基于*数据科学家薪水情况*的机器学习与预测

武汉大学经济与管理学院 管理科学

指导老师：陈植元　**报告人**：陈实

目录

- 1.数据介绍与初步探索
- 2.基础机器学习
- 3.集成与朴素贝叶斯
- 4.总结

1.数据介绍与初步探索

数据介绍与初步探索

- **引言**：近年来随着大数据、人工智能的发展，数据科学类职业已然成为了最火热的岗位方向之一；作为管理科学专业出身的我，在kaggle上找到了一份截至2023年国外数据科学家薪水相关的数据，对此颇感兴趣。

- **数据介绍：Data Science Salaries 2023**（3755行 11列）

- work_year: The year the salary was paid.
- experience_level: The experience level in the job during the year
- employment_type: The type of employment for the role
- job_title: The role worked in during the year.
- salary: The total gross salary amount paid.
- salary_currency: The currency of the salary paid as an ISO 4217 currency code.
- salary_in_usd: The salary in USD
- employee_residence: Employee's primary country of residence.
- remote_ratio: The overall amount of work done remotely
- company_location: The country of the employer's main office or contracting branch
- company_size: The median number of people that worked for the company during the year

数据介绍与初步探索

➢ 初步探索

```
#在职工的住所地点中，住在美国的占了绝大部分
n_total = nrow(ds_salaries)
ds_salaries |>
  group_by(employee_residence) |>
  summarise(prop = round(n() / n_total,4)) |>
  arrange(desc(prop))
```

employee_resid... <chr>	prop <dbl>
US	0.8000
GB	0.0445
CA	0.0226
ES	0.0213
IN	0.0189
DE	0.0128
FR	0.0101

```
#使用USD作为薪水的记录占大多数；但salary_in_usd统一了薪水情况
n_total = nrow(ds_salaries)
ds_salaries |>
  group_by(salary_currency) |>
  summarise(prop = round(n() / n_total,4)) |>
  arrange(desc(prop))
```

salary_currency <chr>	prop <dbl>
USD	0.8586
EUR	0.0628
GBP	0.0429
INR	0.0160
CAD	0.0067
AUD	0.0024
BRL	0.0016
SGD	0.0016
PLN	0.0013
CHF	0.0011

数据介绍与初步探索

➢ 初步探索

```
#全日制工作占了99%以上，我们剔除其他数据；显然，该属性不能作为一个特征
n_total = nrow(ds_salaries)
ds_salaries |>
  group_by(employment_type) |>
  summarise(prop = round(n() / n_total,4)) |>
  arrange(desc(prop))
```

employment_ty... <chr>	prop <dbl>
FT	0.9901
PT	0.0045
CT	0.0027
FL	0.0027

```
#对原数据集进行修改
ds_salaries <- ds_salaries |>
  filter(employment_type == 'FT')
```

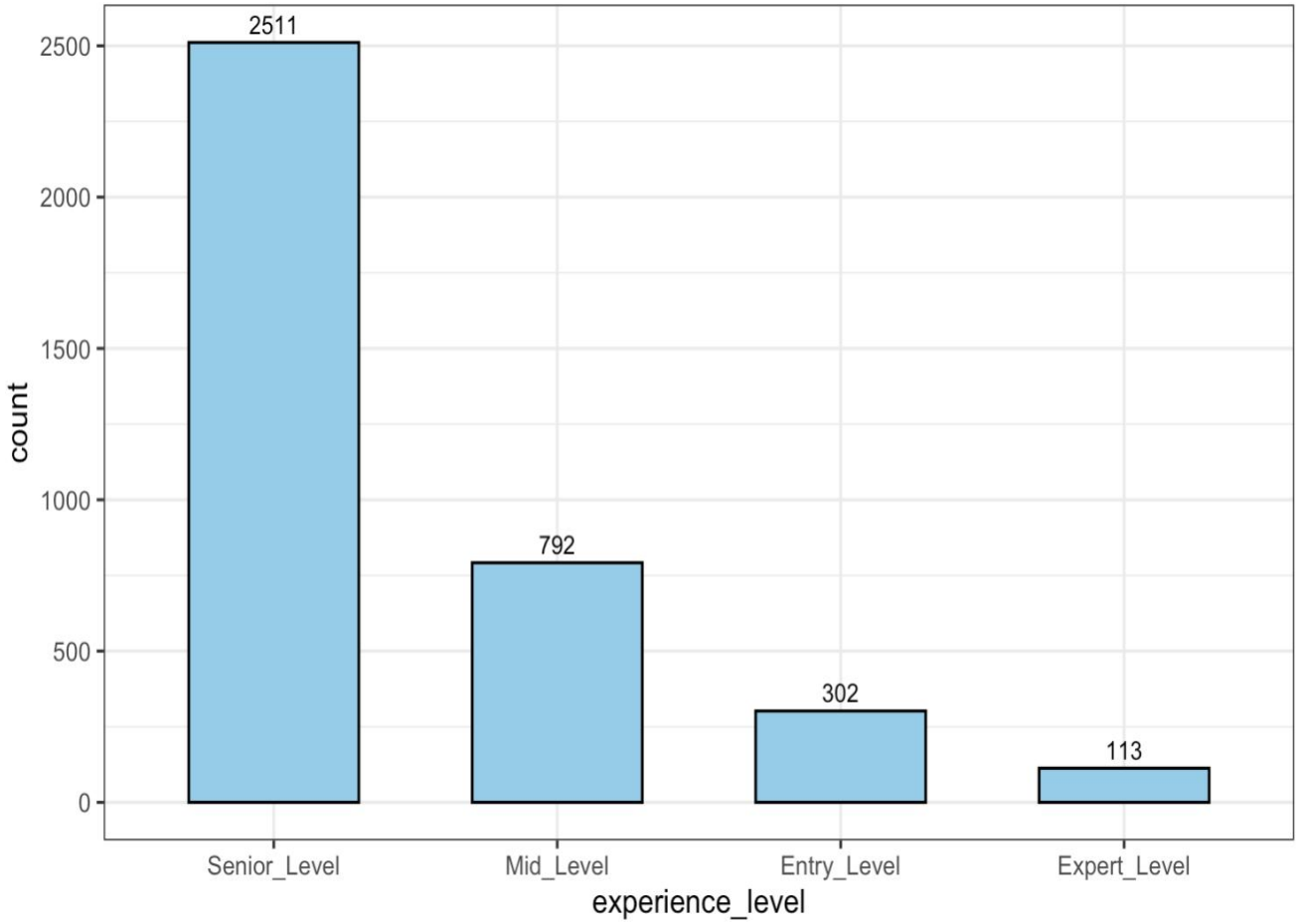
数据介绍与初步探索

➢ 可视化分析

按照经验水平分成四类：

- EN（Entry - level）表示初级水平
- MI（Mid - level）指中级水平
- SE（Senior - level）即高级水平
- EX（Expert - level）意为专家级

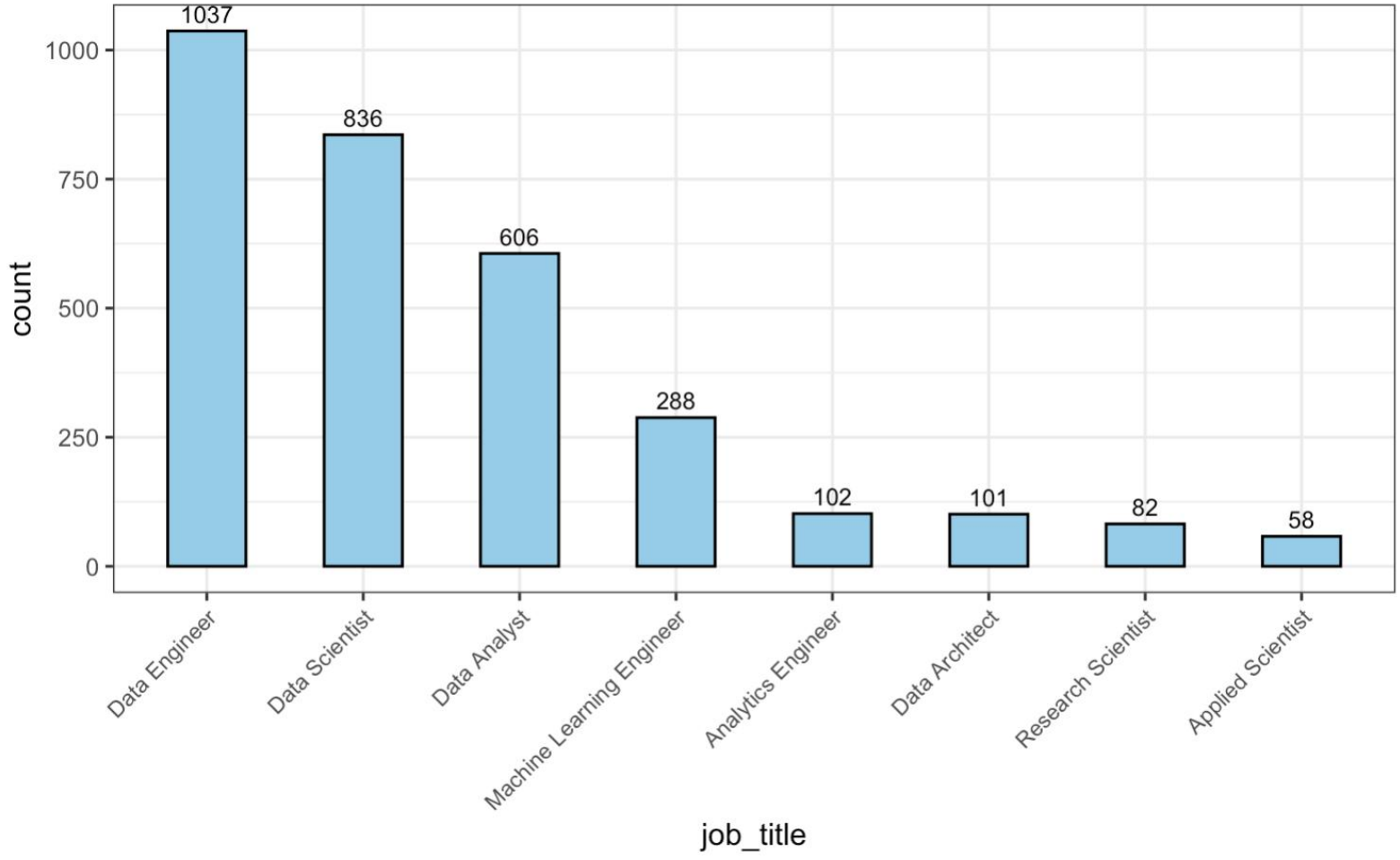
由图所示，处于高级水平的数据科学家最多



数据介绍与初步探索

➢ **可视化分析**

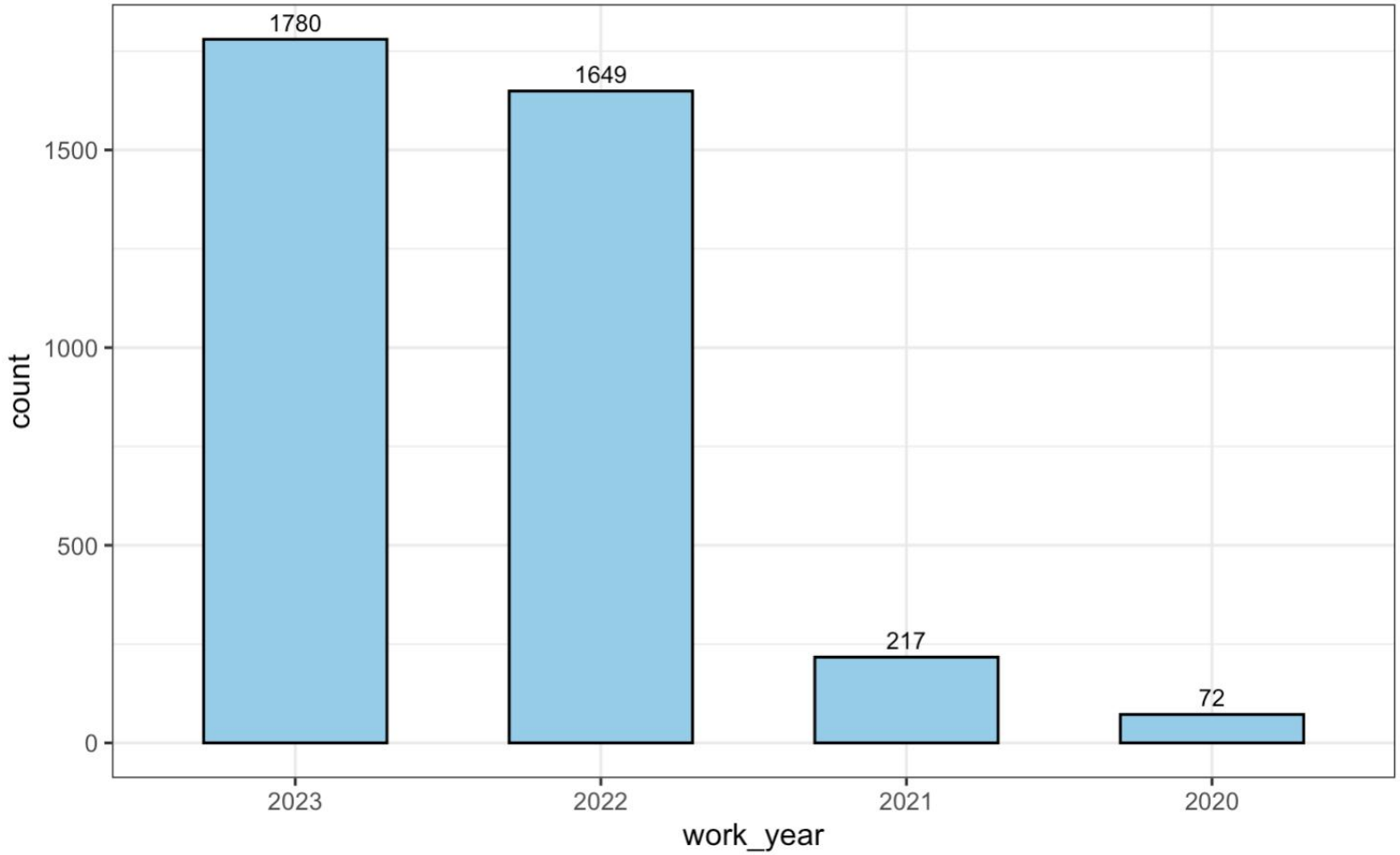
工作类型记录最多的是数据工程师



数据介绍与初步探索

➢ **可视化分析**

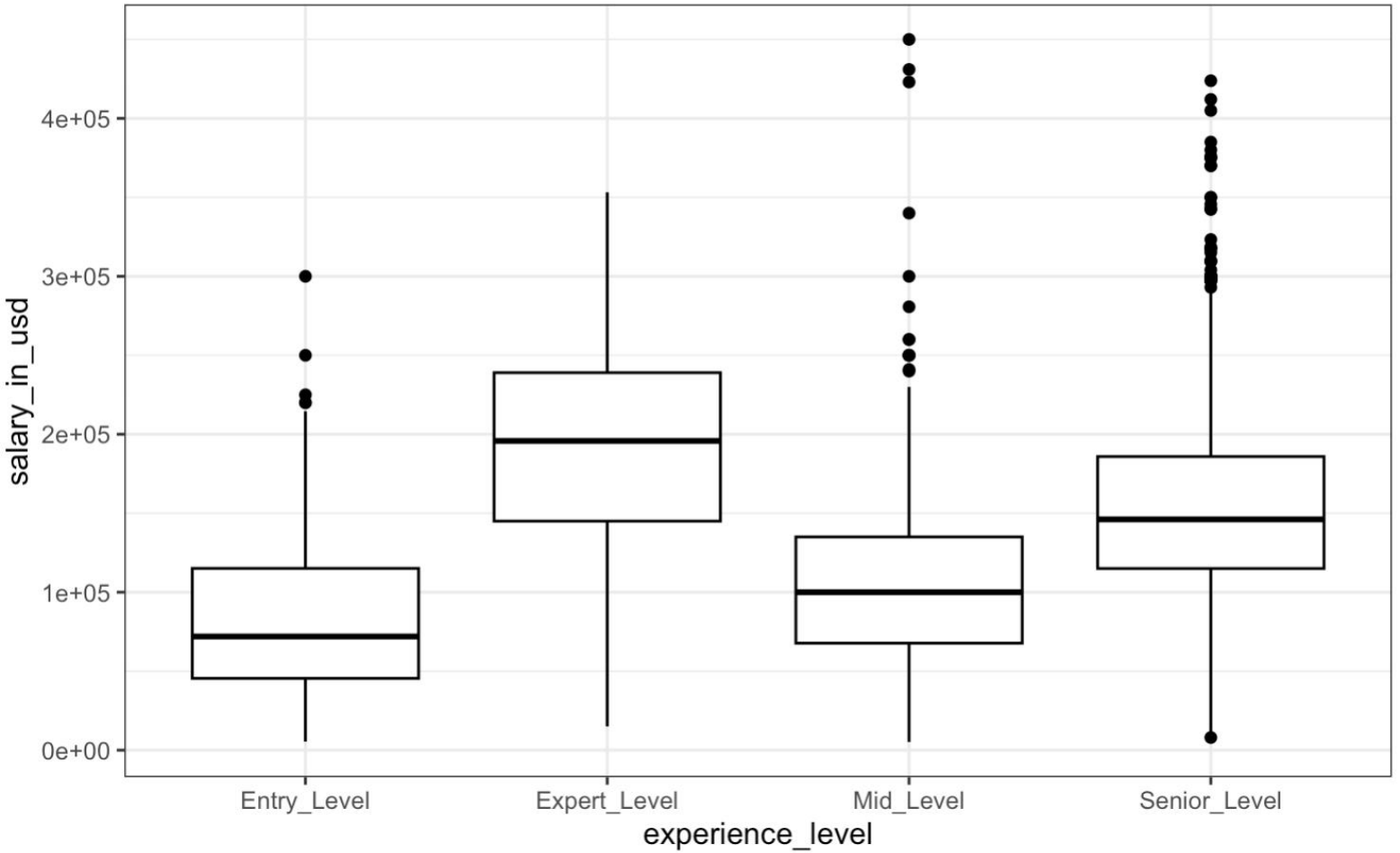
根据开始工作的年份分组汇总



数据介绍与初步探索

➢ **可视化分析**

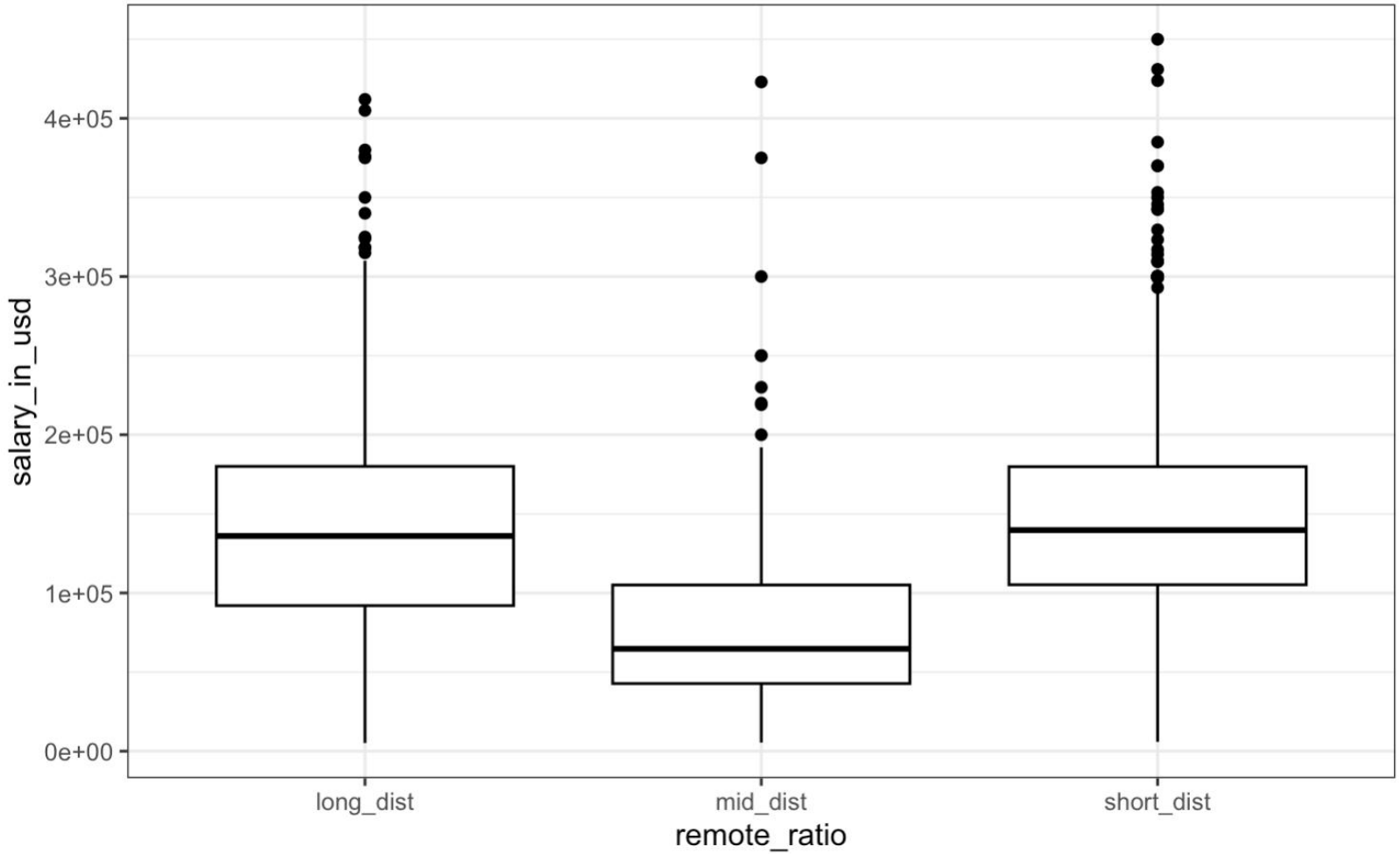
根据经验水平和薪资水平分组汇总:专家水平的平均薪资是最高的；相反地，入门水平最低



数据介绍与初步探索

➢ **可视化分析**

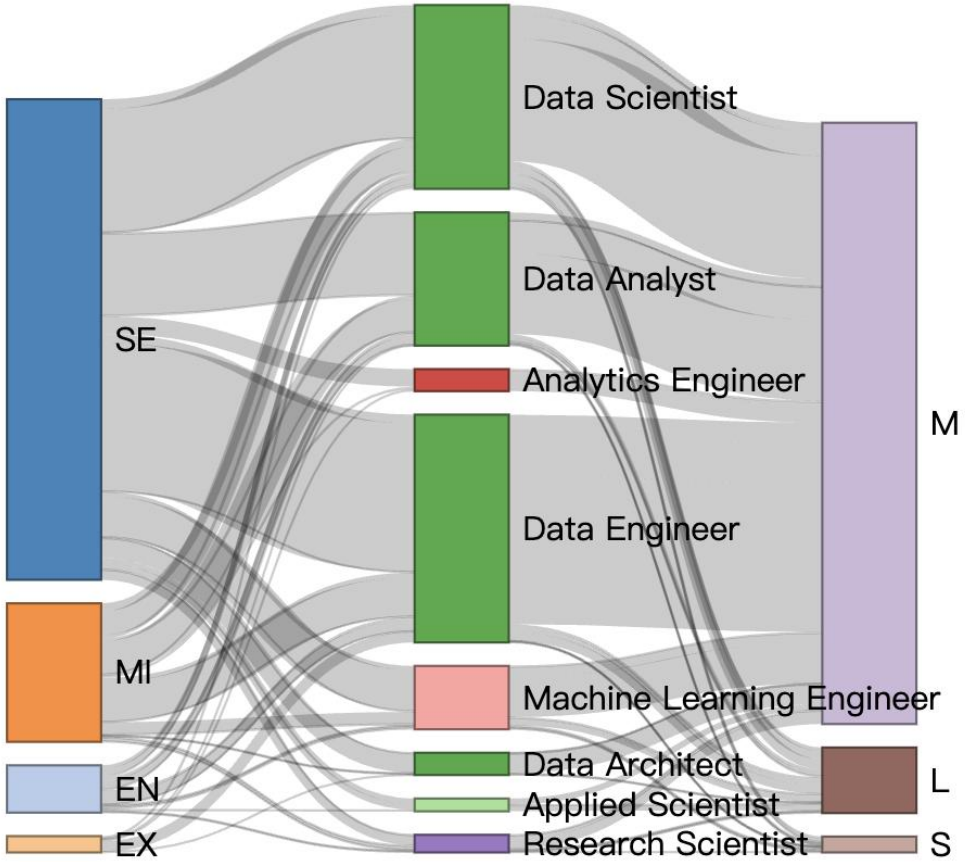
根据距离远近和薪资水平分组:中等距离的薪水最低，其他亮着并无显著区别



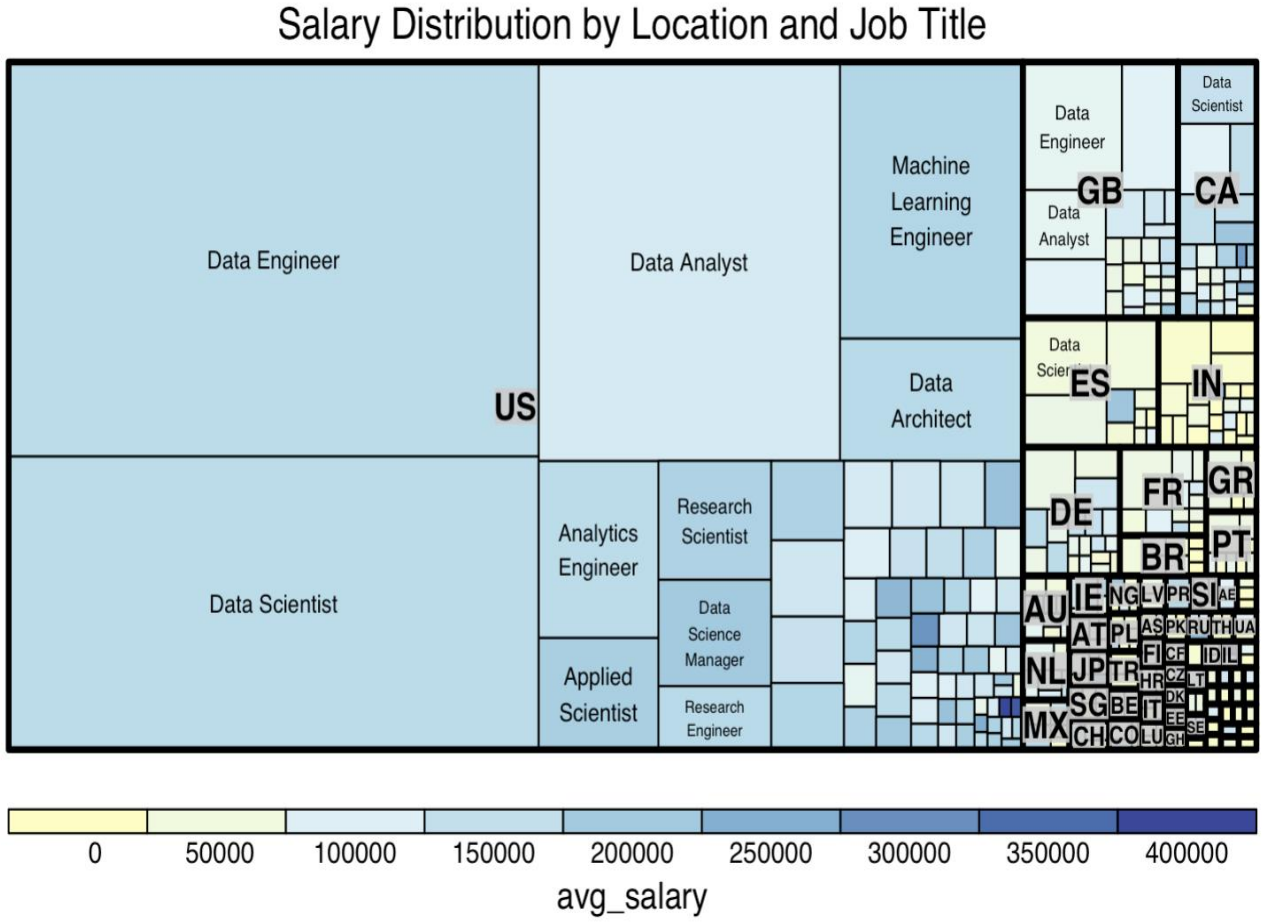
数据介绍与初步探索

- **高级可视化**

交互式桑基图



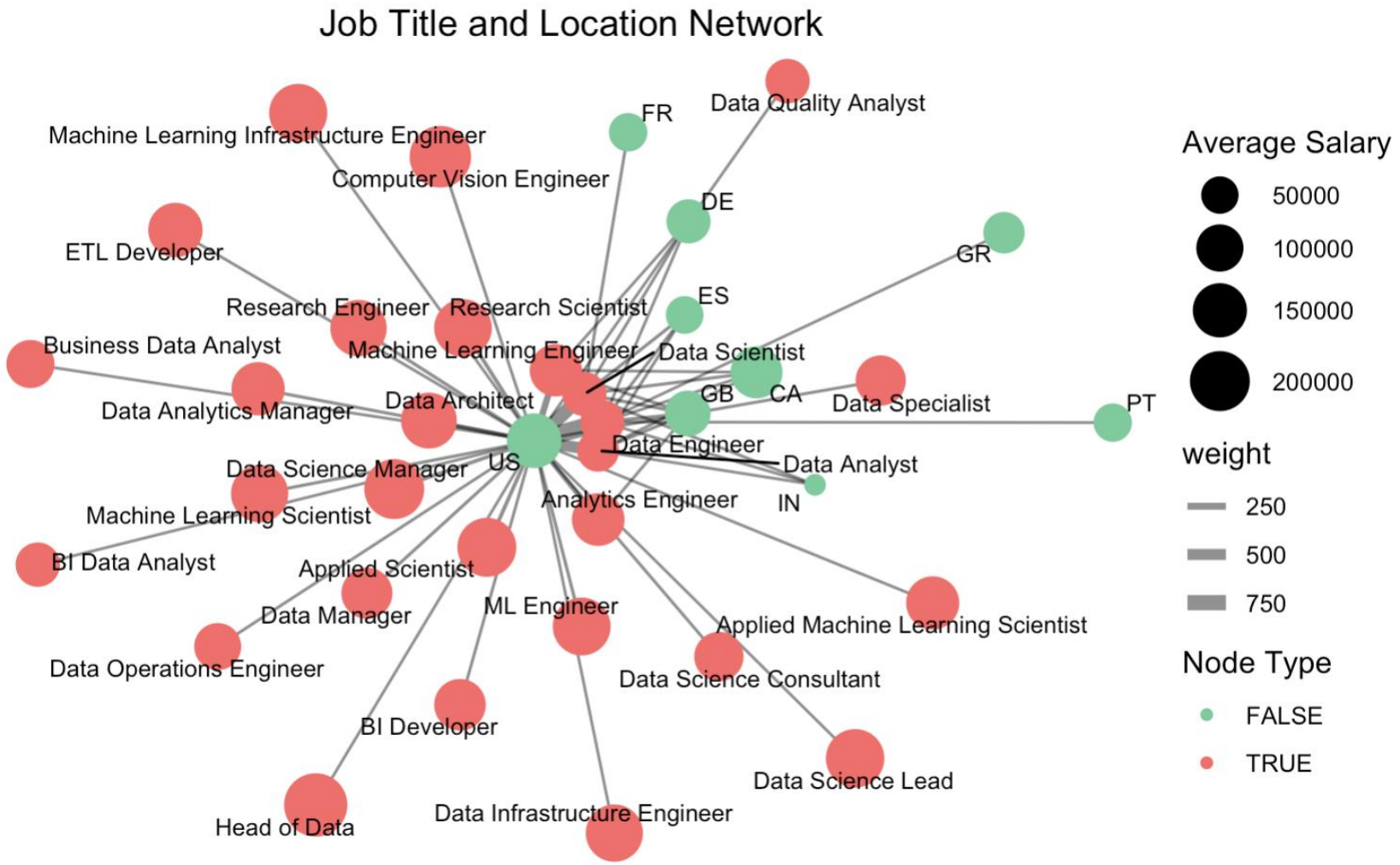
热力树图



数据介绍与初步探索

➢ **高级可视化**

职位网络图



2.基础机器学习

线性回归

salary_in_usd = 33422 - 2947remote_ratio + 39445exprience_level + 2834company_size

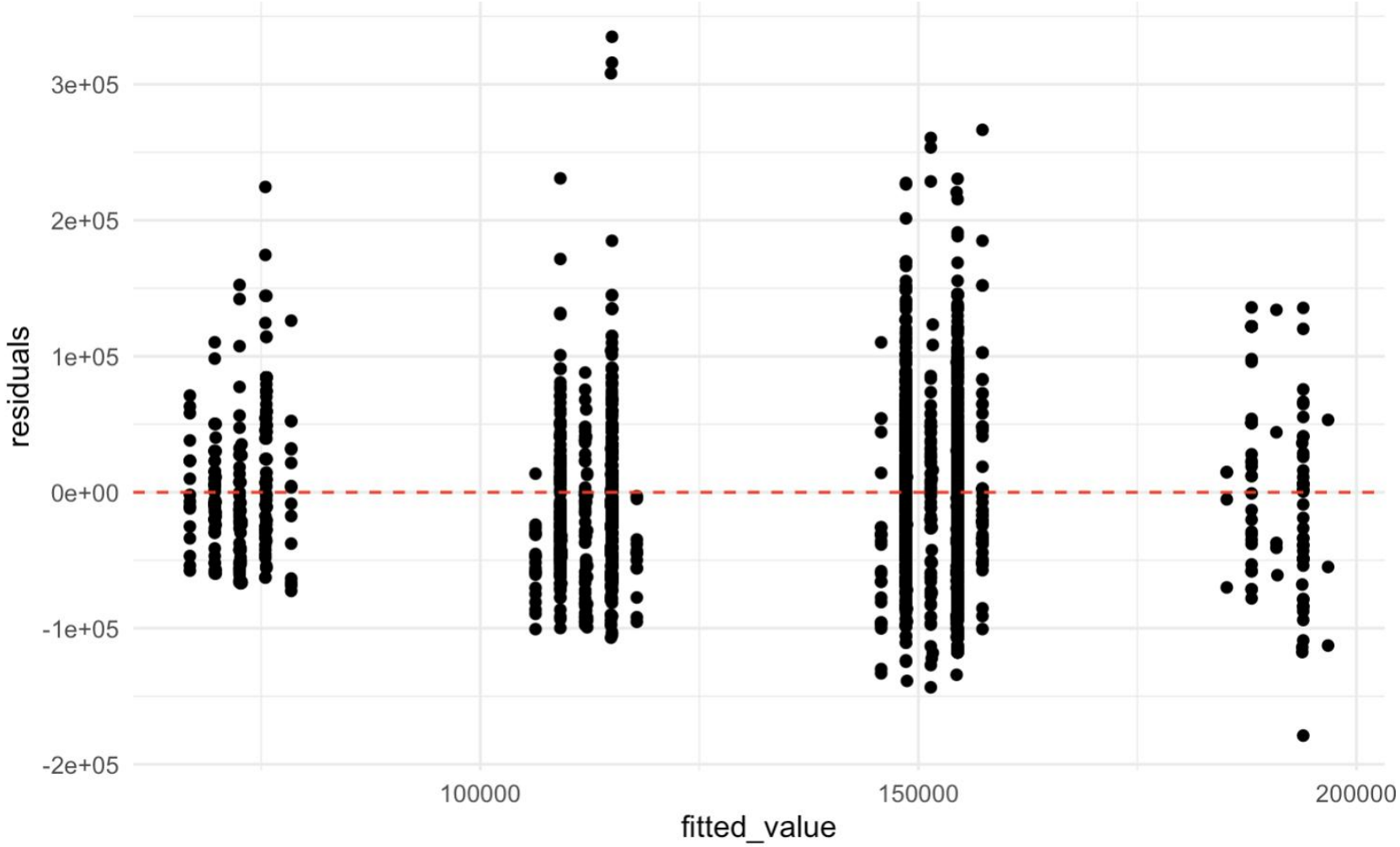
```
线性回归模型摘要:

Call:
lm(formula = salary_in_usd ~ remote_ratio + experience_level +
    company_size, data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max
-178922  -38010   -7058   34167  334968

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)         33422       7579   4.410 1.07e-05 ***
remote_ratio        -2947       1076  -2.737  0.00623 **
experience_level    39445       1556  25.344  < 2e-16 ***
company_size         2834       2704   1.048  0.29483
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 56920 on 2971 degrees of freedom
Multiple R-squared:  0.1811,    Adjusted R-squared:  0.1802
F-statistic:   219 on 3 and 2971 DF,  p-value: < 2.2e-16


线性回归模型的均方误差 (MSE): 2866421151
线性回归模型的均方根误差 (RMSE): 53538.97
线性回归模型的平均绝对误差 (MAE): 42458.26
```

可以显然的看到MSE非常大，这是由于数据集的特征比较少，而因变量薪水的波动范围较大且无法简单地归一化；

简单的线性回归无法很好地拟合模型

线性回归

➢ 残差可视化



残差图显示残差并非随机分布，而是呈现出一定的模式，说明模型没有很好地拟合数据。

此外，残差的散布随着拟合值的增加而增大，表现出异方差性，说明残差的方差不稳定。

逻辑回归

```
Call:
glm(formula = is_fully_remote ~ salary_in_usd_norm + experience_level +
    company_size, family = "binomial", data = train_data)

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)        0.25319    0.21594   1.173 0.240986
salary_in_usd_norm -0.09087    0.28707  -0.317 0.751589
experience_level2   0.02098    0.15477   0.136 0.892179
experience_level3   0.27223    0.14827   1.836 0.066356 .
experience_level4   0.23995    0.25615   0.937 0.348886
company_size2      -0.74379    0.20168  -3.688 0.000226 ***
company_size3      -0.46126    0.21895  -2.107 0.035146 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 4329.5  on 3160  degrees of freedom
Residual deviance: 4306.2  on 3154  degrees of freedom
AIC: 4320.2

Number of Fisher Scoring iterations: 4
```

采用把工作通勤距离remote_ratio转换成0-1变量is_fully_remote（距离为100则为1，50或0则为0）作为因变量；
由分类问题的性质，我们采用accuracy作为评价指标

训练集准确率：0.5659601
测试集准确率：0.5691203

逻辑回归

➢ LASSO 逻辑回归

```
最佳 lambda: 0.001434255
7 x 1 sparse Matrix of class "dgCMatrix"
                              s1
(Intercept)          0.15258168
salary_in_usd_norm -0.02732696
experience_level2    .
experience_level3    0.22451917
experience_level4    0.15876969
company_size2       -0.61910760
company_size3       -0.33051445
```

LASSO 训练集准确率: 0.5669092
LASSO 测试集准确率: 0.5691203
并无显著提高

精确率: 0.5085911
召回率: 0.10748
F1 分数: 0.177458

多项式逻辑回归

```
# weights:  16 (9 variable)
initial  value 4124.225724
iter  10 value 2344.737962
final  value 2339.402106
converged
多项式逻辑回归训练集准确率: 0.6910924
多项式逻辑回归测试集准确率: 0.6729475
```

多项式逻辑回归训练集准确率: 0.6910924
多项式逻辑回归测试集准确率: 0.6729475

相比于之前，有一个明显的提升!

神经网络

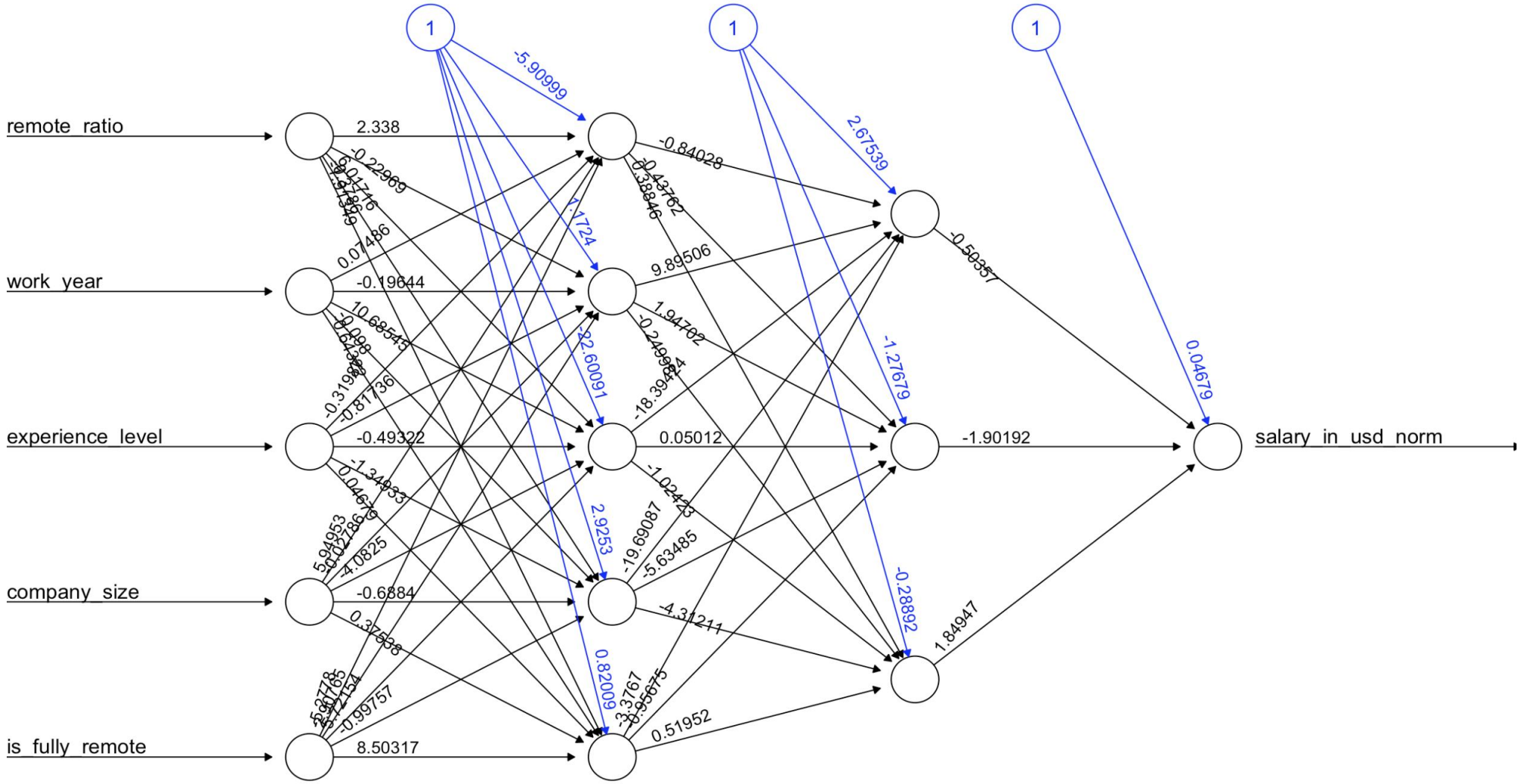
```
# 训练模型
nn_model <- neuralnet(
  salary_in_usd_norm ~ remote_ratio + work_year +
experience_level + company_size + is_fully_remote,
  data = train_data_nn,
  hidden = c(5, 3),
  linear.output = TRUE)
```

```
# 使用 compute 而非 prediction!
nn_pred_train <- compute(nn_model, train_data_nn[,
c("remote_ratio", "work_year", "experience_level",
"company_size", "is_fully_remote")])
predicted_train <- nn_pred_train$net.result
mse_train <- mean((predicted_train -
train_data_nn$salary_in_usd_norm)^2)
cat("神经网络训练集的 MSE:", round(mse_train, 5), "\n")

nn_pred <- compute(nn_model, test_data_nn[, c("remote_ratio",
"work_year", "experience_level", "company_size",
"is_fully_remote")])
predicted_values <- nn_pred$net.result
mse <- mean((predicted_values - test_data$salary_in_usd_norm)^2)
cat("神经网络测试集的 MSE:", round(mse, 5), "\n")
```

神经网络训练集的 MSE: 0.01457
神经网络测试集的 MSE: 0.01558

与之前相比，拟合程度大大提升

神经网络



Error: 21.802544 Steps: 55791

支持向量机

```r
#训练SVM
svm_model <- svm(
  salary_in_usd_norm ~ .,
  data = train_data_combined,
  type = "eps-regression",  # 用于回归的 SVR
  kernel = "radial",        # 使用 RBF 核
  cost = 1,                 # 正则化参数
  epsilon = 0.1             # 误差容忍度
```

```
Call:
svm(formula = salary_in_usd_norm ~ ., data = train_data_combined, type = "eps-
regression",
    kernel = "radial", cost = 1, epsilon = 0.1)


Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  radial
       cost:  1
      gamma:  0.1666667
    epsilon:  0.1


Number of Support Vectors:  166
```

支持向量机

```r
# 在测试集上预测
predictions_test <- predict(svm_model, test_inputs_scaled)
# 在训练集上预测
predictions_train <- predict(svm_model, train_inputs_scaled)
# 定义MSE函数
mse <- function(y_p, y) {
  return(mean((y - y_p)^2))
}
mse_test <- mse(predictions_test, test_outputs)
correlation_test <- cor(predictions_test, test_outputs)
mse_train <- mse(predictions_train, train_inputs)
# 输出结果
cat("训练集均方误差 (MSE):", mse_train, "\n")
cat("测试集均方误差 (MSE):", mse_test, "\n")
cat("测试集相关系数:", correlation_test, "\n")
```

训练集均方误差 (MSE): 0.0002332345
测试集均方误差 (MSE): 0.000517909
MSE又减小了许多!
测试集相关系数: 0.9882916

支持向量机

- **超参数调优**

```
tune_result <- tune(
  svm,
  salary_in_usd_norm ~ .,
  data = train_data_combined,
  ranges = list(
    cost = c(0.1, 1, 10),
    epsilon = c(0.01, 0.1, 0.5),
    gamma = c(0.01, 0.1, 1)
  ),
  type = "eps-regression",
  kernel = "radial"
)
```

cost <dbl>	epsilon <dbl>	gamma <dbl>
10	0.01	0.01

最佳参数:
调优后均方误差 (MSE): 2.210989e-06
调优后相关系数: 0.999954

经过超参数调优，进一步优化了MSE和相关系数

树形方法

➢ 决策树

```
# 训练决策树
tree_model <- rpart(
  salary_in_usd_norm ~ .,
  data = train_data_combined,
  method = "anova"  # 用于回归
)
```

训练集均方误差 (MSE): 0.0006690401
测试集均方误差 (MSE): 0.0007695463
决策树的MSE也是非常小的

决策树模型的伪 Accuracy（容差 = 0.05）：0.9757739
决策树模型的相关系数: 0.9799777

随机森林

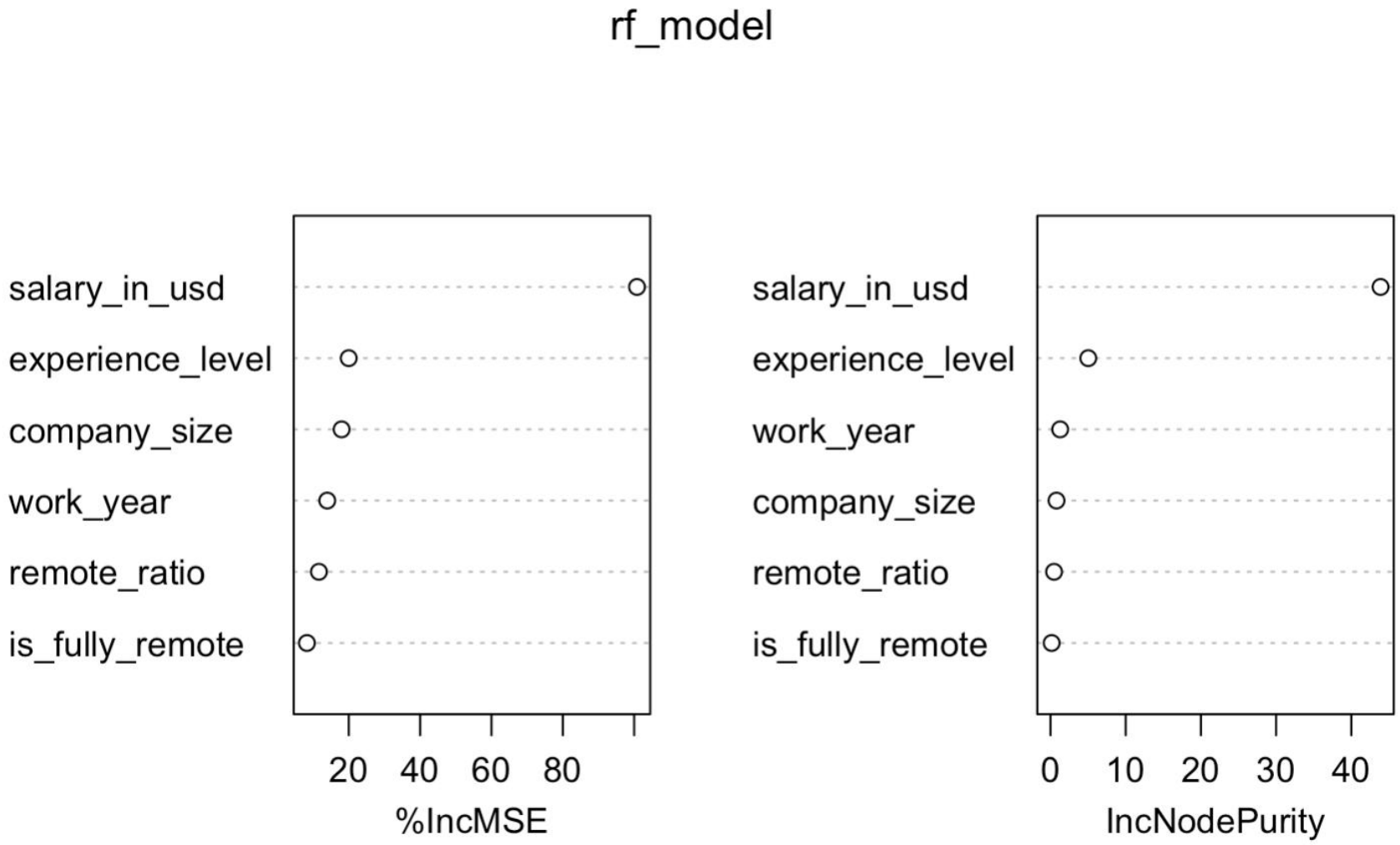
```
# 训练随机森林
rf_model <- randomForest(
  salary_in_usd_norm ~ .,
  data = train_data_combined,
  ntree = 500,
  mtry = 2,
  importance = TRUE
)
```

训练集均方误差 (MSE): 0.0006108438
测试集均方误差 (MSE): 0.001171349
相差不大

随机森林模型的伪 Accuracy（容差 = 0.05）：0.9434724
随机森林模型的相关系数: 0.9775013

树形方法

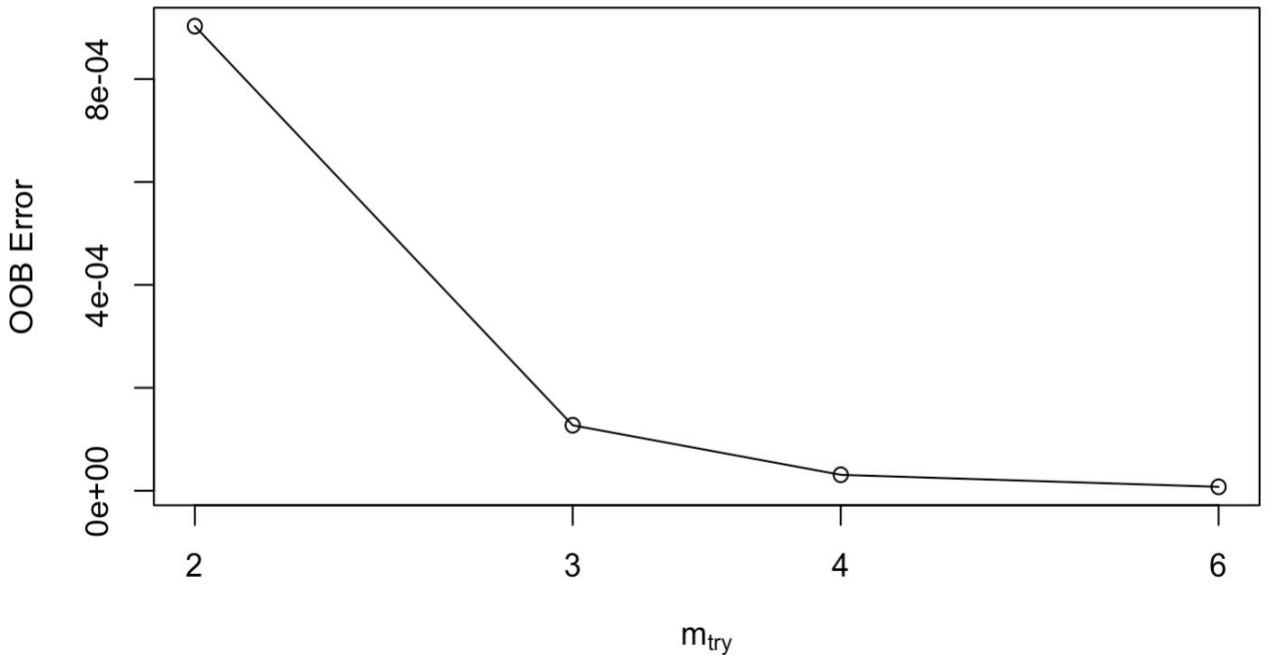
- 特征重要性可视化



树形方法

➢ 超参数调优

```
开始超参数调优...
mtry = 2  OOB error = 0.0009027784
Searching left ...
Searching right ...
mtry = 3        OOB error = 0.000127219
0.8590806 0.01
mtry = 4        OOB error = 3.093626e-05
0.7568268 0.01
mtry = 6        OOB error = 7.570332e-06
0.7552926 0.01
最佳 mtry 值: 6
使用调优后的 mtry 训练随机森林模型...
调优后随机森林模型性能:
均方误差 (MSE): 1.475129e-05
均方根误差 (RMSE): 0.00384074
平均绝对误差 (MAE): 0.0004988944
```



这张图展示了随机森林模型中超参数mtry（每次分裂随机选择的特征数量）与袋外误差（OOB Error）之间的关系

3.集成与朴素贝叶斯

Bagging

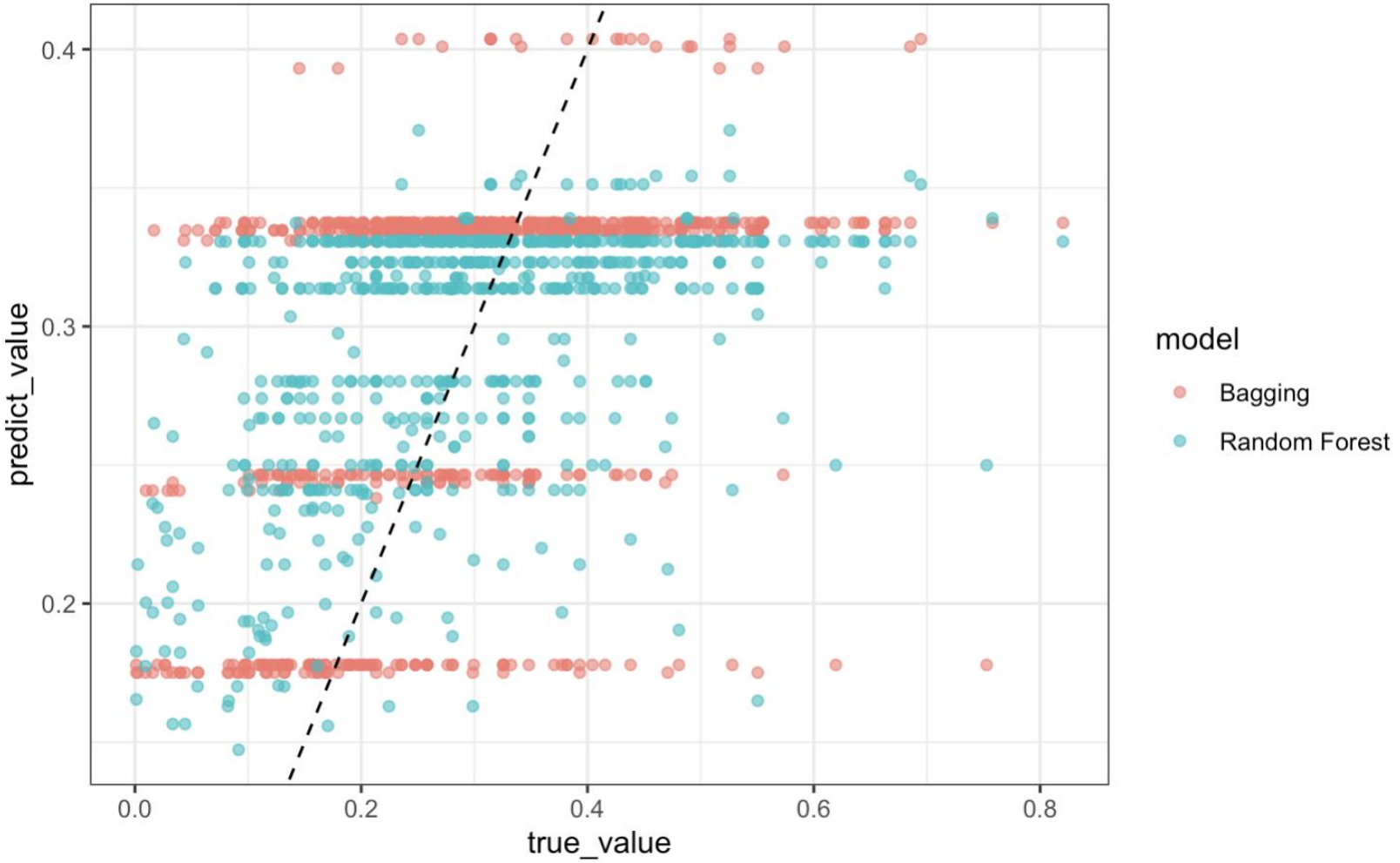
```
# 训练Bagging模型
bagging_model <- bagging(
  formula = salary_in_usd_norm ~ remote_ratio + work_year +
experience_level + company_size + is_fully_remote,
  data = train_data,
  coob = TRUE  # Out-of-bag error estimate
)
```

Bagging训练集 MSE: 0.01571
Bagging测试集 MSE: 0.01581

比之前的模型要稍微差一些

Bagging

➢ **Bagging与随机森林的比较**



从图中可以看出，
两种模型的预测点大部分集中在0.2到0.4之间，
且随机森林的点分布更靠近对角线，
表明其预测精度略高于Bagging模型。

朴素贝叶斯

```
# naiveBayes 训练模型
nb_model <- naiveBayes(
  salary_level ~ remote_ratio + work_year + experience_level +
company_size + is_fully_remote,
  data = train_data)
```

```
[1] "训练集混淆矩阵及评估："
Confusion Matrix and Statistics

          Reference
Prediction low median high
    low    801    271   43
    median 739    922  199
    high     0      0    0
```

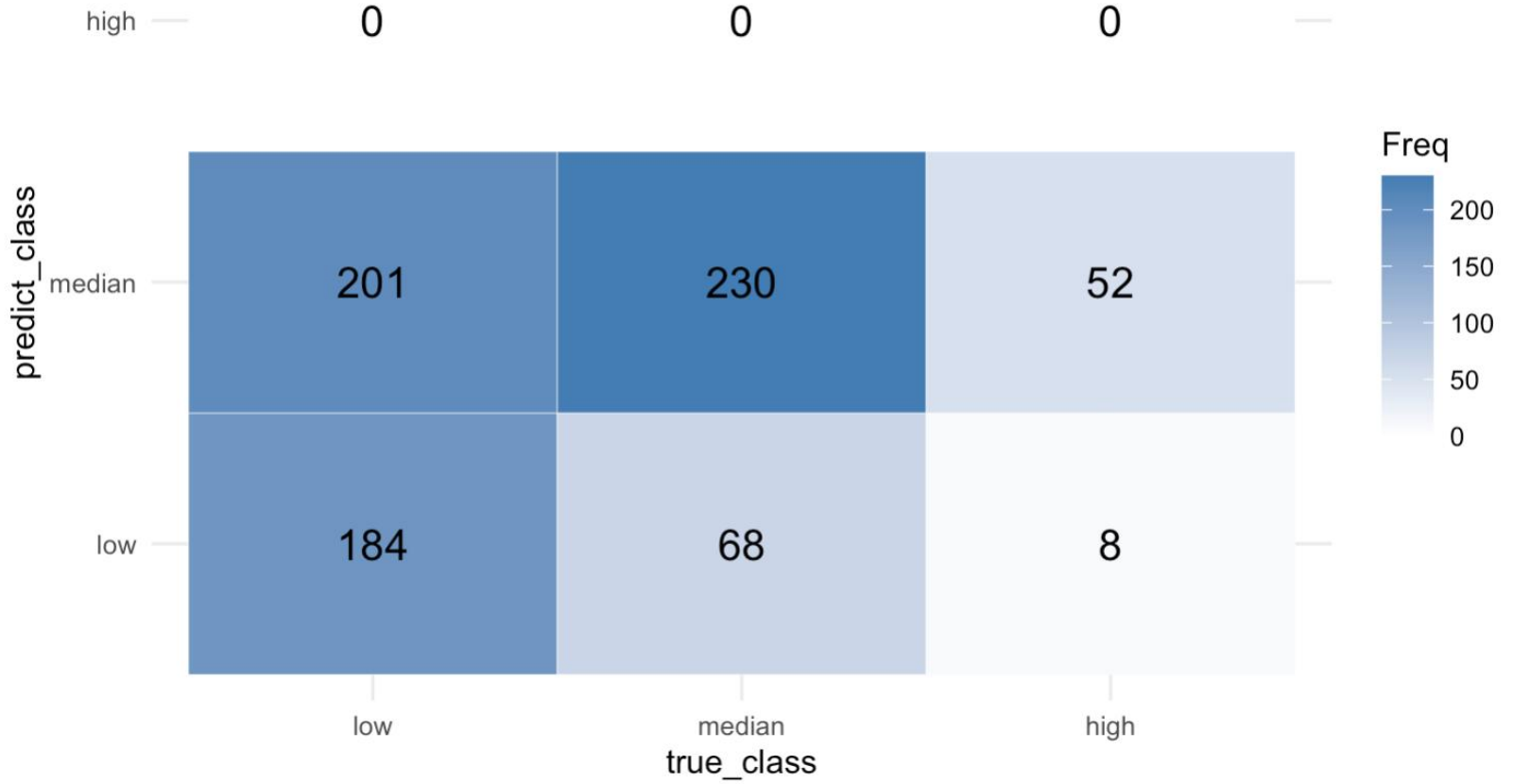
```
[1] "测试集混淆矩阵及评估："
Confusion Matrix and Statistics

          Reference
Prediction low median high
    low    202     56    8
    median 183    242   52
    high     0      0    0
```

训练集伪MSE（分类标签转数值后）：0.46118　　训练集正确率：05802
测试集伪MSE（分类标签转数值后）：0.44684　　测试集正确率：0.5935

朴素贝叶斯是一个分类模型，基于特征条件独立假设，适合处理离散类别；我将连续变量salary_in_used强制标签之后，MSE要比之前的模型大许多

朴素贝叶斯

- 混淆矩阵可视化



混淆矩阵显示朴素贝叶斯模型在测试集上对low（184/260）和median（230/483）类别的预测较准确，但对high类别完全失败（0/60），全被误分类为low或median，表明模型在识别high类别时表现很差。

4.总结

模型结果比较

回归模型 MSE

线性回归	2866421151
神经网络	训练集：0.01457 测试集：0.01558
支持向量机	训练集：0.0002332345 测试集：0.000517909
决策树	训练集：0.0006690401 测试集：0.0007695463
随机森林	训练集：0.0006108438 测试集：0.001171349
Bagging	训练集：0.01571 测试集：0.01581

抛开极端情况的线性回归不谈，支持向量机表现最好，Bagging和神经网络较差，决策树和随机森林可能存在过拟合

分类模型 Accuracy

逻辑回归	训练集：0.5659601 测试集：0.5691203
LASSO逻辑回归	训练集：0.5669092 测试集：0.5691203
多项式逻辑回归	训练集：0.6910924 测试集：0.6729475
决策树	0.9757739
随机森林	0.9434724
朴素贝叶斯	训练集：05802 测试集：0.5935

LASSO逐步回归表现最佳，逐步回归稍逊，多项式逐步回归测试集很好；决策树和随机森林为最高但可能过拟合，而朴素贝叶斯最差

研究总结

- 本研究通过对数据科学家薪水数据的深入分析，初步探索了数据特征，发现数据集中使用USD作为薪水的记录占比大，全日制工作占99%以上等特点。

- 在机器学习模型应用上，不同模型各有优劣，为后续研究数据科学家薪水影响因素及预测提供了参考。然而，研究也存在一定局限性，如部分模型拟合效果不佳、可能存在过拟合等问题。

- 未来研究可考虑增加数据集特征、优化模型参数选择方法，以及尝试更多复杂的集成学习算法，以提高模型的预测精度和泛化能力，更准确地揭示数据背后的规律。

武汉大学 经济与管理学院

Economics and Management School of Wuhan University

感谢聆听!

报告人：陈实

2025/6/13