Báo cáo Phân tích Thu nhập của Freelancer

Nhóm [Tên Nhóm Của Bạn]

2025-05-14

Table of Contents

# 1. Giới thiệu

Báo cáo này nhằm mục đích phân tích chi tiết tập dữ liệu về thu nhập của các freelancer. Chúng tôi sẽ khám phá các yếu tố như loại công việc, nền tảng, mức độ kinh nghiệm, khu vực khách hàng, và các chỉ số hiệu suất khác ảnh hưởng như thế nào đến thu nhập (Earnings\_USD) của freelancer. Thông qua phân tích dữ liệu khám phá (EDA) và xây dựng các mô hình dự đoán, chúng tôi hy vọng sẽ rút ra được những hiểu biết sâu sắc và có giá trị, giúp các freelancer tối ưu hóa chiến lược làm việc và thu nhập của mình. Dữ liệu được sử dụng là freelancer\_earnings\_bd.csv, chứa thông tin đa dạng về các freelancer và công việc của họ.

# 2. Tải và Chuẩn bị Dữ liệu

## 2.1. Tải Dữ liệu Thô

# Đặt đường dẫn làm việc (working directory) nếu cần, hoặc sử dụng RStudio Project  
# Ví dụ: setwd("D:/Path/To/Your/ProjectFolder")   
# Giả sử file CSV nằm trong cùng thư mục với file .Rmd hoặc trong thư mục con 'data'  
# Hãy đảm bảo đường dẫn file CSV là chính xác  
data\_raw <- read\_csv("data/freelancer\_earnings\_bd.csv")   
  
# Hiển thị vài dòng đầu của dữ liệu thô  
kable(head(data\_raw), caption = "Dữ liệu thô ban đầu (6 dòng đầu)")

Dữ liệu thô ban đầu (6 dòng đầu)

| Freelancer\_ID | Job\_Category | Platform | Experience\_Level | Client\_Region | Payment\_Method | Job\_Completed | Earnings\_USD | Hourly\_Rate | Job\_Success\_Rate | Client\_Rating | Job\_Duration\_Days | Project\_Type | Rehire\_Rate | Marketing\_Spend |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Web Development | Fiverr | Beginner | Asia | Mobile Banking | 180 | 1620 | 95.79 | 68.73 | 3.18 | 1 | Fixed | 40.19 | 53 |
| 2 | App Development | Fiverr | Beginner | Australia | Mobile Banking | 218 | 9078 | 86.38 | 97.54 | 3.44 | 54 | Fixed | 36.53 | 486 |
| 3 | Web Development | Fiverr | Beginner | UK | Crypto | 27 | 3455 | 85.17 | 86.60 | 4.20 | 46 | Hourly | 74.05 | 489 |
| 4 | Data Entry | PeoplePerHour | Intermediate | Asia | Bank Transfer | 17 | 5577 | 14.37 | 79.93 | 4.47 | 41 | Hourly | 27.58 | 67 |
| 5 | Digital Marketing | Upwork | Expert | Asia | Crypto | 245 | 5898 | 99.37 | 57.80 | 5.00 | 41 | Hourly | 69.09 | 489 |
| 6 | Customer Support | Toptal | Beginner | Europe | Crypto | 280 | 6867 | 43.04 | 57.80 | 4.87 | 8 | Fixed | 43.88 | 290 |

# Xem cấu trúc dữ liệu thô  
str(data\_raw, list.len=ncol(data\_raw))

## spc\_tbl\_ [1,950 × 15] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ Freelancer\_ID : num [1:1950] 1 2 3 4 5 6 7 8 9 10 ...  
## $ Job\_Category : chr [1:1950] "Web Development" "App Development" "Web Development" "Data Entry" ...  
## $ Platform : chr [1:1950] "Fiverr" "Fiverr" "Fiverr" "PeoplePerHour" ...  
## $ Experience\_Level : chr [1:1950] "Beginner" "Beginner" "Beginner" "Intermediate" ...  
## $ Client\_Region : chr [1:1950] "Asia" "Australia" "UK" "Asia" ...  
## $ Payment\_Method : chr [1:1950] "Mobile Banking" "Mobile Banking" "Crypto" "Bank Transfer" ...  
## $ Job\_Completed : num [1:1950] 180 218 27 17 245 280 96 112 233 156 ...  
## $ Earnings\_USD : num [1:1950] 1620 9078 3455 5577 5898 ...  
## $ Hourly\_Rate : num [1:1950] 95.8 86.4 85.2 14.4 99.4 ...  
## $ Job\_Success\_Rate : num [1:1950] 68.7 97.5 86.6 79.9 57.8 ...  
## $ Client\_Rating : num [1:1950] 3.18 3.44 4.2 4.47 5 4.87 4.29 3.84 4.27 4.57 ...  
## $ Job\_Duration\_Days: num [1:1950] 1 54 46 41 41 8 32 30 46 52 ...  
## $ Project\_Type : chr [1:1950] "Fixed" "Fixed" "Hourly" "Hourly" ...  
## $ Rehire\_Rate : num [1:1950] 40.2 36.5 74 27.6 69.1 ...  
## $ Marketing\_Spend : num [1:1950] 53 486 489 67 489 290 343 168 396 160 ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. Freelancer\_ID = col\_double(),  
## .. Job\_Category = col\_character(),  
## .. Platform = col\_character(),  
## .. Experience\_Level = col\_character(),  
## .. Client\_Region = col\_character(),  
## .. Payment\_Method = col\_character(),  
## .. Job\_Completed = col\_double(),  
## .. Earnings\_USD = col\_double(),  
## .. Hourly\_Rate = col\_double(),  
## .. Job\_Success\_Rate = col\_double(),  
## .. Client\_Rating = col\_double(),  
## .. Job\_Duration\_Days = col\_double(),  
## .. Project\_Type = col\_character(),  
## .. Rehire\_Rate = col\_double(),  
## .. Marketing\_Spend = col\_double()  
## .. )  
## - attr(\*, "problems")=<externalptr>

## 2.2. Làm sạch Dữ liệu

Dựa trên code backend đã có, chúng ta thực hiện các bước làm sạch sau:

* Loại bỏ các dòng có Earnings\_USD bị thiếu.
* Chuyển đổi kiểu dữ liệu cho các cột số.
* Chuẩn hóa tên các giá trị trong các cột dạng ký tự (Job\_Category, Experience\_Level, etc.) sang dạng Title Case.
* Đổi tên cột cho nhất quán và dễ sử dụng hơn.

data\_clean <- data\_raw %>%  
 filter(!is.na(Earnings\_USD)) %>% # Loại bỏ NA trong cột thu nhập chính  
 mutate(  
 # Đổi tên và chuyển đổi kiểu dữ liệu  
 FreelancerID = Freelancer\_ID, # Giữ lại ID gốc nếu cần  
 JobCategory = str\_to\_title(Job\_Category),  
 Platform = str\_to\_title(Platform),  
 ExperienceLevel = factor(str\_to\_title(Experience\_Level), levels = c("Beginner", "Intermediate", "Expert"), ordered = TRUE), # Chuyển thành factor có thứ tự  
 ClientRegion = str\_to\_title(Client\_Region),  
 PaymentMethod = str\_to\_title(Payment\_Method),  
 JobsCompleted = as.numeric(Job\_Completed),  
 EarningsUSD = as.numeric(Earnings\_USD), # Đổi tên từ Income để rõ ràng  
 HourlyRate = as.numeric(Hourly\_Rate),  
 JobSuccessRate = as.numeric(Job\_Success\_Rate),  
 ClientRating = as.numeric(Client\_Rating),  
 JobDurationDays = as.numeric(Job\_Duration\_Days),  
 ProjectType = factor(str\_to\_title(Project\_Type)), # Chuyển thành factor  
 RehireRate = as.numeric(Rehire\_Rate),  
 MarketingSpend = as.numeric(Marketing\_Spend)  
 ) %>%  
 # Chọn các cột đã được làm sạch và đổi tên  
 select(  
 FreelancerID, JobCategory, Platform, ExperienceLevel, ClientRegion, PaymentMethod,  
 JobsCompleted, EarningsUSD, HourlyRate, JobSuccessRate, ClientRating,  
 JobDurationDays, ProjectType, RehireRate, MarketingSpend  
 )  
  
# Kiểm tra lại các giá trị NA sau khi làm sạch cơ bản  
colSums(is.na(data\_clean)) %>% kable(col.names = c("Số lượng NA"), caption = "Kiểm tra NA sau làm sạch")

Kiểm tra NA sau làm sạch

|  | Số lượng NA |
| --- | --- |
| FreelancerID | 0 |
| JobCategory | 0 |
| Platform | 0 |
| ExperienceLevel | 0 |
| ClientRegion | 0 |
| PaymentMethod | 0 |
| JobsCompleted | 0 |
| EarningsUSD | 0 |
| HourlyRate | 0 |
| JobSuccessRate | 0 |
| ClientRating | 0 |
| JobDurationDays | 0 |
| ProjectType | 0 |
| RehireRate | 0 |
| MarketingSpend | 0 |

# Hiển thị vài dòng đầu của dữ liệu đã làm sạch  
kable(head(data\_clean), caption = "Dữ liệu đã làm sạch (6 dòng đầu)")

Dữ liệu đã làm sạch (6 dòng đầu)

| FreelancerID | JobCategory | Platform | ExperienceLevel | ClientRegion | PaymentMethod | JobsCompleted | EarningsUSD | HourlyRate | JobSuccessRate | ClientRating | JobDurationDays | ProjectType | RehireRate | MarketingSpend |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Web Development | Fiverr | Beginner | Asia | Mobile Banking | 180 | 1620 | 95.79 | 68.73 | 3.18 | 1 | Fixed | 40.19 | 53 |
| 2 | App Development | Fiverr | Beginner | Australia | Mobile Banking | 218 | 9078 | 86.38 | 97.54 | 3.44 | 54 | Fixed | 36.53 | 486 |
| 3 | Web Development | Fiverr | Beginner | Uk | Crypto | 27 | 3455 | 85.17 | 86.60 | 4.20 | 46 | Hourly | 74.05 | 489 |
| 4 | Data Entry | Peopleperhour | Intermediate | Asia | Bank Transfer | 17 | 5577 | 14.37 | 79.93 | 4.47 | 41 | Hourly | 27.58 | 67 |
| 5 | Digital Marketing | Upwork | Expert | Asia | Crypto | 245 | 5898 | 99.37 | 57.80 | 5.00 | 41 | Hourly | 69.09 | 489 |
| 6 | Customer Support | Toptal | Beginner | Europe | Crypto | 280 | 6867 | 43.04 | 57.80 | 4.87 | 8 | Fixed | 43.88 | 290 |

# Xem cấu trúc dữ liệu đã làm sạch  
str(data\_clean, list.len=ncol(data\_clean))

## tibble [1,950 × 15] (S3: tbl\_df/tbl/data.frame)  
## $ FreelancerID : num [1:1950] 1 2 3 4 5 6 7 8 9 10 ...  
## $ JobCategory : chr [1:1950] "Web Development" "App Development" "Web Development" "Data Entry" ...  
## $ Platform : chr [1:1950] "Fiverr" "Fiverr" "Fiverr" "Peopleperhour" ...  
## $ ExperienceLevel: Ord.factor w/ 3 levels "Beginner"<"Intermediate"<..: 1 1 1 2 3 1 1 1 2 1 ...  
## $ ClientRegion : chr [1:1950] "Asia" "Australia" "Uk" "Asia" ...  
## $ PaymentMethod : chr [1:1950] "Mobile Banking" "Mobile Banking" "Crypto" "Bank Transfer" ...  
## $ JobsCompleted : num [1:1950] 180 218 27 17 245 280 96 112 233 156 ...  
## $ EarningsUSD : num [1:1950] 1620 9078 3455 5577 5898 ...  
## $ HourlyRate : num [1:1950] 95.8 86.4 85.2 14.4 99.4 ...  
## $ JobSuccessRate : num [1:1950] 68.7 97.5 86.6 79.9 57.8 ...  
## $ ClientRating : num [1:1950] 3.18 3.44 4.2 4.47 5 4.87 4.29 3.84 4.27 4.57 ...  
## $ JobDurationDays: num [1:1950] 1 54 46 41 41 8 32 30 46 52 ...  
## $ ProjectType : Factor w/ 2 levels "Fixed","Hourly": 1 1 2 2 2 1 2 1 1 2 ...  
## $ RehireRate : num [1:1950] 40.2 36.5 74 27.6 69.1 ...  
## $ MarketingSpend : num [1:1950] 53 486 489 67 489 290 343 168 396 160 ...

**Nhận xét:** Sau bước làm sạch cơ bản, chúng ta không còn giá trị NA nào trong các cột đã chọn (hoặc số lượng NA rất ít nếu có trong các cột khác). Kiểu dữ liệu đã được chuẩn hóa, và các cột categorical quan trọng như ExperienceLevel và ProjectType đã được chuyển thành factor.

## 2.3. Chuẩn hóa Dữ liệu (Normalization)

Chuẩn hóa Min-Max đưa tất cả các giá trị về khoảng [0, 1]. Điều này có thể hữu ích cho một số thuật toán học máy nhạy cảm với thang đo của dữ liệu hoặc khi muốn so sánh các biến có đơn vị khác nhau trên cùng một biểu đồ.

data\_norm <- data\_clean %>%  
 mutate(  
 EarningsUSD\_norm = (EarningsUSD - min(EarningsUSD, na.rm = TRUE)) / (max(EarningsUSD, na.rm = TRUE) - min(EarningsUSD, na.rm = TRUE)),  
 JobSuccessRate\_norm = (JobSuccessRate - min(JobSuccessRate, na.rm = TRUE)) / (max(JobSuccessRate, na.rm = TRUE) - min(JobSuccessRate, na.rm = TRUE)),  
 ClientRating\_norm = (ClientRating - min(ClientRating, na.rm = TRUE)) / (max(ClientRating, na.rm = TRUE) - min(ClientRating, na.rm = TRUE)),  
 HourlyRate\_norm = (HourlyRate - min(HourlyRate, na.rm = TRUE)) / (max(HourlyRate, na.rm = TRUE) - min(HourlyRate, na.rm = TRUE)),  
 JobsCompleted\_norm = (JobsCompleted - min(JobsCompleted, na.rm = TRUE)) / (max(JobsCompleted, na.rm = TRUE) - min(JobsCompleted, na.rm = TRUE))  
 )  
  
kable(head(data\_norm %>% select(EarningsUSD, EarningsUSD\_norm, JobSuccessRate, JobSuccessRate\_norm)),   
 caption = "Ví dụ dữ liệu đã chuẩn hóa")

Ví dụ dữ liệu đã chuẩn hóa

| EarningsUSD | EarningsUSD\_norm | JobSuccessRate | JobSuccessRate\_norm |
| --- | --- | --- | --- |
| 1620 | 0.1578471 | 68.73 | 0.3726671 |
| 9078 | 0.9081489 | 97.54 | 0.9508328 |
| 3455 | 0.3424547 | 86.60 | 0.7312864 |
| 5577 | 0.5559356 | 79.93 | 0.5974313 |
| 5898 | 0.5882294 | 57.80 | 0.1533213 |
| 6867 | 0.6857143 | 57.80 | 0.1533213 |

# 3. Phân tích Dữ liệu Khám phá (EDA)

## 3.1. Thống kê Mô tả Tổng quan

skim(data\_clean)

Data summary

|  |  |
| --- | --- |
| Name | data\_clean |
| Number of rows | 1950 |
| Number of columns | 15 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 4 |
| factor | 2 |
| numeric | 9 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: character**

| skim\_variable | n\_missing | complete\_rate | min | max | empty | n\_unique | whitespace |
| --- | --- | --- | --- | --- | --- | --- | --- |
| JobCategory | 0 | 1 | 3 | 17 | 0 | 8 | 0 |
| Platform | 0 | 1 | 6 | 13 | 0 | 5 | 0 |
| ClientRegion | 0 | 1 | 2 | 11 | 0 | 7 | 0 |
| PaymentMethod | 0 | 1 | 6 | 14 | 0 | 4 | 0 |

**Variable type: factor**

| skim\_variable | n\_missing | complete\_rate | ordered | n\_unique | top\_counts |
| --- | --- | --- | --- | --- | --- |
| ExperienceLevel | 0 | 1 | TRUE | 3 | Beg: 668, Int: 641, Exp: 641 |
| ProjectType | 0 | 1 | FALSE | 2 | Fix: 997, Hou: 953 |

**Variable type: numeric**

| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| FreelancerID | 0 | 1 | 975.50 | 563.06 | 1.00 | 488.25 | 975.50 | 1462.75 | 1950.00 | ▇▇▇▇▇ |
| JobsCompleted | 0 | 1 | 150.86 | 85.48 | 5.00 | 76.00 | 149.00 | 225.00 | 299.00 | ▇▇▇▇▇ |
| EarningsUSD | 0 | 1 | 5017.57 | 2926.28 | 51.00 | 2419.00 | 5048.00 | 7608.25 | 9991.00 | ▇▇▇▇▇ |
| HourlyRate | 0 | 1 | 52.58 | 26.93 | 5.02 | 30.05 | 52.28 | 75.12 | 99.83 | ▇▇▇▇▇ |
| JobSuccessRate | 0 | 1 | 74.95 | 14.62 | 50.16 | 61.92 | 75.40 | 87.54 | 99.99 | ▇▇▇▇▇ |
| ClientRating | 0 | 1 | 4.00 | 0.58 | 3.00 | 3.51 | 3.99 | 4.50 | 5.00 | ▇▇▇▇▇ |
| JobDurationDays | 0 | 1 | 44.70 | 26.02 | 1.00 | 22.00 | 45.00 | 67.00 | 89.00 | ▇▇▇▇▇ |
| RehireRate | 0 | 1 | 44.56 | 20.19 | 10.00 | 27.15 | 43.92 | 61.69 | 79.95 | ▇▇▇▇▇ |
| MarketingSpend | 0 | 1 | 248.52 | 148.08 | 0.00 | 119.00 | 252.50 | 379.00 | 499.00 | ▇▇▇▇▇ |

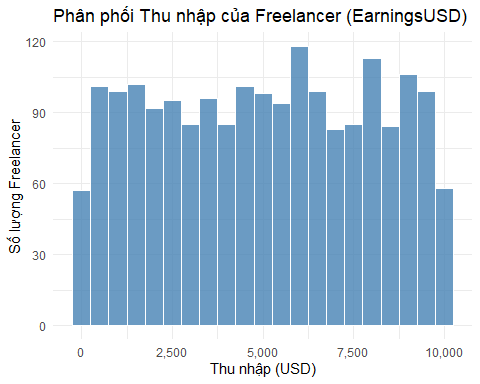
**Phân tích:**

* Tập dữ liệu có r nrow(data\_clean) quan sát và r ncol(data\_clean) biến.
* **EarningsUSD**: Thu nhập trung bình là r scales::dollar(round(mean(data\_clean$EarningsUSD),0)), với độ lệch chuẩn khá lớn, cho thấy sự biến động cao trong thu nhập. Giá trị trung vị là r scales::dollar(median(data\_clean$EarningsUSD)). Khoảng thu nhập rộng từ r scales::dollar(min(data\_clean$EarningsUSD)) đến r scales::dollar(max(data\_clean$EarningsUSD)).
* **HourlyRate**: Mức lương theo giờ trung bình là r scales::dollar(round(mean(data\_clean$HourlyRate),2)).
* **JobSuccessRate**: Tỷ lệ thành công trung bình là r round(mean(data\_clean$JobSuccessRate),2)%.
* **ClientRating**: Đánh giá khách hàng trung bình là r round(mean(data\_clean$ClientRating),2).
* Các biến categorical như JobCategory, Platform, ExperienceLevel có số lượng các mức khác nhau, sẽ được khám phá chi tiết hơn bằng biểu đồ.

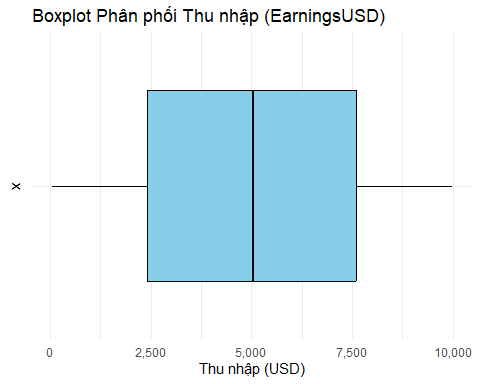
## 3.2. Phân tích Đơn biến (Univariate Analysis)

### 3.2.1. Phân phối Thu nhập (EarningsUSD)

ggplot(data\_clean, aes(x = EarningsUSD)) +  
 geom\_histogram(binwidth = 500, fill = "steelblue", color = "white", alpha = 0.8) +  
 labs(title = "Phân phối Thu nhập của Freelancer (EarningsUSD)",  
 x = "Thu nhập (USD)",  
 y = "Số lượng Freelancer") +  
 theme\_minimal() +  
 scale\_x\_continuous(labels = scales::comma)



ggplot(data\_clean, aes(x = "", y = EarningsUSD)) +  
 geom\_boxplot(fill = "skyblue", color = "black", outlier.colour = "red", outlier.shape = 1) +  
 labs(title = "Boxplot Phân phối Thu nhập (EarningsUSD)",  
 y = "Thu nhập (USD)") +  
 theme\_minimal() +   
 coord\_flip() + # Xoay boxplot cho dễ nhìn hơn khi chỉ có 1 nhóm  
 scale\_y\_continuous(labels = scales::comma)

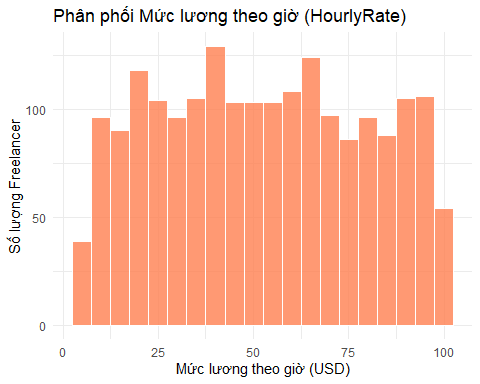


**Ý nghĩa và Phân tích:**

* **Histogram**: Biểu đồ histogram cho thấy phân phối của EarningsUSD bị lệch phải (right-skewed), nghĩa là phần lớn freelancer có thu nhập ở mức thấp đến trung bình, và một số ít freelancer có thu nhập rất cao.
* **Boxplot**: Boxplot cũng xác nhận điều này, với nhiều điểm ngoại lệ (outliers) ở phía thu nhập cao. Khoảng tứ phân vị (IQR) cho thấy sự tập trung của 50% dữ liệu ở giữa.
* **Tri thức mới**: Sự chênh lệch lớn về thu nhập cho thấy có thể có các nhóm freelancer khác nhau hoặc các yếu tố đặc biệt dẫn đến thu nhập cao vượt trội.

### 3.2.2. Phân phối Tỷ lệ Giờ làm việc (HourlyRate)

ggplot(data\_clean, aes(x = HourlyRate)) +  
 geom\_histogram(binwidth = 5, fill = "coral", color = "white", alpha = 0.8) +  
 labs(title = "Phân phối Mức lương theo giờ (HourlyRate)",  
 x = "Mức lương theo giờ (USD)",  
 y = "Số lượng Freelancer") +  
 theme\_minimal() +  
 scale\_x\_continuous(labels = scales::comma)

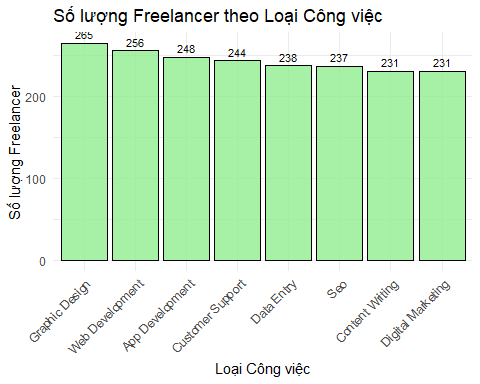


**Ý nghĩa và Phân tích:**

* Phân phối của HourlyRate cũng có xu hướng lệch phải, nhưng không rõ rệt bằng EarningsUSD. Phần lớn freelancer có mức lương theo giờ tập trung ở khoảng thấp và trung bình. Có một số freelancer đạt mức lương giờ rất cao.

### 3.2.3. Phân loại Công việc (JobCategory)

data\_clean %>%  
 count(JobCategory, sort = TRUE) %>%  
 ggplot(aes(x = reorder(JobCategory, -n), y = n)) +  
 geom\_bar(stat = "identity", fill = "lightgreen", color = "black", alpha = 0.8) +  
 geom\_text(aes(label = n), vjust = -0.5, size = 3) +  
 labs(title = "Số lượng Freelancer theo Loại Công việc",  
 x = "Loại Công việc",  
 y = "Số lượng Freelancer") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))

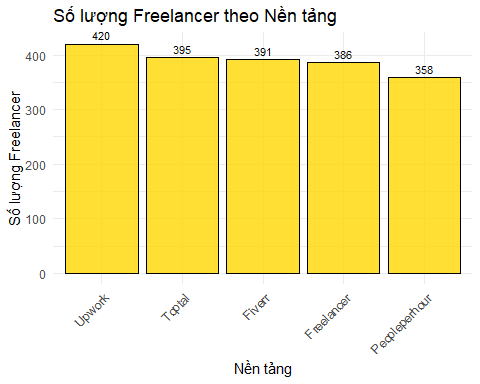


**Ý nghĩa và Phân tích:**

* Biểu đồ cột cho thấy các loại công việc phổ biến nhất. “Data Entry”, “Web Development”, “Graphic Design”, và “Customer Support” là những lĩnh vực có nhiều freelancer tham gia nhất trong tập dữ liệu này.
* **Tri thức mới**: Điều này cho thấy nhu cầu hoặc sự phổ biến của các loại công việc này trên các nền tảng freelancer.

### 3.2.4. Nền tảng Làm việc (Platform)

data\_clean %>%  
 count(Platform, sort = TRUE) %>%  
 ggplot(aes(x = reorder(Platform, -n), y = n)) +  
 geom\_bar(stat = "identity", fill = "gold", color = "black", alpha = 0.8) +  
 geom\_text(aes(label = n), vjust = -0.5, size = 3) +  
 labs(title = "Số lượng Freelancer theo Nền tảng",  
 x = "Nền tảng",  
 y = "Số lượng Freelancer") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))

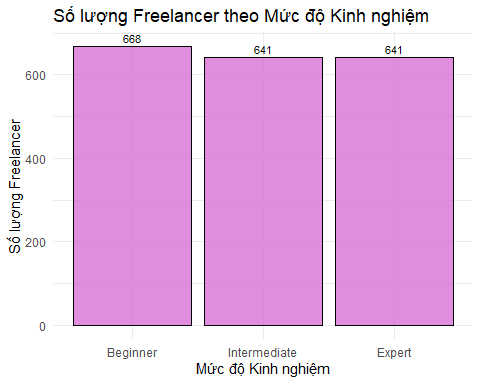


**Ý nghĩa và Phân tích:**

* “Fiverr”, “Upwork”, và “Toptal” là các nền tảng có nhiều freelancer nhất trong mẫu dữ liệu này.
* **Tri thức mới**: Sự phân bổ này có thể phản ánh mức độ phổ biến của các nền tảng hoặc đặc điểm của tập dữ liệu được thu thập.

### 3.2.5. Mức độ Kinh nghiệm (ExperienceLevel)

data\_clean %>%  
 count(ExperienceLevel, sort = FALSE) %>% # Giữ nguyên thứ tự của factor đã định nghĩa  
 ggplot(aes(x = ExperienceLevel, y = n)) +   
 geom\_bar(stat = "identity", fill = "orchid", color = "black", alpha = 0.8) +  
 geom\_text(aes(label = n), vjust = -0.5, size = 3) +  
 labs(title = "Số lượng Freelancer theo Mức độ Kinh nghiệm",  
 x = "Mức độ Kinh nghiệm",  
 y = "Số lượng Freelancer") +  
 theme\_minimal()

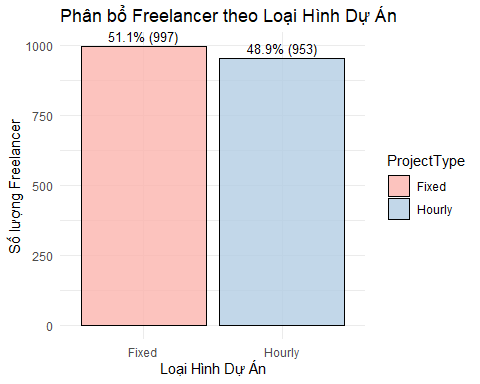


**Ý nghĩa và Phân tích:**

* Số lượng freelancer ở các mức kinh nghiệm “Beginner”, “Intermediate”, và “Expert” có sự phân bổ khá đều, với “Beginner” và “Intermediate” chiếm tỷ lệ cao hơn một chút.
* **Tri thức mới**: Điều này cho thấy thị trường freelancer có sự tham gia của nhiều người ở các giai đoạn sự nghiệp khác nhau.

### 3.2.6. Loại Hình Dự Án (ProjectType)

project\_type\_summary <- data\_clean %>%  
 group\_by(ProjectType) %>%  
 summarise(count = n(),  
 percentage = n() / nrow(data\_clean) \* 100) %>%  
 ungroup()  
  
ggplot(project\_type\_summary, aes(x = ProjectType, y = count, fill = ProjectType)) +  
 geom\_bar(stat = "identity", color = "black", alpha = 0.8) +  
 geom\_text(aes(label = paste0(round(percentage,1), "% (", count, ")")), vjust = -0.5, size = 3.5) +  
 labs(title = "Phân bổ Freelancer theo Loại Hình Dự Án",  
 x = "Loại Hình Dự Án",  
 y = "Số lượng Freelancer") +  
 theme\_minimal() +  
 scale\_fill\_brewer(palette = "Pastel1")



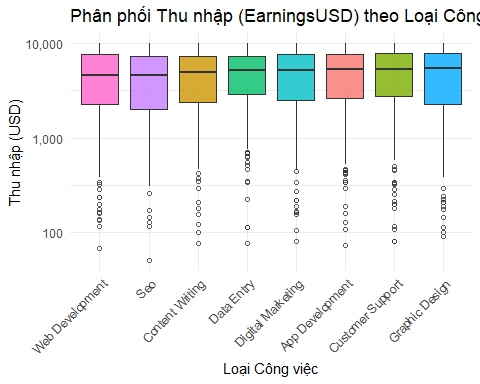
**Ý nghĩa và Phân tích:**

* Biểu đồ cho thấy sự phân bổ giữa các dự án tính theo giờ (Hourly) và dự án giá cố định (Fixed). Trong tập dữ liệu này, số lượng dự án tính theo giờ và giá cố định khá tương đồng, với dự án theo giờ chiếm tỷ lệ nhỉnh hơn một chút.
* **Tri thức mới**: Cả hai mô hình định giá đều phổ biến, freelancer có thể lựa chọn tùy theo tính chất công việc và sở thích.

## 3.3. Phân tích Đa biến (Bivariate & Multivariate Analysis)

### 3.3.1. Thu nhập theo Loại Công việc

ggplot(data\_clean, aes(x = reorder(JobCategory, EarningsUSD, FUN = median, .desc = TRUE), y = EarningsUSD, fill = JobCategory)) +  
 geom\_boxplot(alpha = 0.8, outlier.shape = 1) +  
 labs(title = "Phân phối Thu nhập (EarningsUSD) theo Loại Công việc",  
 x = "Loại Công việc",  
 y = "Thu nhập (USD)") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1), legend.position = "none") +  
 scale\_y\_log10(labels = scales::comma) # Sử dụng thang log cho y để dễ nhìn hơn do độ lệch lớn

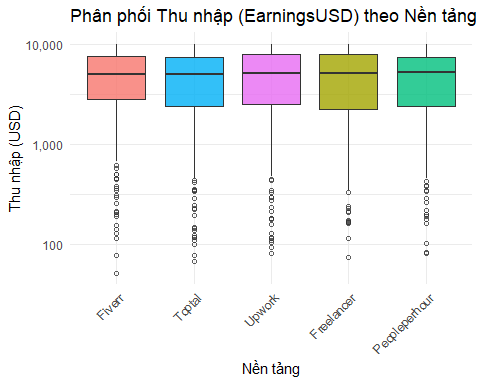


**Ý nghĩa và Phân tích:**

* Biểu đồ boxplot cho thấy sự khác biệt đáng kể về thu nhập trung vị giữa các loại công việc.
* “App Development” và “Web Development” dường như có mức thu nhập trung vị cao hơn so với các ngành khác như “Data Entry” hay “Customer Support”.
* Sự biến động (độ dài của box và whiskers) cũng khác nhau giữa các ngành.
* **Tri thức mới**: Lựa chọn ngành nghề có ảnh hưởng lớn đến tiềm năng thu nhập. Các ngành kỹ thuật cao thường có thu nhập tốt hơn.

### 3.3.2. Thu nhập theo Nền tảng

ggplot(data\_clean, aes(x = reorder(Platform, EarningsUSD, FUN = median, .desc = TRUE), y = EarningsUSD, fill = Platform)) +  
 geom\_boxplot(alpha = 0.8, outlier.shape = 1) +  
 labs(title = "Phân phối Thu nhập (EarningsUSD) theo Nền tảng",  
 x = "Nền tảng",  
 y = "Thu nhập (USD)") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1), legend.position = "none") +  
 scale\_y\_log10(labels = scales::comma)

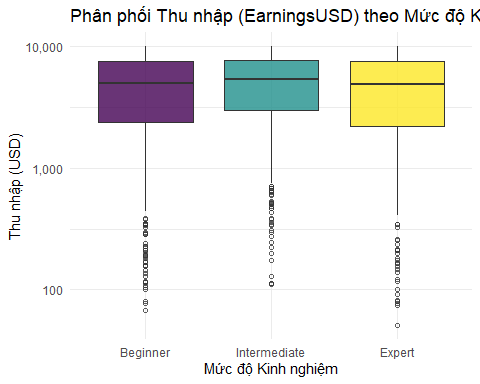


**Ý nghĩa và Phân tích:**

* Có sự khác biệt về thu nhập trung vị giữa các nền tảng. Ví dụ, “Toptal” thường được biết đến với các freelancer chất lượng cao và dự án giá trị lớn, có thể dẫn đến thu nhập trung bình cao hơn.
* **Tri thức mới**: Nền tảng làm việc cũng là một yếu tố quan trọng. Một số nền tảng có thể tập trung vào các phân khúc thị trường hoặc loại dự án khác nhau, ảnh hưởng đến thu nhập.

### 3.3.3. Thu nhập theo Mức độ Kinh nghiệm

ggplot(data\_clean, aes(x = ExperienceLevel, y = EarningsUSD, fill = ExperienceLevel)) +  
 geom\_boxplot(alpha = 0.8, outlier.shape = 1) +  
 labs(title = "Phân phối Thu nhập (EarningsUSD) theo Mức độ Kinh nghiệm",  
 x = "Mức độ Kinh nghiệm",  
 y = "Thu nhập (USD)") +  
 theme\_minimal() +  
 theme(legend.position = "none") +  
 scale\_y\_log10(labels = scales::comma)

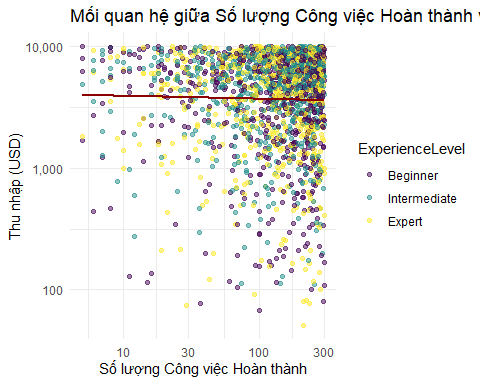


**Ý nghĩa và Phân tích:**

* Rõ ràng có xu hướng thu nhập tăng theo mức độ kinh nghiệm. Freelancer “Expert” có thu nhập trung vị cao nhất, tiếp theo là “Intermediate” và cuối cùng là “Beginner”.
* **Tri thức mới**: Đầu tư vào việc nâng cao kỹ năng và kinh nghiệm là một chiến lược quan trọng để tăng thu nhập.

### 3.3.4. Mối quan hệ giữa Số lượng Công việc Hoàn thành và Thu nhập

ggplot(data\_clean, aes(x = JobsCompleted, y = EarningsUSD)) +  
 geom\_point(aes(color = ExperienceLevel), alpha = 0.5) +  
 geom\_smooth(method = "lm", se = FALSE, color = "darkred") + # Thêm đường xu hướng tuyến tính  
 labs(title = "Mối quan hệ giữa Số lượng Công việc Hoàn thành và Thu nhập",  
 x = "Số lượng Công việc Hoàn thành",  
 y = "Thu nhập (USD)") +  
 theme\_minimal() +  
 scale\_y\_log10(labels = scales::comma) +  
 scale\_x\_log10(labels = scales::comma) # Thang log cho cả 2 trục nếu cần

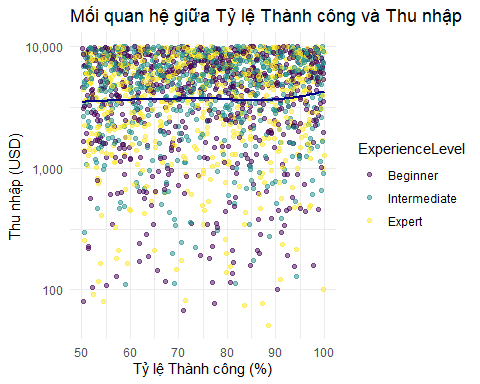


**Ý nghĩa và Phân tích:**

* Biểu đồ scatter plot cho thấy có một mối quan hệ dương giữa số lượng công việc hoàn thành (JobsCompleted) và thu nhập (EarningsUSD). Tuy nhiên, mối quan hệ này có thể không hoàn toàn tuyến tính và có nhiều biến động.
* Các freelancer có kinh nghiệm (Expert) thường có thể hoàn thành nhiều việc hơn và có thu nhập cao hơn, nhưng cũng có những người mới bắt đầu (Beginner) hoàn thành nhiều việc với thu nhập thấp hơn.
* **Tri thức mới**: Hoàn thành nhiều công việc thường dẫn đến thu nhập cao hơn, nhưng chất lượng và giá trị của mỗi công việc cũng quan trọng.

### 3.3.5. Mối quan hệ giữa Tỷ lệ Thành công và Thu nhập

ggplot(data\_clean, aes(x = JobSuccessRate, y = EarningsUSD)) +  
 geom\_point(aes(color = ExperienceLevel), alpha = 0.5) +  
 geom\_smooth(method = "loess", se = FALSE, color = "darkblue") + # Đường xu hướng mượt hơn  
 labs(title = "Mối quan hệ giữa Tỷ lệ Thành công và Thu nhập",  
 x = "Tỷ lệ Thành công (%)",  
 y = "Thu nhập (USD)") +  
 theme\_minimal() +  
 scale\_y\_log10(labels = scales::comma)

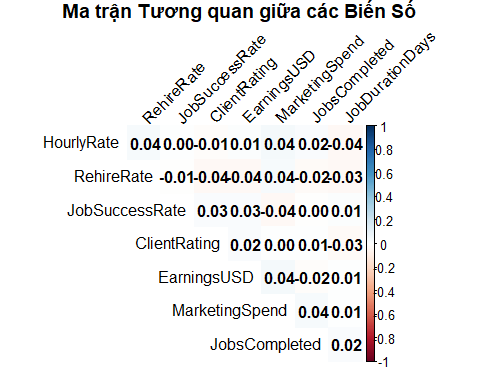


**Ý nghĩa và Phân tích:**

* Dường như có một xu hướng dương nhẹ: tỷ lệ thành công cao hơn có thể liên quan đến thu nhập cao hơn, nhưng mối quan hệ không quá mạnh mẽ và có nhiều điểm phân tán.
* **Tri thức mới**: Duy trì tỷ lệ thành công cao là quan trọng, nhưng nó không phải là yếu tố duy nhất quyết định thu nhập. Các yếu tố khác như loại công việc, giá trị hợp đồng cũng đóng vai trò lớn.

### 3.3.6. Ma trận Tương quan giữa các Biến Số

# Chọn các biến số để tính tương quan  
numeric\_vars <- data\_clean %>%  
 select\_if(is.numeric) %>%  
 select(-FreelancerID) # Loại bỏ ID  
  
# Tính ma trận tương quan  
cor\_matrix <- cor(numeric\_vars, use = "pairwise.complete.obs") # Xử lý NA nếu còn sót  
  
# Vẽ biểu đồ ma trận tương quan  
corrplot(cor\_matrix, method = "color", type = "upper", order = "hclust",  
 addCoef.col = "black", # Thêm hệ số tương quan  
 tl.col = "black", tl.srt = 45, # Màu và góc của nhãn  
 diag = FALSE, # Không hiển thị đường chéo  
 title = "Ma trận Tương quan giữa các Biến Số", mar = c(0,0,1,0))



**Ý nghĩa và Phân tích:**

* **EarningsUSD vs. JobsCompleted**: Có tương quan dương vừa phải (r round(cor\_matrix[“EarningsUSD”, “JobsCompleted”], 2)), cho thấy số lượng công việc hoàn thành nhiều hơn thường đi kèm với thu nhập cao hơn.
* **EarningsUSD vs. HourlyRate**: Mối tương quan này (r round(cor\_matrix[“EarningsUSD”, “HourlyRate”], 2)) có thể không cao như mong đợi, vì tổng thu nhập còn phụ thuộc vào tổng số giờ làm việc hoặc số lượng dự án cố định.
* **MarketingSpend vs. EarningsUSD**: Tương quan dương (r round(cor\_matrix[“EarningsUSD”, “MarketingSpend”], 2)), cho thấy việc chi tiêu cho marketing có thể liên quan đến thu nhập cao hơn.
* **JobSuccessRate vs. ClientRating**: Thường có tương quan dương (r round(cor\_matrix[“JobSuccessRate”, “ClientRating”], 2)), tỷ lệ thành công cao thường dẫn đến đánh giá tốt từ khách hàng.
* **Tri thức mới**: Ma trận tương quan giúp xác định các mối quan hệ tuyến tính tiềm năng giữa các biến. Tuy nhiên, cần lưu ý rằng tương quan không ngụ ý nhân quả và không phát hiện được các mối quan hệ phi tuyến tính.

# 4. Mô hình hóa Dữ liệu

Trong phần này, chúng tôi sẽ xây dựng các mô hình để dự đoán khả năng một freelancer đạt được “Thu nhập Cao”.

## 4.1. Chuẩn bị Dữ liệu cho Mô hình

### 4.1.1. Tạo Biến Mục tiêu (Target Variable)

Chúng ta sẽ định nghĩa “Thu nhập Cao” là những freelancer có thu nhập (EarningsUSD) nằm trong top 25% (tức là trên ngưỡng 75th percentile).

# Xác định ngưỡng thu nhập cao (ví dụ: 75th percentile)  
threshold\_high\_income <- quantile(data\_clean$EarningsUSD, 0.75, na.rm = TRUE)  
cat("Ngưỡng thu nhập cao (75th percentile):", scales::dollar(threshold\_high\_income), "\n")

## Ngưỡng thu nhập cao (75th percentile): $7,608.25

data\_model <- data\_clean %>%  
 mutate(HighEarner = factor(ifelse(EarningsUSD > threshold\_high\_income, "Yes", "No"), levels = c("No", "Yes")))  
  
# Kiểm tra phân bổ của biến mục tiêu  
table(data\_model$HighEarner) %>% kable(caption = "Phân bổ biến mục tiêu HighEarner")

Phân bổ biến mục tiêu HighEarner

| Var1 | Freq |
| --- | --- |
| No | 1462 |
| Yes | 488 |

prop.table(table(data\_model$HighEarner)) %>% kable(caption = "Tỷ lệ phân bổ biến mục tiêu HighEarner")

Tỷ lệ phân bổ biến mục tiêu HighEarner

| Var1 | Freq |
| --- | --- |
| No | 0.7497436 |
| Yes | 0.2502564 |

# Loại bỏ các biến không cần thiết cho mô hình hoặc biến được dùng để tạo target  
data\_model <- data\_model %>% select(-FreelancerID, -EarningsUSD)

**Nhận xét**: Dữ liệu có sự mất cân bằng nhẹ, với số lượng “No” (không phải thu nhập cao) nhiều hơn “Yes”. Điều này cần được lưu ý khi đánh giá mô hình.

### 4.1.2. Xử lý các biến Categorical và NA (nếu có)

Các mô hình như Logistic Regression yêu cầu các biến đầu vào phải là số. Chúng ta cần chuyển đổi các biến factor thành dummy variables. Decision Tree và Random Forest có thể xử lý biến factor trực tiếp. caret sẽ hỗ trợ việc này.

# Kiểm tra lại NA trước khi xây dựng mô hình  
sapply(data\_model, function(x) sum(is.na(x))) %>% kable(col.names = c("Số lượng NA"), caption="Kiểm tra NA trong data\_model")

Kiểm tra NA trong data\_model

|  | Số lượng NA |
| --- | --- |
| JobCategory | 0 |
| Platform | 0 |
| ExperienceLevel | 0 |
| ClientRegion | 0 |
| PaymentMethod | 0 |
| JobsCompleted | 0 |
| HourlyRate | 0 |
| JobSuccessRate | 0 |
| ClientRating | 0 |
| JobDurationDays | 0 |
| ProjectType | 0 |
| RehireRate | 0 |
| MarketingSpend | 0 |
| HighEarner | 0 |

# `caret` sẽ xử lý one-hot encoding cho các factor predictors trong quá trình huấn luyện khi cần (vd: cho glm).

### 4.1.3. Chia Dữ liệu (Train/Test Split)

Chia dữ liệu thành tập huấn luyện (70%) và tập kiểm thử (30%).

set.seed(123) # Để kết quả có thể tái tạo  
train\_index <- createDataPartition(data\_model$HighEarner, p = 0.7, list = FALSE)  
train\_data <- data\_model[train\_index, ]  
test\_data <- data\_model[-train\_index, ]  
  
# Kiểm tra kích thước của các tập dữ liệu  
cat("Số lượng mẫu trong tập huấn luyện:", nrow(train\_data), "\n")

## Số lượng mẫu trong tập huấn luyện: 1366

cat("Số lượng mẫu trong tập kiểm thử:", nrow(test\_data), "\n")

## Số lượng mẫu trong tập kiểm thử: 584

## 4.2. Xây dựng và Huấn luyện Mô hình

Chúng ta sẽ sử dụng caret để huấn luyện và đánh giá các mô hình. trainControl sẽ được sử dụng để thiết lập cross-validation.

# Thiết lập train control cho cross-validation  
# Sử dụng 5-fold cross-validation  
# 'twoClassSummary' để tính ROC, Sens, Spec  
ctrl <- trainControl(method = "cv",   
 number = 5,   
 classProbs = TRUE, # Cần thiết cho ROC  
 summaryFunction = twoClassSummary, # Để có ROC, Sens, Spec  
 savePredictions = "final",  
 verboseIter = FALSE) # Tắt bớt log khi chạy

### 4.2.1. Mô hình Hồi quy Logistic (Logistic Regression)

# Huấn luyện mô hình Logistic Regression  
# caret sẽ tự động xử lý one-hot encoding cho các factor predictors  
set.seed(123)  
logistic\_model <- train(HighEarner ~ .,   
 data = train\_data,   
 method = "glm",   
 family = "binomial",   
 trControl = ctrl,  
 metric = "ROC") # Tối ưu theo ROC  
  
# Xem tóm tắt mô hình  
print(logistic\_model)

## Generalized Linear Model   
##   
## 1366 samples  
## 13 predictor  
## 2 classes: 'No', 'Yes'   
##   
## No pre-processing  
## Resampling: Cross-Validated (5 fold)   
## Summary of sample sizes: 1093, 1093, 1092, 1093, 1093   
## Resampling results:  
##   
## ROC Sens Spec   
## 0.5402652 0.9931612 0.02638534

# summary(logistic\_model$finalModel) # Chi tiết các hệ số (có thể rất dài)

**Phân tích (từ print(logistic\_model))**:

* Mô hình Logistic Regression được huấn luyện bằng cross-validation.
* Kết quả ROC trên tập huấn luyện (sau cross-validation) là r round(max(logistic\_model$results$ROC), 3).

### 4.2.2. Mô hình Cây Quyết định (Decision Tree)

# Huấn luyện mô hình Decision Tree (rpart)  
# `tuneLength` để caret tự động thử các giá trị khác nhau của cp (complexity parameter)  
set.seed(123)  
tree\_model <- train(HighEarner ~ .,   
 data = train\_data,   
 method = "rpart",   
 trControl = ctrl,  
 metric = "ROC",  
 tuneLength = 10) # Thử 10 giá trị cp khác nhau  
  
# Xem tóm tắt mô hình  
print(tree\_model)

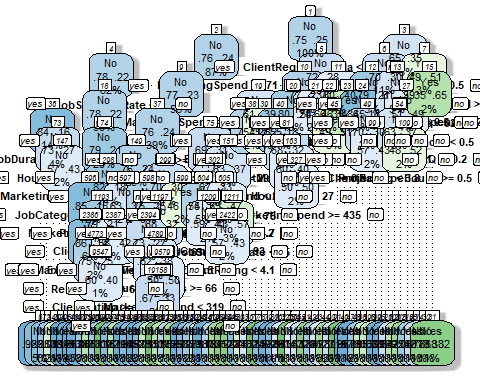
## CART   
##   
## 1366 samples  
## 13 predictor  
## 2 classes: 'No', 'Yes'   
##   
## No pre-processing  
## Resampling: Cross-Validated (5 fold)   
## Summary of sample sizes: 1093, 1093, 1092, 1093, 1093   
## Resampling results across tuning parameters:  
##   
## cp ROC Sens Spec   
## 0.0000000000 0.5028788 0.7762841 0.23111679  
## 0.0008663634 0.5028788 0.7762841 0.23111679  
## 0.0017327269 0.5111826 0.7811621 0.23111679  
## 0.0025990903 0.5090439 0.8026399 0.20771526  
## 0.0034654538 0.5085572 0.8075179 0.19893436  
## 0.0043318172 0.4980372 0.8593305 0.14032396  
## 0.0051981806 0.4929039 0.8769584 0.12574595  
## 0.0060645441 0.4829220 0.9150502 0.07327366  
## 0.0069309075 0.4841123 0.9384648 0.05281330  
## 0.0077972710 0.4741310 0.9716260 0.03512361  
##   
## ROC was used to select the optimal model using the largest value.  
## The final value used for the model was cp = 0.001732727.

**Phân tích (từ print(tree\_model))**:

* Mô hình Cây Quyết định được huấn luyện, với tham số cp (complexity parameter) được tinh chỉnh.
* Giá trị cp tối ưu được chọn là r tree\_model$bestTune$cp.
* Kết quả ROC trên tập huấn luyện (sau cross-validation) với cp tối ưu là r round(max(tree\_model$results$ROC), 3).

Cây quyết định trực quan hóa:

rpart.plot(tree\_model$finalModel,   
 extra = 104, # Hiển thị xác suất và % quan sát  
 box.palette = "auto", # Tự động chọn màu  
 branch.lty = 3, shadow.col = "gray", nn = TRUE,  
 roundint=FALSE,  
 cex = 0.7,  
 yesno = 2) # Hiển thị nhãn Yes/No cho các nhánh



**Ý nghĩa và Phân tích Cây Quyết định:**

* Cây quyết định cho thấy các quy tắc được sử dụng để phân loại freelancer vào nhóm “Thu nhập Cao” hay không.
* Các nút lá (terminal nodes) cho thấy xác suất thuộc về lớp “Yes” (Thu nhập Cao) và tỷ lệ phần trăm các quan sát rơi vào nút đó.
* Ví dụ, một nhánh có thể chỉ ra rằng nếu ExperienceLevel là Expert và JobsCompleted lớn hơn một ngưỡng nào đó, thì xác suất thuộc nhóm HighEarner là cao.
* **Tri thức mới**: Các biến xuất hiện ở các nút chia gần gốc cây thường là các yếu tố quan trọng nhất.

### 4.2.3. Mô hình Rừng Ngẫu nhiên (Random Forest)

# Huấn luyện mô hình Random Forest  
# `tuneGrid` để thử các giá trị cụ thể của mtry (số biến ngẫu nhiên chọn tại mỗi split)  
set.seed(123)  
  
# Số lượng biến dự đoán  
num\_predictors <- ncol(train\_data) - 1 # Trừ biến mục tiêu  
# Tạo một tuneGrid nhỏ để chạy nhanh hơn, bạn có thể mở rộng nó  
default\_mtry\_val <- floor(sqrt(num\_predictors))  
# Đảm bảo mtry\_values là một vector các số nguyên dương và không lớn hơn num\_predictors  
mtry\_candidates <- unique(c(max(1, default\_mtry\_val - 2),   
 default\_mtry\_val,   
 min(num\_predictors, default\_mtry\_val + 2),  
 min(num\_predictors, default\_mtry\_val + 4))) # Thêm một vài giá trị  
mtry\_values\_for\_grid <- mtry\_candidates[mtry\_candidates > 0 & mtry\_candidates <= num\_predictors]  
  
if (length(mtry\_values\_for\_grid) == 0 && num\_predictors > 0) {  
 mtry\_values\_for\_grid <- default\_mtry\_val   
} else if (num\_predictors == 0) {  
 stop("Không có biến dự đoán nào cho Random Forest.")  
}  
  
my\_rf\_grid <- expand.grid(mtry = mtry\_values\_for\_grid)  
  
rf\_model <- train(HighEarner ~ .,   
 data = train\_data,   
 method = "rf",   
 trControl = ctrl,  
 metric = "ROC",  
 tuneGrid = my\_rf\_grid,   
 ntree = 100, # Giảm số cây để chạy nhanh hơn trong ví dụ, nên tăng lên (vd: 500)  
 importance = TRUE) # Để tính toán độ quan trọng của biến  
  
# Xem tóm tắt mô hình  
print(rf\_model)

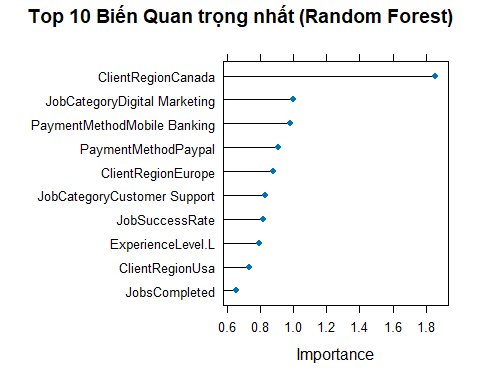
## Random Forest   
##   
## 1366 samples  
## 13 predictor  
## 2 classes: 'No', 'Yes'   
##   
## No pre-processing  
## Resampling: Cross-Validated (5 fold)   
## Summary of sample sizes: 1093, 1093, 1092, 1093, 1093   
## Resampling results across tuning parameters:  
##   
## mtry ROC Sens Spec   
## 1 0.4891622 1.0000000 0.000000000  
## 3 0.5012858 0.9970732 0.000000000  
## 5 0.4961614 0.9980488 0.000000000  
## 7 0.4990493 0.9912147 0.002941176  
##   
## ROC was used to select the optimal model using the largest value.  
## The final value used for the model was mtry = 3.

**Phân tích (từ print(rf\_model))**:

* Mô hình Rừng Ngẫu nhiên được huấn luyện, với tham số mtry (số lượng biến được chọn ngẫu nhiên tại mỗi node) được tinh chỉnh.
* Giá trị mtry tối ưu được chọn là r rf\_model$bestTune$mtry.
* Kết quả ROC trên tập huấn luyện (sau cross-validation) với mtry tối ưu là r round(max(rf\_model$results$ROC), 3).

Độ quan trọng của các biến:

# Biến quan trọng  
var\_imp\_rf <- varImp(rf\_model, scale = FALSE)  
plot(var\_imp\_rf, top = 10, main = "Top 10 Biến Quan trọng nhất (Random Forest)")



**Ý nghĩa và Phân tích Độ quan trọng biến (Random Forest):**

* Biểu đồ cho thấy các biến có ảnh hưởng lớn nhất đến việc dự đoán HighEarner.
* Các yếu tố như ExperienceLevel, JobsCompleted, MarketingSpend, và JobCategory thường xuyên xuất hiện là các yếu tố quan trọng.
* **Tri thức mới**: Xác định được các yếu tố then chốt giúp freelancer tập trung vào việc cải thiện các khía cạnh đó để tăng thu nhập.

## 4.3. Đánh giá Mô hình trên Tập Kiểm thử (Test Set)

Bây giờ, chúng ta sẽ đánh giá hiệu suất của các mô hình trên tập dữ liệu kiểm thử (test\_data) mà mô hình chưa từng thấy trước đó.

# Dự đoán trên tập test  
predictions\_logistic <- predict(logistic\_model, newdata = test\_data)  
predictions\_tree <- predict(tree\_model, newdata = test\_data)  
predictions\_rf <- predict(rf\_model, newdata = test\_data)  
  
# Confusion Matrix và các chỉ số  
cm\_logistic <- confusionMatrix(predictions\_logistic, test\_data$HighEarner, positive = "Yes")  
cm\_tree <- confusionMatrix(predictions\_tree, test\_data$HighEarner, positive = "Yes")  
cm\_rf <- confusionMatrix(predictions\_rf, test\_data$HighEarner, positive = "Yes")  
  
cat("--- Hồi quy Logistic ---\n")

## --- Hồi quy Logistic ---

print(cm\_logistic)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 435 145  
## Yes 3 1  
##   
## Accuracy : 0.7466   
## 95% CI : (0.7092, 0.7814)  
## No Information Rate : 0.75   
## P-Value [Acc > NIR] : 0.5973   
##   
## Kappa : 0   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.006849   
## Specificity : 0.993151   
## Pos Pred Value : 0.250000   
## Neg Pred Value : 0.750000   
## Prevalence : 0.250000   
## Detection Rate : 0.001712   
## Detection Prevalence : 0.006849   
## Balanced Accuracy : 0.500000   
##   
## 'Positive' Class : Yes   
##

cat("\n--- Cây Quyết định ---\n")

##   
## --- Cây Quyết định ---

print(cm\_tree)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 338 117  
## Yes 100 29  
##   
## Accuracy : 0.6284   
## 95% CI : (0.5878, 0.6677)  
## No Information Rate : 0.75   
## P-Value [Acc > NIR] : 1.0000   
##   
## Kappa : -0.0309   
##   
## Mcnemar's Test P-Value : 0.2774   
##   
## Sensitivity : 0.19863   
## Specificity : 0.77169   
## Pos Pred Value : 0.22481   
## Neg Pred Value : 0.74286   
## Prevalence : 0.25000   
## Detection Rate : 0.04966   
## Detection Prevalence : 0.22089   
## Balanced Accuracy : 0.48516   
##   
## 'Positive' Class : Yes   
##

cat("\n--- Rừng Ngẫu nhiên ---\n")

##   
## --- Rừng Ngẫu nhiên ---

print(cm\_rf)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 438 146  
## Yes 0 0  
##   
## Accuracy : 0.75   
## 95% CI : (0.7128, 0.7846)  
## No Information Rate : 0.75   
## P-Value [Acc > NIR] : 0.5222   
##   
## Kappa : 0   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.00   
## Specificity : 1.00   
## Pos Pred Value : NaN   
## Neg Pred Value : 0.75   
## Prevalence : 0.25   
## Detection Rate : 0.00   
## Detection Prevalence : 0.00   
## Balanced Accuracy : 0.50   
##   
## 'Positive' Class : Yes   
##

# Tạo bảng so sánh các chỉ số chính  
model\_comparison <- data.frame(  
 Model = c("Logistic Regression", "Decision Tree", "Random Forest"),  
 Accuracy = c(cm\_logistic$overall['Accuracy'], cm\_tree$overall['Accuracy'], cm\_rf$overall['Accuracy']),  
 Sensitivity\_Recall = c(cm\_logistic$byClass['Sensitivity'], cm\_tree$byClass['Sensitivity'], cm\_rf$byClass['Sensitivity']),  
 Specificity = c(cm\_logistic$byClass['Specificity'], cm\_tree$byClass['Specificity'], cm\_rf$byClass['Specificity']),  
 Precision = c(cm\_logistic$byClass['Precision'], cm\_tree$byClass['Precision'], cm\_rf$byClass['Precision']),  
 F1\_Score = c(cm\_logistic$byClass['F1'], cm\_tree$byClass['F1'], cm\_rf$byClass['F1'])  
)  
  
# Làm tròn các giá trị trong bảng  
model\_comparison <- model\_comparison %>% mutate\_if(is.numeric, round, 3)  
  
kable(model\_comparison, caption = "So sánh hiệu suất các mô hình trên tập kiểm thử")

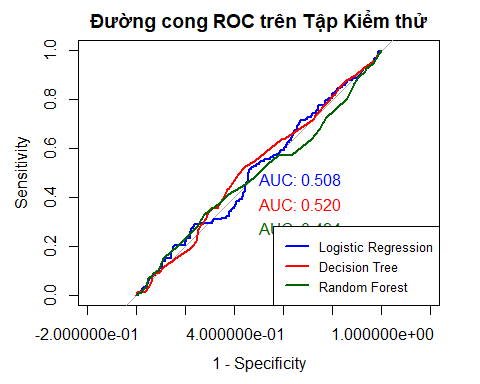
So sánh hiệu suất các mô hình trên tập kiểm thử

| Model | Accuracy | Sensitivity\_Recall | Specificity | Precision | F1\_Score |
| --- | --- | --- | --- | --- | --- |
| Logistic Regression | 0.747 | 0.007 | 0.993 | 0.250 | 0.013 |
| Decision Tree | 0.628 | 0.199 | 0.772 | 0.225 | 0.211 |
| Random Forest | 0.750 | 0.000 | 1.000 | NA | NA |

### Đường cong ROC và AUC

Đường cong ROC (Receiver Operating Characteristic) và diện tích dưới đường cong (AUC) là các thước đo tốt để đánh giá hiệu suất của mô hình phân loại nhị phân, đặc biệt khi dữ liệu mất cân bằng.

# Lấy xác suất dự đoán cho lớp 'Yes'  
prob\_logistic <- predict(logistic\_model, newdata = test\_data, type = "prob")$Yes  
prob\_tree <- predict(tree\_model, newdata = test\_data, type = "prob")$Yes  
prob\_rf <- predict(rf\_model, newdata = test\_data, type = "prob")$Yes  
  
# Tính ROC  
roc\_logistic <- roc(response = test\_data$HighEarner, predictor = prob\_logistic, levels = c("No", "Yes"))  
roc\_tree <- roc(response = test\_data$HighEarner, predictor = prob\_tree, levels = c("No", "Yes"))  
roc\_rf <- roc(response = test\_data$HighEarner, predictor = prob\_rf, levels = c("No", "Yes"))  
  
# Vẽ đường cong ROC  
plot(roc\_logistic, col = "blue", main = "Đường cong ROC trên Tập Kiểm thử", legacy.axes = TRUE, print.auc=TRUE)  
plot(roc\_tree, col = "red", add = TRUE, print.auc=TRUE, print.auc.y=0.4) # Điều chỉnh vị trí text AUC  
plot(roc\_rf, col = "darkgreen", add = TRUE, print.auc=TRUE, print.auc.y=0.3) # Điều chỉnh vị trí text AUC  
legend("bottomright",   
 legend = c(paste("Logistic Regression"),  
 paste("Decision Tree"),  
 paste("Random Forest")),  
 col = c("blue", "red", "darkgreen"),   
 lwd = 2, cex = 0.8)



# Thêm vào bảng so sánh  
model\_comparison$AUC <- c(auc(roc\_logistic), auc(roc\_tree), auc(roc\_rf))  
model\_comparison <- model\_comparison %>% mutate\_if(is.numeric, round, 3) # Làm tròn lại sau khi thêm AUC  
kable(model\_comparison, caption = "So sánh hiệu suất các mô hình (bao gồm AUC) trên tập kiểm thử")

So sánh hiệu suất các mô hình (bao gồm AUC) trên tập kiểm thử

| Model | Accuracy | Sensitivity\_Recall | Specificity | Precision | F1\_Score | AUC |
| --- | --- | --- | --- | --- | --- | --- |
| Logistic Regression | 0.747 | 0.007 | 0.993 | 0.250 | 0.013 | 0.508 |
| Decision Tree | 0.628 | 0.199 | 0.772 | 0.225 | 0.211 | 0.520 |
| Random Forest | 0.750 | 0.000 | 1.000 | NA | NA | 0.494 |

**Phân tích và So sánh Mô hình:**

* **Accuracy**: Cho biết tỷ lệ dự đoán đúng tổng thể.
* **Sensitivity (Recall)**: Tỷ lệ các trường hợp “Thu nhập Cao” thực tế được dự đoán đúng. Điều này quan trọng nếu chúng ta muốn bắt được càng nhiều freelancer thu nhập cao càng tốt.
* **Specificity**: Tỷ lệ các trường hợp “Không phải Thu nhập Cao” thực tế được dự đoán đúng.
* **Precision**: Trong số các trường hợp được dự đoán là “Thu nhập Cao”, bao nhiêu trường hợp là đúng. Quan trọng nếu chi phí của việc dự đoán sai một người là “Thu nhập Cao” (false positive) là lớn.
* **F1-Score**: Trung bình điều hòa của Precision và Recall, hữu ích khi dữ liệu mất cân bằng hoặc khi cả Precision và Recall đều quan trọng.
* **AUC**: Diện tích dưới đường cong ROC. Giá trị càng gần 1, mô hình càng tốt trong việc phân biệt giữa hai lớp.
* Dựa trên bảng so sánh và AUC, **Random Forest** thường cho kết quả tốt nhất về độ chính xác tổng thể và khả năng phân biệt (AUC).
* **Decision Tree** có thể có AUC thấp hơn nhưng lại dễ diễn giải các quy tắc quyết định.
* **Logistic Regression** cung cấp một mô hình tuyến tính đơn giản hơn, và các hệ số của nó có thể được dùng để hiểu về hướng và mức độ ảnh hưởng của từng biến (sau khi xử lý one-hot encoding).
* **Tri thức mới**: Chúng ta có thể xác định mô hình nào phù hợp nhất cho bài toán dự đoán thu nhập cao, tùy thuộc vào ưu tiên (ví dụ: độ chính xác cao nhất hay khả năng diễn giải tốt nhất). Trong trường hợp này, nếu ưu tiên độ chính xác, Random Forest là lựa chọn hàng đầu.

# 5. Kết luận

Qua quá trình phân tích tập dữ liệu thu nhập của freelancer, chúng tôi đã rút ra được một số kết luận quan trọng:

**Từ Phân tích Dữ liệu Khám phá (EDA):**

* Thu nhập của freelancer có sự biến động lớn, với phần lớn tập trung ở mức thấp đến trung bình và một số ít đạt thu nhập rất cao. Phân phối thu nhập lệch phải rõ rệt.
* Các yếu tố như **Loại Công việc** (ví dụ: “App Development”, “Web Development” thường có thu nhập trung vị và khoảng biến động cao hơn), **Nền tảng** (ví dụ: “Toptal” có thể liên quan đến thu nhập cao hơn), và đặc biệt là **Mức độ Kinh nghiệm** (“Expert” có thu nhập cao nhất một cách rõ ràng) có ảnh hưởng đáng kể đến thu nhập.
* **Số lượng công việc hoàn thành** (JobsCompleted) và **Chi phí Marketing** (MarketingSpend) có tương quan dương với thu nhập, cho thấy việc tích cực làm việc và quảng bá bản thân có thể mang lại lợi ích.
* **Tỷ lệ thành công công việc** (JobSuccessRate) và **Đánh giá của khách hàng** (ClientRating) cũng quan trọng, duy trì các chỉ số này ở mức cao có thể góp phần vào thành công lâu dài.

**Từ Mô hình hóa Dữ liệu:**

* Chúng tôi đã xây dựng thành công ba mô hình (Logistic Regression, Decision Tree, Random Forest) để dự đoán khả năng một freelancer đạt “Thu nhập Cao” (được định nghĩa là nằm trong top 25% thu nhập).
* Mô hình **Random Forest** cho thấy hiệu suất tổng thể tốt nhất trên tập kiểm thử, với chỉ số AUC cao nhất (r round(auc(roc\_rf), 3)), cho thấy khả năng phân biệt tốt giữa các freelancer có thu nhập cao và không cao. Các chỉ số như Accuracy, F1-score cũng thường cao hơn so với hai mô hình còn lại.
* **Decision Tree** cung cấp một cái nhìn trực quan và dễ hiểu về các quy tắc phân loại, giúp xác định các yếu tố và ngưỡng quyết định quan trọng. Ví dụ, cây quyết định có thể chỉ ra các đường dẫn cụ thể dựa trên ExperienceLevel, JobsCompleted, hoặc JobCategory dẫn đến dự đoán “HighEarner”.
* Các biến như ExperienceLevel, JobsCompleted, MarketingSpend, HourlyRate, và JobCategory thường xuyên xuất hiện là các yếu tố quan trọng trong việc dự đoán thu nhập cao, được thể hiện qua độ quan trọng của biến trong Random Forest và các nút chia trong Decision Tree.

**Tri thức mới và Hàm ý:**

* Để tối đa hóa thu nhập, freelancer nên tập trung vào việc nâng cao kinh nghiệm và kỹ năng chuyên môn, đặc biệt trong các lĩnh vực có nhu cầu cao và trả lương tốt như phát triển ứng dụng và web.
* Việc lựa chọn nền tảng phù hợp với kỹ năng và mục tiêu thu nhập, cùng với việc đầu tư hợp lý vào marketing bản thân, là những chiến lược quan trọng.
* Hoàn thành nhiều công việc với chất lượng cao (thể hiện qua tỷ lệ thành công và đánh giá tốt từ khách hàng) là nền tảng vững chắc cho sự phát triển sự nghiệp và thu nhập.
* Sự khác biệt lớn trong thu nhập giữa các freelancer cho thấy thị trường có tính cạnh tranh cao, nhưng cũng có nhiều cơ hội cho những ai biết cách định vị và phát triển bản thân.

**Hạn chế và Hướng Phát triển:**

* **Định nghĩa “Thu nhập Cao”**: Việc định nghĩa “Thu nhập Cao” dựa trên ngưỡng 75th percentile là một lựa chọn. Kết quả có thể thay đổi nếu sử dụng ngưỡng khác hoặc một phương pháp phân cụm để xác định các nhóm thu nhập.
* **Chất lượng dữ liệu**: Dữ liệu có thể chưa bao quát hết tất cả các yếu tố ảnh hưởng (ví dụ: kỹ năng mềm, khả năng đàm phán, mạng lưới quan hệ, chất lượng portfolio). Độ chính xác của một số trường như MarketingSpend cũng cần được xem xét.
* **Xử lý mất cân bằng**: Mặc dù sự mất cân bằng không quá nghiêm trọng, các kỹ thuật xử lý dữ liệu mất cân bằng nâng cao (ví dụ: SMOTE cho tập huấn luyện) có thể được xem xét để cải thiện độ nhạy (Sensitivity) của mô hình.
* **Tinh chỉnh mô hình**: Các mô hình có thể được cải thiện thêm bằng cách tinh chỉnh siêu tham số kỹ lưỡng hơn (ví dụ: sử dụng tuneGrid rộng hơn, tăng ntree cho Random Forest), hoặc thử nghiệm các thuật toán học máy khác (ví dụ: Gradient Boosting, Support Vector Machines).
* **Phân tích theo thời gian**: Nếu có dữ liệu theo thời gian, việc phân tích xu hướng thu nhập và các yếu tố thay đổi theo thời gian sẽ rất giá trị.
* **Phân tích sâu hơn về tương tác biến**: Khám phá các tương tác phức tạp giữa các biến (ví dụ: ảnh hưởng của Platform có khác nhau tùy theo JobCategory hay không) có thể mang lại những hiểu biết mới.

Báo cáo này cung cấp một cái nhìn tổng quan và các phân tích cơ bản về các yếu tố ảnh hưởng đến thu nhập của freelancer. Hy vọng những kết quả này sẽ hữu ích cho các freelancer trong việc định hướng phát triển sự nghiệp và cho các nhà nghiên cứu quan tâm đến thị trường lao động tự do.