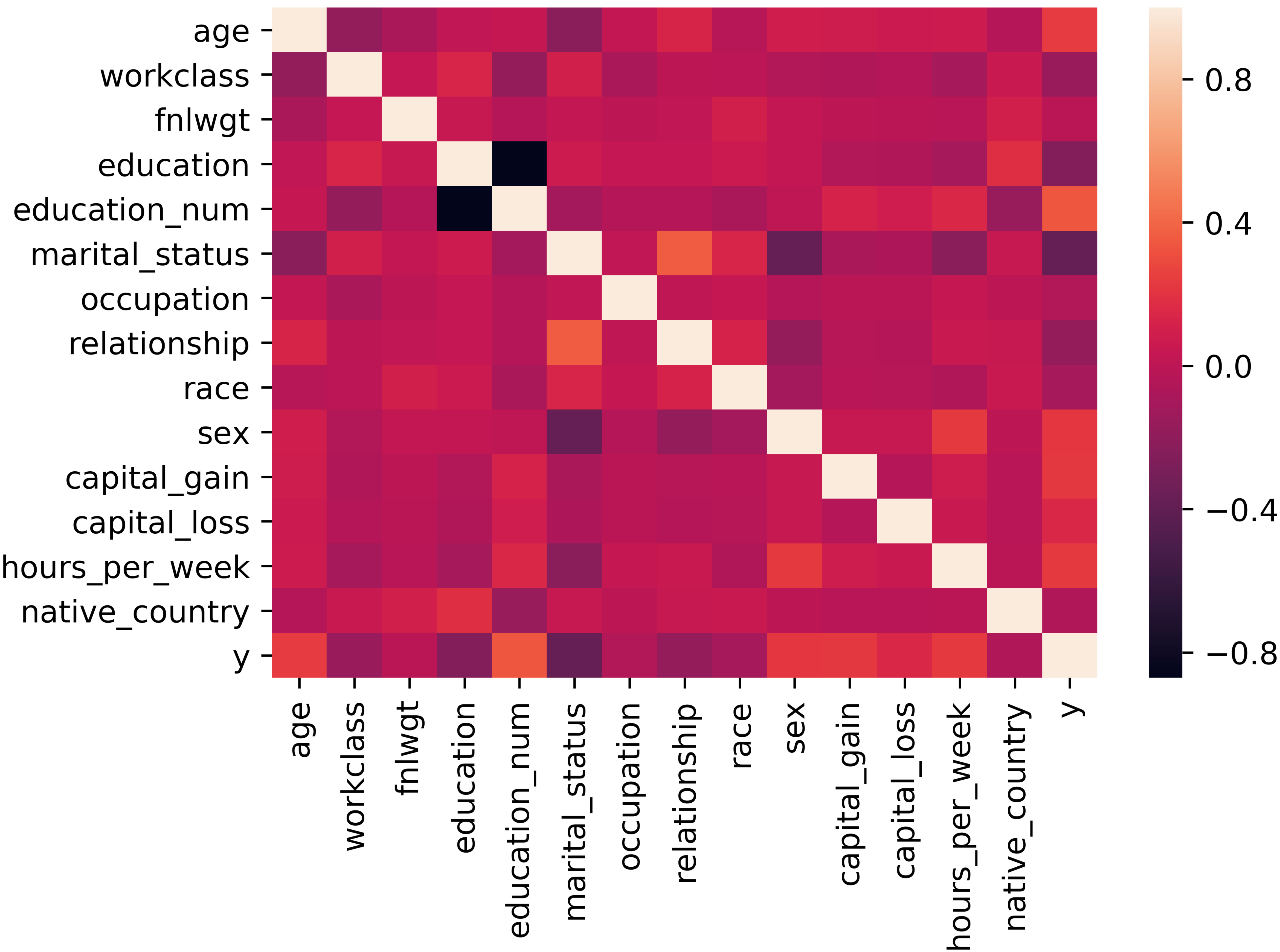
Step 1 数据探索

Correlation analysis



Step 1 数据探索

Correlation analysis



Step 1 数据探索

Step 2 数据清洗&预处理

Step 3 模型选择

Step 4 模型优化

缺失值处理

Missing Data

平均值（Mean）

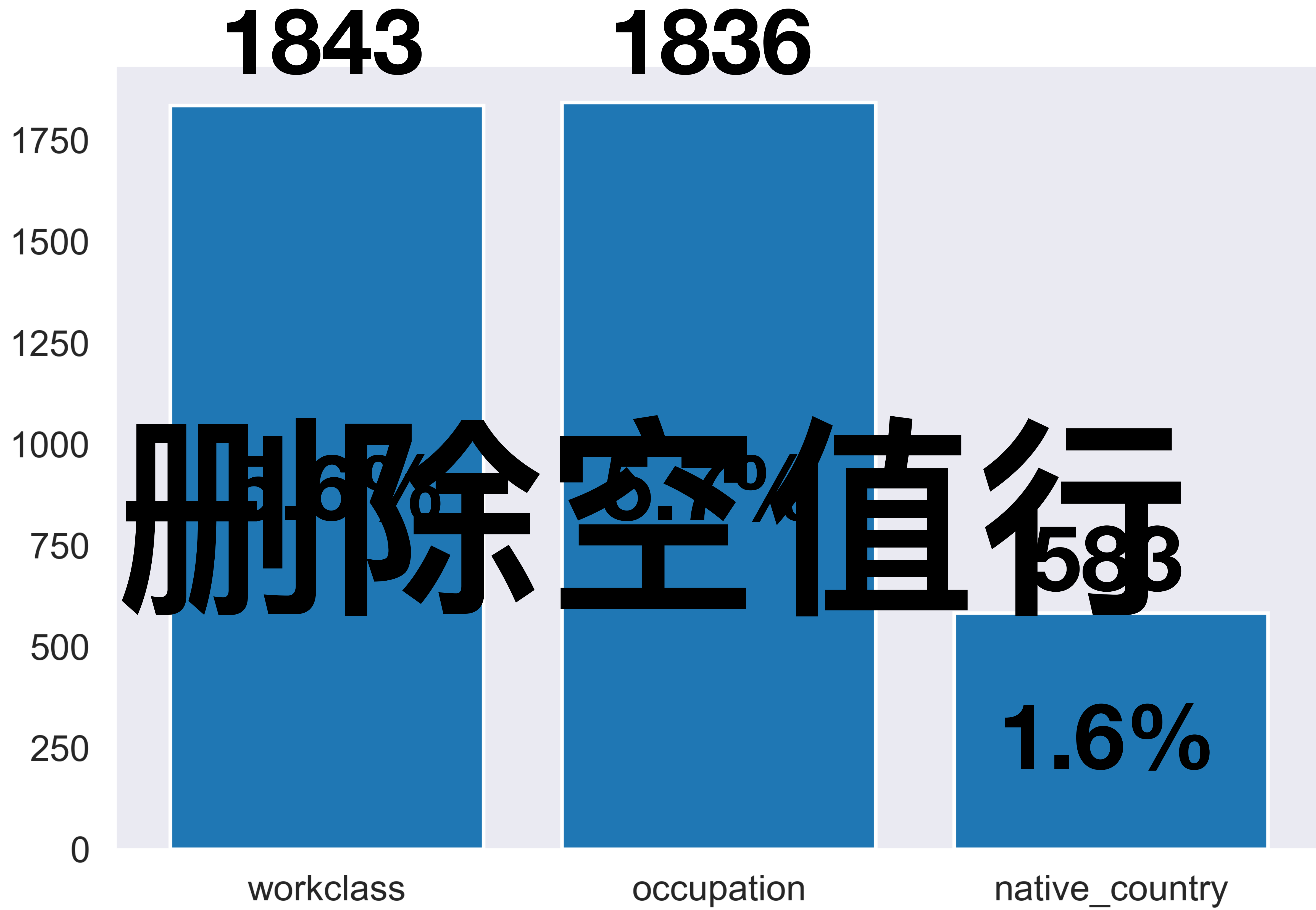
众数（Mode）

中位数（Median）

成为新的类别（Unknown）

Step 2 数据清洗&预处理

Missing Data



删除空值行

变量编码

Variables Encoding

Step 2 数据清洗&预处理

Variables Encoding

Data Columns	Data	count	mean	std	Min	25%	50%	75%	max
age	Int64	32561.0	38.6	13.6	17.0	28.0	37.0	48.0	90.0
workclass	object								
fnlwgt	Int64	32561.0	189778.4	105550.0	12285.0	117827.0	178356.0	237051.0	1484705.0
education	object								
education_num	Int64	32561.0	10.1	2.6	1.0	9.0	10.0	12.0	16.0
marital_status	object								
occupation	object								
relationship	object								
race	object								
sex	object								
capital_gain	Int64	32561.0	1077.7	7385.3	0.0	0.0	0.0	0.0	99999.0
capital_loss	Int64	32561.0	87.3	403.0	0.0	0.0	0.0	0.0	4356.0
hours_per_week	Int64	32561.0	40.4	40.4	1.0	40.0	40.0	45.0	99.0
native_country	object								
y	object								

Step 2 数据清洗&预处理

Variables Encoding

数据行	数据类型	编码值域
workclass	object	0, 1 …. 7, np.nan
education	object	0, 1 … 15
marital_status	object	0, 1 … 6
occupation	object	0, 1 … 13, np.nan
relationship	object	0, 1 … 12
race	object	0, 1 … 15
sex	object	0, 1
native_country	object	0, 1 … 40, np.nan
y	object	0, 1

非平衡数据处理

Un-Balanced Samples

欠采样（Under-sampling）

ClusterCentroids, RandomUnderSampler, NearMiss, EditedNearestNeighbours

过采样（Over-sampling）

RandomOverSampler,SMOTE,ADASYN

过采样算法SMOTE

设训练集少数类样本数为T,则SMOTE至少合成NT个样本（N为正整数）

考虑该少数类的一个样本 i，其特征向量为：$\boldsymbol{x}_i, i \in \{1, \ldots, T\}$

1. 首先从该少数类的全部 T 个样本中找到样本 xi 的 k 个近邻（例如用欧氏距离），记为 $\boldsymbol{x}_{i(near)}, near \in \{1, \ldots, k\}$

2. 然后从这 k 个近邻中随机选择一个样本 xi(nn)，再生成一个 0 到 1 之间的随机数 ζ1，从而合成一个新样本 xi1：

$$\boldsymbol{x}_{i1} = \boldsymbol{x}_i + \zeta_1 \cdot (\boldsymbol{x}_{i(nn)} - \boldsymbol{x}_i)$$

3. 将步骤2重复进行 N 次，从而可以合成 N 个新样本：xinew,new∈1,...,N。

那么，对全部的 T 个少数类样本进行上述操作，便可为该少数类合成 NT 个新样本。

标准化（Standardization）

归一化（Normalization）

（不适用）

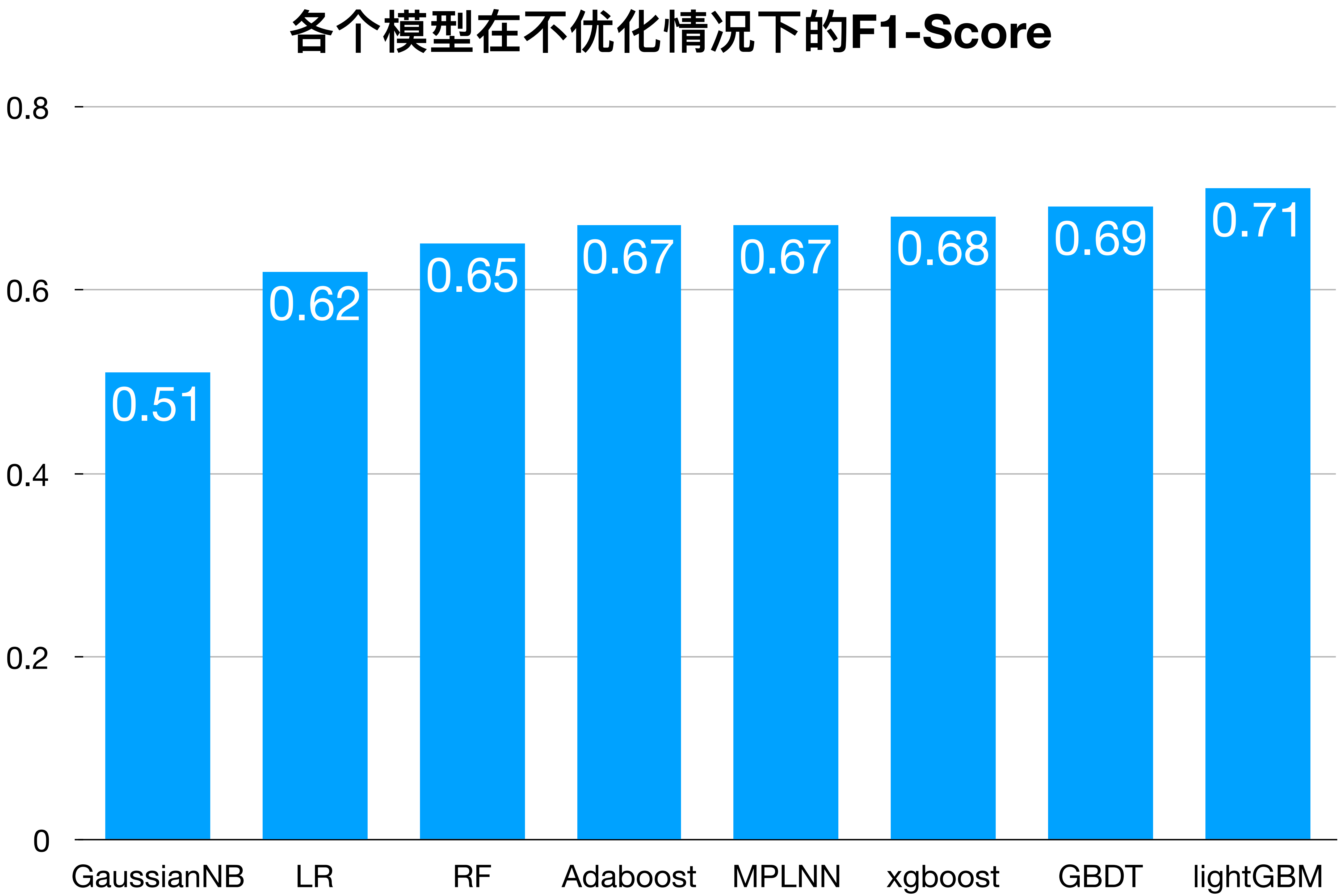
Step 1　数据探索

Step 2　数据清洗&预处理

Step 3　模型选择

Step 4　模型优化

Step 3　模型选择


各个模型在不优化情况下的F1-Score
0.8
0.6
0.4
0.2
0
0.51
0.62
0.65
0.67
0.67
0.68
0.69
0.71
GaussianNB
LR
RF
Adaboost
MPLNN
xgboost
GBDT
lightGBM

Step 3　模型选择

lightGBM等树状模型

Step 1 数据探索

Step 2 数据清洗&预处理

Step 3 模型选择

Step 4 模型优化

Step 4　模型优化

对于lightGBM,有如下关键参数：

`max_depth` ：设置树深度，深度越大可能过拟合

`num_leaves`：树的复杂程度。设置应该小于 2^(max_depth)，否则可能会导致过拟合。

`min_data_in_leaf` :它的值取决于训练数据的样本个数.

Step 4　模型优化

对于lightGBM,有如下关键参数：

`learning_rate` ：设置学习率

`lambda_l1(reg_alpha), lambda_l2(reg_lambda)`:正则化参数.降低过拟合，两者分别对应l1正则化和l2正则化

Step 4 模型优化

Core Parameters

- `config` □, default = `""`, type = string, aliases: `config_file`
  - path of config file
  - **Note**: can be used only in CLI version
- `task` □, default = `train`, type = enum, options: `train`, `predict`, `convert_model`, `refit`, aliases: `task_type`
  - `train`, for training, aliases: `training`
  - `predict`, for prediction, aliases: `prediction`, `test`
  - `convert_model`, for converting model file into if-else format, see more information in IO Parameters
  - `refit`, for refitting existing models with new data, aliases: `refit_tree`
  - **Note**: can be used only in CLI version; for language-specific packages you can use the correspondent functions
- `objective` □, default = `regression`, type = enum, options: `regression`, `regression_l1`, `huber`, `fair`, `poisson`, `quantile`, `mape`, `gammma`, `tweedie`, `binary`, `multiclass`, `multiclassova`, `xentropy`, `xentlambda`, `lambdarank`, aliases: `objective_type`, `app`, `application`
  - regression application
    - `regression_l2`, L2 loss, aliases: `regression`, `mean_squared_error`, `mse`, `l2_root`, `root_mean_squared_error`, `rmse`
    - `regression_l1`, L1 loss, aliases: `mean_absolute_error`, `mae`
    - `huber`, Huber loss
    - `fair`, Fair loss
    - `poisson`, Poisson regression
    - `quantile`, Quantile regression
    - `mape`, MAPE loss, aliases: `mean_absolute_percentage_error`
    - `gamma`, Gamma regression with log-link. It might be useful, e.g., for modeling insurance claims severity, or for any target that might be gamma-distributed

https://lightgbm.readthedocs.io/en/latest/Parameters.html

Step 4 模型优化

调参策略：grid-search

```
# #1.{'n_estimators': 35}
param_test1 = {'n_estimators':list(range(35,46,2))}
gsearch1 = GridSearchCV(estimator = lgb.LGBMClassifier(
learning_rate=0.1,
subsample=0.8,
max_depth=13,
num_leaves=21
),param_grid =
param_test1,scoring='f1_socre',iid=False,cv=5)
gsearch1.fit(data_train,data_train_result)
print(gsearch1.grid_scores_)
print(gsearch1.best_params_)
print(gsearch1.best_score_)
```

Step 4 模型优化

调参结果

application=‘binary’,
objective=‘binary',
is_unbalance=True,
num_leaves=100,
colsample_bytree=0.8,
reg_alpha=0.001,
reg_lambda=0.06

Sweet Point

Something to BB

Something to BB

数据预处理十分重要!

Something to BB

Google

scikit learn

English

stackoverflow

Something to BB

Thanks

Designed by EX-VISION®