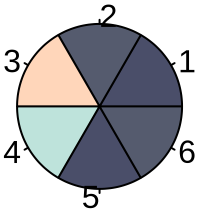
Plots of the restaurants analysis

Contents

```
library(gridExtra)
library(cowplot)
library(grid)
library(basicUtils)
library(tidyverse)
library(ggthemes)
```

```
to_knit = T
l = getColors("greenbeige")
```



```
baseColor = l$baseColor
bgColor = l$darkColor
if (to_knit == T) {
  baseSize = 9
  shape_size = 2
  annot_size = 2
  line_size = 0.5
} else{
  baseSize = 21
  shape_size = 5
  annot_size = 6
  line_size = 2
}
highlightColor = l$highlightColor
baseShape = 7
highlightShape = 15
fontTheme = 'sans'
serror = function(x)
  sqrt(var(x, na.rm = T) / length(x))
bgColor = l$highlightColor
```

1 Define necessary functions

```
get_within_opening_perf_plot <- function() {
  d <-
    read.csv(
      paste(
        "../../data/",
        dataset,
        "/evaluation_results/within_openings",
        dataset,
        ".csv",
        sep = ""
      )
    ) %>% filter(n == 3)
  get_focal_imprvs = function(algorithm) {
    new_d = get_imprvement(d, algorithm) %>% bind_cols(d %>% select(fold))
    new_d %>% rename(improvement = algorithm) %>% mutate(model = algorithm) %>%
      group_by(model) %>% summarise(avg = mean(improvement),
                                    se = serror(improvement)) -> new_d
    return(new_d)
  }
  baselines = unique(d$algorithm)
  baselines = baselines[baselines != 'hmm']
  d %>% pivot_wider(names_from = algorithm, values_from = score) -> d
  baselines %>% map_dfr(get_focal_imprvs) -> r
  r %>% update_model_names %>% ggplot(aes(
    x = model,
    y = avg,
    ymin = avg - 1.645 * se,
    ymax = avg + 1.645 * se,
    color = '1',
    shape = '1'
  )) +
    geom_point(size = shape_size) +
    geom_errorbar(width = 0.2) +
    ylab("Improvement (%)") +
    xlab(expression(paste(""))) +
    theme_hc(base_size = baseSize) +
    theme(strip.background = element_rect(fill = alpha(bgColor, 0.4))) +
    scale_color_manual(values = c(baseColor, baseColor, baseColor)) +
    scale_shape_manual(name = "",
                       values = c(baseShape, baseShape, baseShape)) +
    theme(legend.position = "none") +
    geom_abline(slope = 0,
                intercept = 0,
                linetype = 'dashed') +
    theme(axis.title = element_text()) +
    facet_wrap(
  ~"Within-task improvement of the HMM framework over alternative recommenders",
      strip.position = "top"
    )
}
```

```
update_model_names <- function(k1) {
  k1$model <- factor(k1$model)
  levels(k1$model)[levels(k1$model) == "svdtrinaryexplicit_worker"] = "SVD\n(explicit,\noption
  levels(k1$model)[levels(k1$model) == "svdbinaryworker"] = "SVD\n(implicit,\noption 1)"
  levels(k1$model)[levels(k1$model) == "svdtrinaryworker"] = "SVD\n(p-aware,\noption 1)"
  levels(k1$model)[levels(k1$model) == "svdbinary"] = "SVD\n(implicit,\noption 2)"
  levels(k1$model)[levels(k1$model) == "svdtrinary"] = "SVD\n(p-aware,\noption 2)"
  levels(k1$model)[levels(k1$model) == "svdtrinaryexplicit"] = "SVD\n(explicit,\noption 2)"
  levels(k1$model)[levels(k1$model) == "explicit_SVD"] = "SVD\n(explicit)"
  levels(k1$model)[levels(k1$model) == "cnn"] = "CNN\n(implicit)"
  levels(k1$model)[levels(k1$model) == "lstm"] = "LSTM"
  levels(k1$model)[levels(k1$model) == "svm"] = "SVM"
  levels(k1$model)[levels(k1$model) == "xg"] = "XGBoost"
  levels(k1$model)[levels(k1$model) == "trinary_SVD"] = "SVD\n(p-aware)"
  levels(k1$model)[levels(k1$model) == "sahoo"] = "HMM-CF\n(implicit)"
  levels(k1$model)[levels(k1$model) == "logit"] = "Lgistic\nregression"
  levels(k1$model)[levels(k1$model) == "rf"] = "Random\nforest"
  levels(k1$model)[levels(k1$model) == "hmm"] = "HMM"
  levels(k1$model)[levels(k1$model) == "fb"] = "Reputation\n(implicit)"
  if ('metric' %in% colnames(k1)) {
    k1$metric <- factor(k1$metric)
    levels(k1$metric)[levels(k1$metric) == "negSR"] <-
      "Within-opening likelihood of 'Hire-negative'"
    levels(k1$metric)[levels(k1$metric) == "topVsBottomSR"] <-
      "Within-opening performance"
  }
  levels(k1$model)[levels(k1$model) == "diff"] <- "Difference"
  levels(k1$model)[levels(k1$model) == "rate25"] <-
    "75%-Hire-negative\n25%-Hire-positive"
  levels(k1$model)[levels(k1$model) == "rate75"] <-
    "25%-Hire-negative\n75%-Hire-positive"
  levels(k1$model)[levels(k1$model) == "ratio"] <- "Odds ratio"
  return(k1)
}
```

```
get_imprvement = function(d,algorithm){
  new_d = ((d$hmm - d[,algorithm])/d[,algorithm]) * 100
  return(new_d)
}
```

```
get_labels = function(r, level) {
  #regression labels
  t = tibble()
  avg = se = 1
  for (model in unique(r$model)) {
    r[r$model == model,] -> cur_r
    res = lm(avg ~ n, cur_r)
    from = confint(res, 'n', level = level)[1]
    to = confint(res, 'n', level = level)[2]
    intercept = res$coefficients[1]
    #position to print the conf-int
    yposition = ifelse(intercept > 1,
                       0.9 * intercept,
                ifelse(intercept > 0, -20 * intercept, 2 * intercept))#intercept
    t %>% bind_rows(tibble(from, to, model, yposition, avg, se)) -> t
  }
  return(t)
}
```

```
get_ranking_performance_plot = function() {
  get_focal_imprvs = function(algorithm) {
    new_d = get_imprvement(d, algorithm) %>% bind_cols(d %>% select(prc, fold))
    new_d %>% rename(improvement = algorithm) %>% mutate(model = algorithm) %>%
      group_by(model, prc) %>% summarise(avg = mean(improvement),
                                         se = serror(improvement)) -> new_d
    return(new_d)
  }
  d = read.csv(
    paste(
      "../../data/",
      dataset,
      "/evaluation_results/ranking_performance",
      dataset,
      ".csv",
      sep = ""
    )
  )
  baselines = unique(d$algorithm)
  baselines = baselines[baselines != 'hmm']
  d %>% pivot_wider(names_from = algorithm, values_from = score) -> d
  baselines %>% map_dfr(get_focal_imprvs) -> r
  r %>%  update_model_names %>%  ggplot(aes(
    x = prc,
    y = avg,
    ymin = (avg - 1.645 * se),
    ymax = (avg + 1.645 * se),
    color = baseColor,
    fill = '1'
  )) +
    geom_col(alpha = 0.4, size = line_size) +
    facet_wrap( ~ model, ncol = 6, scales = "free_y") +
    geom_errorbar(width = 0.3) +
    scale_color_manual(values = c(baseColor), name = "Improvement over") +
    scale_fill_manual(values = c(bgColor), name = "Improvement over") +
    xlab("") +
    scale_x_continuous(breaks = c(0.5, 1),
                       labels = c('Bottom 50%', 'Top 50%')) +
    geom_hline(yintercept = 0, linetype = "dashed") +
    geom_vline(xintercept = 0.5,
               linetype = "dashed",
               alpha = 0.2) +
    ylab(" Performance lift (%)") +
    theme_hc(base_size = baseSize) +
    theme(strip.background = element_rect(fill = alpha(bgColor, 0.4))) +
    theme(legend.position = "none") +
    theme(axis.title = element_text()) -> p
  return(p)
}
```

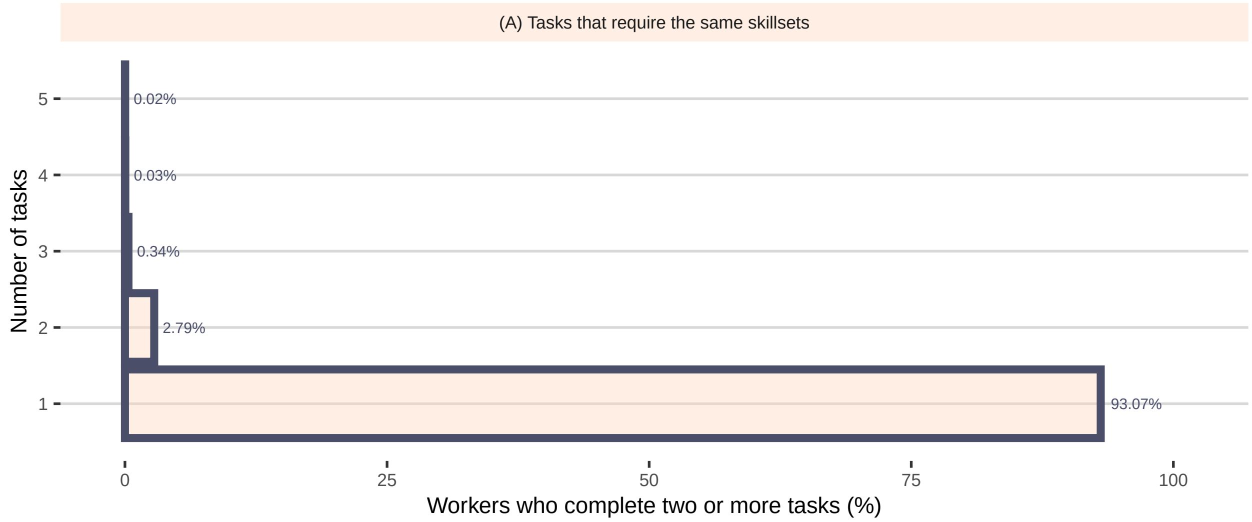
```
get_auc_n_plot = function() {
  get_focal_imprvs = function(algorithm) {
    new_d = get_imprvement(d, algorithm) %>% bind_cols(d %>% select(n, fold))
    new_d %>% rename(improvement = algorithm) %>% mutate(model = algorithm) %>%
      group_by(model, n) %>% summarise(avg = mean(improvement),
                                       se = serror(improvement)) -> new_d
    return(new_d)
  }
  d = read.csv(
    paste(
      "../../data/",
      "/evaluation_results/auc_n",
      ".csv",
      sep = ""
    )
  )
  baselines = unique(d$algorithm)
  baselines = baselines[baselines != 'hmm']
  d %>% pivot_wider(names_from = algorithm, values_from = score) -> d
  baselines %>% map_dfr(get_focal_imprvs) -> r
  r
  get_labels(r, level = 0.9) %>% update_model_names -> t
  t
  r %>% update_model_names %>%
    ggplot(aes(
      x = n,
      y = avg,
      ymin = (avg - 1.645 * se),
      ymax = (avg + 1.645 * se),
      color = baseColor,
      shape = '1'
    )) +
    geom_smooth(
      aes(fill = baseColor),
      method = 'lm',
      alpha = 0.2,
      size = line_size
    ) +
    scale_color_manual(values = c(baseColor), name = "Improvement over") +
    scale_fill_manual(values = c(bgColor), name = "Improvement over") +
    scale_shape_manual(values = c(baseShape), name = "Improvement over") +
    xlab("") +
    geom_text(
      t,
      mapping = aes(
        x = 4,
        y = yposition,
        group = model,
        label = paste("Slope 90% CI: [", round(from, 2), ", ",
                      round(to, 2), "]", sep = ""),
        size = annot_size,
        color = baseColor
      ),
      check_overlap = T,
      inherit.aes = FALSE,
      size = annot_size
    ) +
    scale_x_continuous(breaks = c(1, 3, 5, 7)) +
    facet_wrap( ~ model, ncol = 6,  scales = "free_y") +
    geom_hline(yintercept = 0, linetype = "dashed") +
    ylab("AUC Improvement (%)") +
    xlab("Employer experience (number of completed tasks)") +
    theme_hc(base_size = baseSize) +
    theme(strip.background = element_rect(fill = alpha(bgColor, 0.4))) +
    theme(legend.position = "none") +
    theme(axis.title = element_text()) -> p
  p
  return(p)
}
```

2 File explanation:

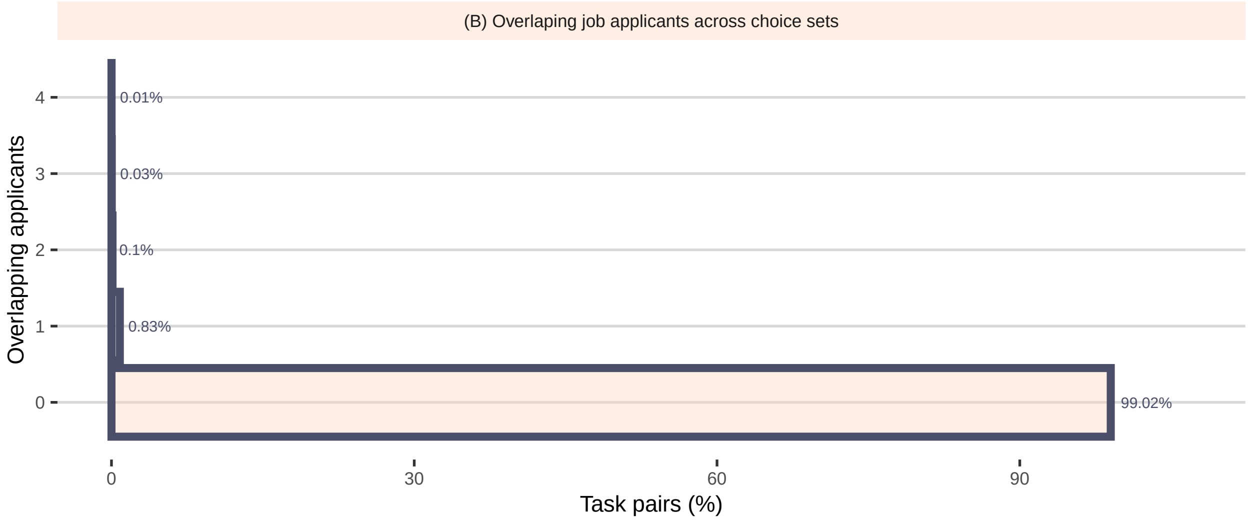
- final_main_ncv.csv: the file that goes into our models for nested cross validation.
- dataRaw.csv: the file that includes the raw query results from the market's database.

3 Figure 2: Comparison with many-assessment: Data evidence

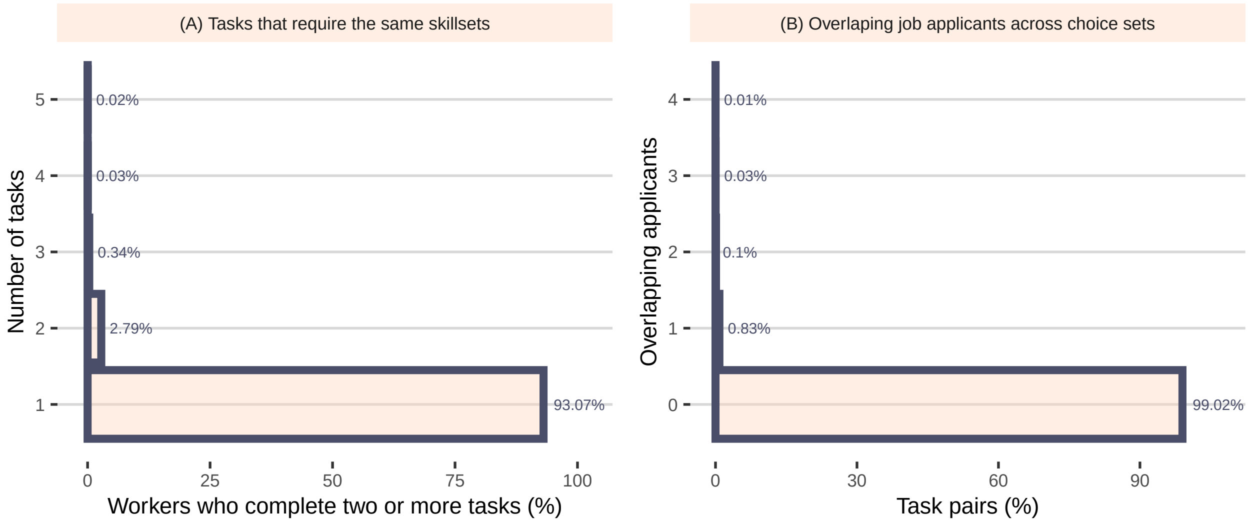
```
d = read_csv("../../data/final_main_ncv.csv")
d %>% filter(outcome > 0) %>%  group_by(worker) %>%
  summarize(n = n_distinct(task_id)) %>% filter(n > 1) -> d1
d %>% filter(outcome > 0 & worker %in% unique(d1$worker)) -> d2
r = read_csv("../../../r0/data/dataRaw.csv") %>%
  filter(application %in% d2$application) %>% filter(!is.na(openingSkills))
r %>%  group_by(contractor) %>% summarize(n = n_distinct(opening)) %>%
  filter(n > 1) -> r1
r1 = r %>% filter(contractor %in% r1$contractor)
r1  %>% count(contractor, openingSkills) %>% count(n) %>%
  mutate(prc = (nn / nrow(r1)) * 100) -> k
k %>% ggplot(aes(
  x = n,
  y = prc,
  color = 'r',
  fill = 'r'
)) + geom_col(alpha = 0.4, size = 1.5) +
  coord_flip() +
  geom_text(
    hjust = -0.2,
    aes(label = paste(round(prc, 2), "%", sep = "")),
    size = annot_size,
    color = baseColor
  ) +
  ylab("Workers who complete two or more tasks (%)") +
  xlab("Number of tasks") +
  theme_hc(base_size = baseSize) +
  theme(strip.background = element_rect(fill = alpha(bgColor, 0.4))) +
  scale_color_manual(values = c(baseColor, baseColor, baseColor)) +
  scale_fill_manual(name = "",
                    values = c(bgColor, baseColor, baseColor)) +
  theme(legend.position = "none") +
  scale_x_continuous(limits = c(0.5, 5.5), breaks = c(1, 2, 3, 4, 5)) +
  scale_y_continuous(limits = c(-1, 102)) +
  theme(axis.title = element_text()) +
  facet_wrap(~ "(A) Tasks that require the same skillsets", strip.position = "top") -> p1
p1
```



```
d = read_csv(
  "../../data/evaluation_results/overlaping_tasks.csv",
  col_names = c('overlapping_workers')
)
d %>% count(overlapping_workers) %>% mutate(prc = (n / nrow(d)) * 100) ->
  k
k %>% ggplot(aes(
  x = overlapping_workers,
  y = prc,
  color = 'r',
  fill = 'r'
)) + geom_col(alpha = 0.4, size = 1.5) +
  coord_flip() +
  geom_text(
    hjust = -0.2,
    aes(label = paste(round(prc, 2), "%", sep = "")),
    size = annot_size,
    color = baseColor
  ) +
  ylab("Task pairs (%)") +
  xlab("Overlapping applicants") +
  theme_hc(base_size = baseSize) +
  theme(strip.background = element_rect(fill = alpha(bgColor, 0.4))) +
  scale_color_manual(values = c(baseColor, baseColor, baseColor)) +
  scale_fill_manual(name = "",
                    values = c(bgColor, baseColor, baseColor)) +
  theme(legend.position = "none") + scale_x_continuous(limits = c(-0.5, 4.5),
                                                       breaks = c(0, 1, 2, 3, 4)) +
  scale_y_continuous(limits = c(-0, 107)) +
  theme(axis.title = element_text()) +
  facet_wrap( ~ "(B) Overlaping job applicants across choice sets",
              strip.position = "top") -> p2
p2
```



```
bottom_row <- plot_grid(p1, p2, label_size = 16, ncol = 2)
g <- grid.arrange(arrangeGrob(bottom_row))
```



```
ggsave(
  file = "../../../../../../../Apps/Overleaf/gbu/ms/r2/figures/data-evidence.pdf",
  width = 18,
  height = 6,
  dpi = 300,
  g
)
```

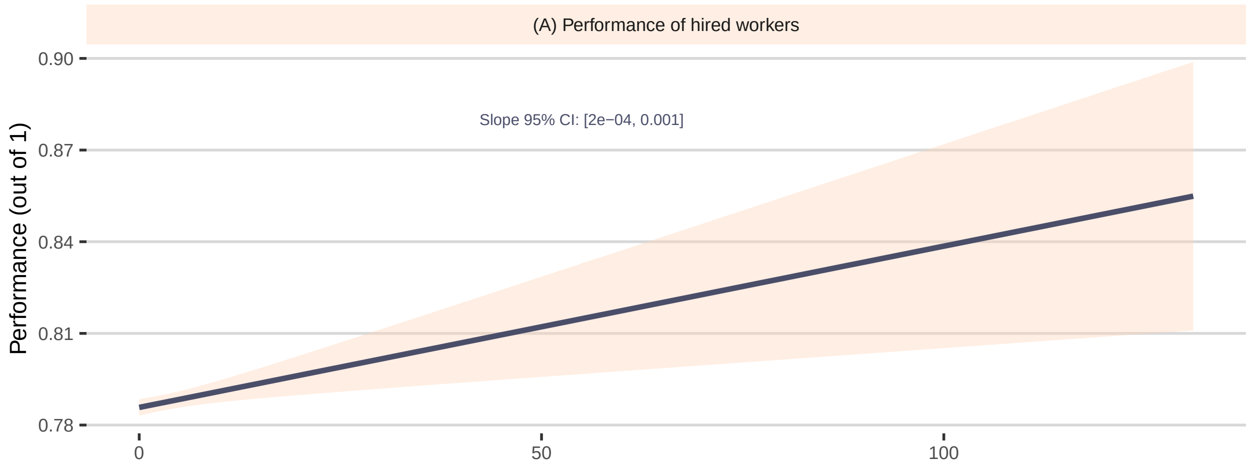
4 Table 1: Descriptive stats

```
d = read_csv("../../data/final_main_ncv.csv")
cat("--- Overall dataset description ---\n")
cat("\nNumber of unique employers:", length(unique(d$employer)))
cat("\nNumber of job applications:", nrow(d))
cat("\nNumber of completed tasks (number of hires):",
    nrow(d %>% filter(Y_it > 0)))
cat("\n\n--- Descriprtive statistics  (latex,tabular format) ---\n\n")
ivs = c(
  'employer_money_spent_so_far',
  'employer_outcome_of_last_task',
  'employer_total_tasks_so_far',
  'employer_fixed_contracts_so_far',
  'employer_hourly_contracts_so_far',
  'employer_successful_tasks_so_far',
  'employer_successful_fixed_contracts_so_far',
  'employer_successful_hourly_contracts_so_far',
  'applicant_completed_work_hours',
  'skills_ip',
  'bid',
  'received_application_order',
  'reputation_score',
  'jobs',
  'countries_pmi',
  'self_reported_experience',
  'exams_pmi',
  'invited',
  'contract_type'
)
printDescrStats(as.data.frame(d %>% select(ivs)), ivs, T)
dvs = c('private_feedback_normalized')
printDescrStats(as.data.frame(d %>% select(dvs)), dvs, T)
```

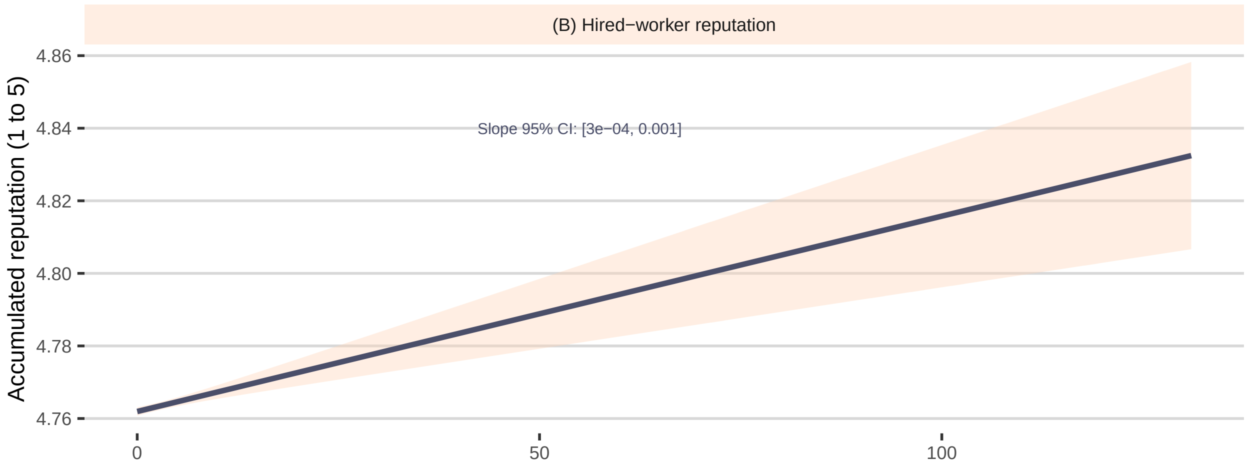
```
## --- Overall dataset description ---
##
## Number of unique employers: 11461
## Number of job applications: 762802
## Number of completed tasks (number of hires): 45331
##
## --- Descriprtive statistics  (latex,tabular format) ---
##
## & \employermoneyspentsofar  &  178  &  4  &  841  &  0  &  36806 \\ \addlinespace
## & \employeroutcomeoflasttask  &  0.62  &  1  &  0.49  &  0  &  1 \\ \addlinespace
## & \employertotaltaskssofar  &  2.3  &  1  &  6  &  0  &  131 \\ \addlinespace
## & \employerfixedcontractssofar  &  1.5  &  0  &  4.7  &  0  &  94 \\ \addlinespace
## & \employerhourlycontractssofar  &  0.84  &  0  &  3.1  &  0  &  120 \\ \addlinespace
## & \employersuccessfultaskssofar  &  1.4  &  0  &  4  &  0  &  104 \\ \addlinespace
## & \employersuccessfulfixedcontractssofar  &  0.96  &  0  &  3.3  &  0  &  89 \\ \addlinespac
## & \employersuccessfulhourlycontractssofar  &  0.46  &  0  &  2.1  &  0  &  99 \\ \addlinespa
## & \applicantcompletedworkhours  &  578  &  45  &  1472  &  0  &  37766 \\ \addlinespace
## & \skillsip  &  1  &  1  &  1.2  &  0  &  18 \\ \addlinespace
## & \bid  &  89  &  11  &  549  &  1  &  50000 \\ \addlinespace
## & \receivedapplicationorder  &  26  &  15  &  32  &  0  &  291 \\ \addlinespace
## & \reputationscore  &  4.8  &  4.9  &  0.4  &  1  &  5 \\ \addlinespace
## & \jobs  &  5  &  0  &  15  &  0  &  403 \\ \addlinespace
## & \countriespmi  &  -0.49  &  -0.46  &  0.4  &  -3.7  &  4.4 \\ \addlinespace
## & \selfreportedexperience  &  4.5  &  4  &  4.1  &  0  &  30 \\ \addlinespace
## & \examspmi  &  2.3  &  2.1  &  1.7  &  -0.71  &  7 \\ \addlinespace
## & \invited  &  0.15  &  0  &  0.35  &  0  &  1 \\ \addlinespace
## & \contracttype  &  0.48  &  0  &  0.5  &  0  &  1 \\ \addlinespace
## & \privatefeedbacknormalized  &  0.79  &  0.89  &  0.28  &  0  &  1 \\ \addlinespace
```

5 Figure 5: Model-free employer behavior

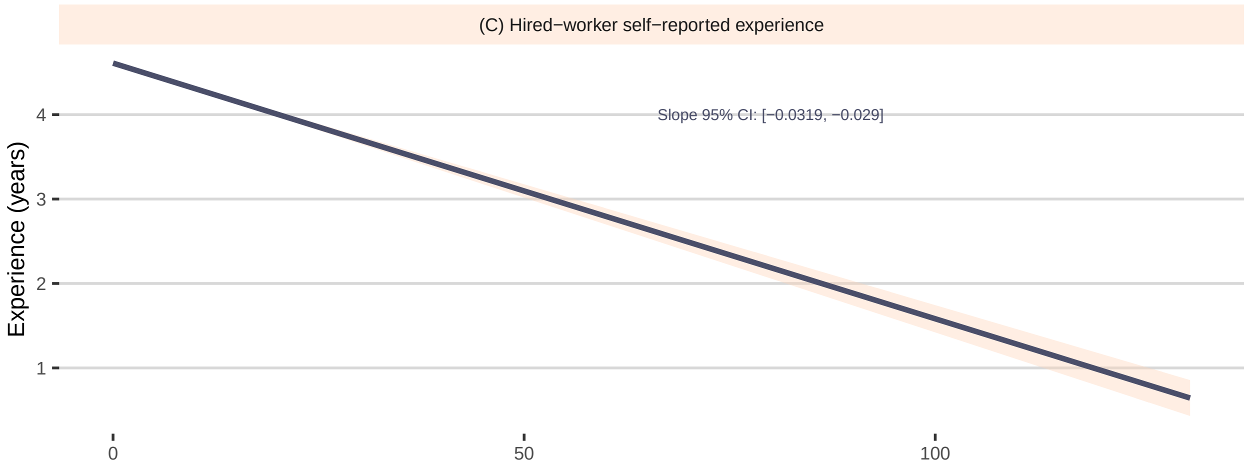
```r
res = lm(private_feedback_normalized ~ employer_total_tasks_so_far, d)
from = confint(res, 'employer_total_tasks_so_far', level = 0.95)[1]
to = confint(res, 'employer_total_tasks_so_far', level = 0.95)[2]
d %>%  ggplot(
  aes(
    x = employer_total_tasks_so_far,
    y = private_feedback_normalized,
    color = 'r',
    fill = 'r'
  )
) +
  geom_smooth(method = 'lm')  +
  scale_colour_manual(values = c(baseColor), name = "") +
  scale_fill_manual(values = c(bgColor), name = "") +
  theme_hc(base_size = baseSize) +
  theme(strip.background = element_rect(fill = alpha(bgColor, 0.4))) +
  theme(legend.position = "None") +
  annotate(
    "text",
    x = c(55),
    y = c(0.88),
    color = baseColor,
    label = paste("Slope 95% CI: [", round(from, 4), ", ", round(to, 3), "]", sep = ""),
    parse = F,
    size = annot_size
  ) +
  xlab("") +
  ylab("Performance (out of 1)") +
  theme(plot.margin = unit(c(1, 0, 0, 1), "cm"))+
  theme(axis.title = element_text()) +
  facet_wrap( ~ "(A) Performance of hired workers",
            strip.position = "top") -> p1

p1
```



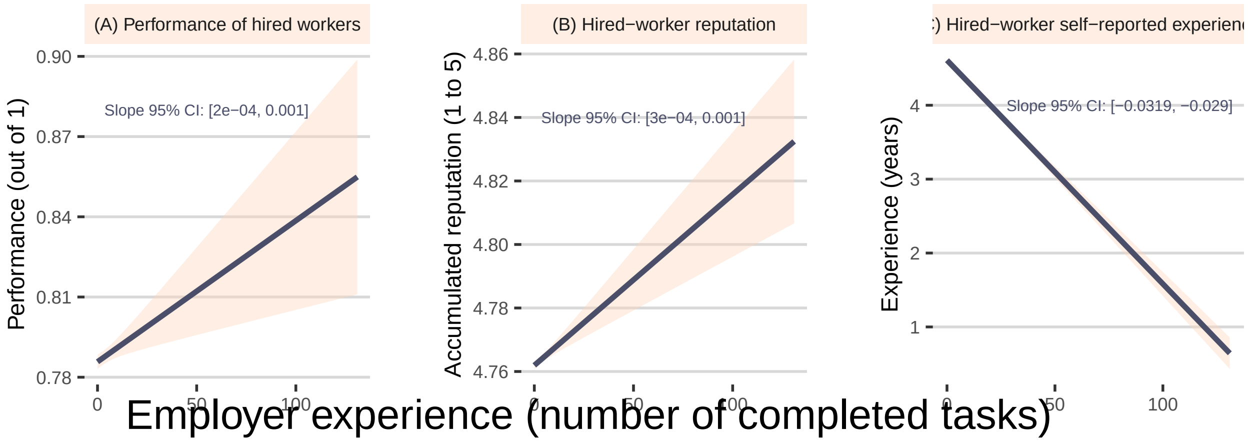
```r
res = lm(reputation_score ~ employer_total_tasks_so_far, d)
from = confint(res, 'employer_total_tasks_so_far', level = 0.95)[1]
to = confint(res, 'employer_total_tasks_so_far', level = 0.95)[2]
d %>%  ggplot(aes(
  x = employer_total_tasks_so_far,
  y = reputation_score,
  color = 'r',
  fill = 'r'
)) +
  geom_smooth(method = 'lm')  +
  scale_colour_manual(values = c(baseColor), name = "") +
  scale_fill_manual(values = c(bgColor), name = "") +
  theme_hc(base_size = baseSize) +
  theme(strip.background = element_rect(fill = alpha(bgColor, 0.4))) +
  theme(legend.position = "None") +
  annotate(
    "text",
    x = c(55),
    y = c(4.84),
    color = baseColor,
    label = paste("Slope 95% CI: [", round(from, 4), ", ", round(to, 3), "]", sep = ""),
    parse = F,
    size = annot_size
  ) +
  xlab("") +
  ylab("Accumulated reputation (1 to 5)") +
  theme(plot.margin = unit(c(1, 0, 0, 1), "cm")) +
  theme(axis.title = element_text()) +
  facet_wrap(~ "(B) Hired-worker reputation",
             strip.position = "top") -> p2
p2
```



```
res = lm(self_reported_experience ~ employer_total_tasks_so_far, d)
from = confint(res, 'employer_total_tasks_so_far', level = 0.95)[1]
to = confint(res, 'employer_total_tasks_so_far', level = 0.95)[2]
d %>%  ggplot(
  aes(
    x = employer_total_tasks_so_far,
    y = self_reported_experience,
    color = 'r',
    fill = 'r'
  )
) +
  geom_smooth(method = 'lm')  +
  scale_colour_manual(values = c(baseColor), name = "") +
  scale_fill_manual(values = c(bgColor), name = "") +
  theme_hc(base_size = baseSize) +
  theme(strip.background = element_rect(fill = alpha(bgColor, 0.4))) +
  theme(legend.position = "None") +
  annotate(
    "text",
    x = c(80),
    y = c(4),
    color = baseColor,
    label = paste("Slope 95% CI: [", round(from, 4), ", ", round(to, 3), "]", sep = ""),
    parse = F,
    size = annot_size
  ) +
  xlab("") +
  ylab("Experience (years)") +
  theme(plot.margin = unit(c(1, 0, 0, 1), "cm")) +
  theme(axis.title = element_text()) +
  facet_wrap( ~ "(C) Hired-worker self-reported experience",
              strip.position = "top") -> p3
p3
```



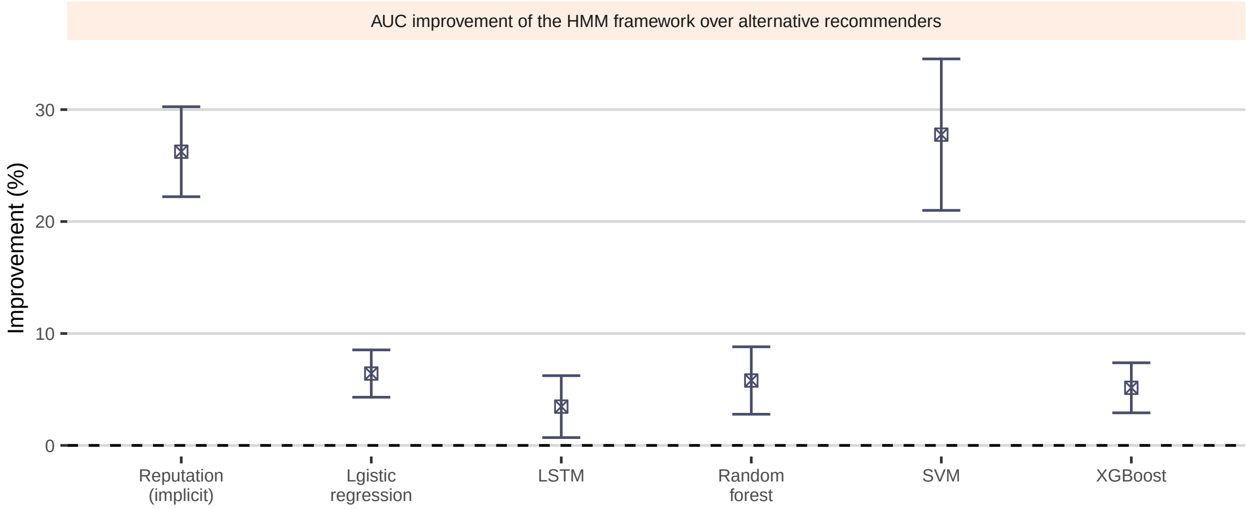
```
bottom_row <- plot_grid(p1, p2, p3,  label_size = 16, ncol = 3)
x.grob <-
  textGrob(
    "Employer experience (number of completed tasks)",
    gp = gpar(fontsize = 16),
    rot = 0,
    vjust = -1.3
  )
g <- grid.arrange(arrangeGrob(bottom_row, bottom = x.grob))
```



```
ggsave(
  file = "../../../../../../../../Apps/Overleaf/gbu/ms/r2/figures/model-free.pdf",
  width = 17,
  height = 6,
  dpi = 300,
  g
)
```

6 Figure 7: AUC comparison

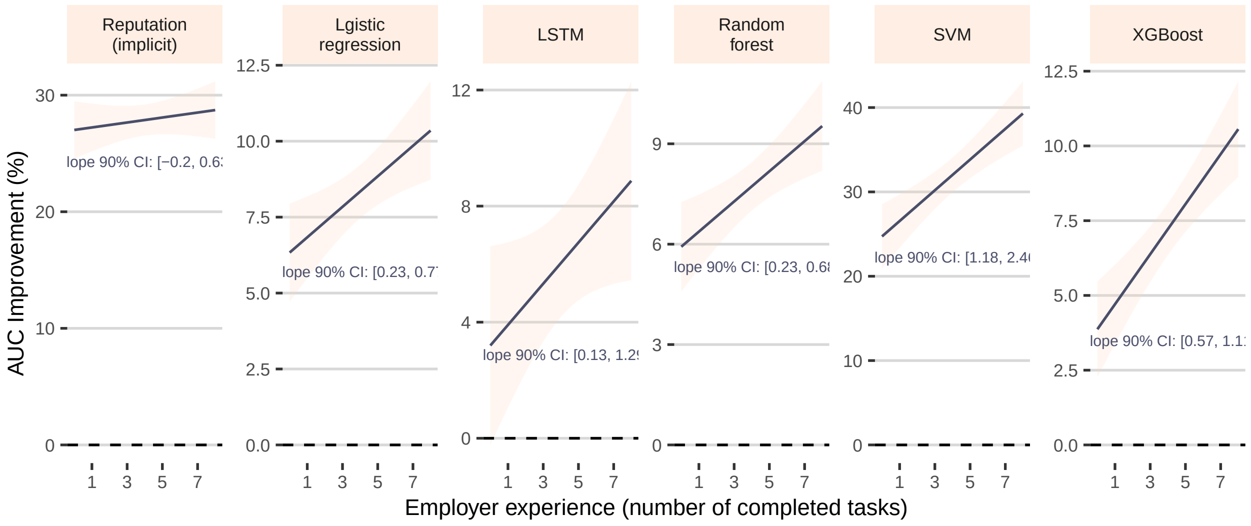
```r
dataset = "" #70, 90
d = read_csv(paste(
  "../../data/",
  dataset,
  "/evaluation_results/auc_n",
  dataset,
  ".csv",
  sep = ""
)) %>% filter(n == 0)
get_focal_imprvs = function(algorithm) {
  new_d = get_imprvement(d, algorithm) %>% bind_cols(d %>% select(fold))
  new_d %>% rename(improvement = algorithm) %>% mutate(model = algorithm) %>%
    group_by(model) %>% summarise(avg = mean(improvement),
                                  se = serror(improvement)) -> new_d
  return(new_d)
}
baselines = unique(d$algorithm)
baselines = baselines[baselines != 'hmm']
d %>% select(fold, algorithm, score) %>%  pivot_wider(names_from = algorithm,
                                                      values_from = score) ->
  d
baselines %>% map_dfr(get_focal_imprvs) -> r
r %>% update_model_names %>% ggplot(aes(
  x = model,
  y = avg,
  ymin = avg - 1.645 * se,
  ymax = avg + 1.645 * se,
  color = 'r',
  shape = 'r'
)) +
  geom_point(size = shape_size) +
  geom_errorbar(width = 0.2) +
  ylab("Improvement (%)") +
  xlab(expression(paste(""))) +
  geom_vline(xintercept = c(10.5), linetype = "dashed") +
  theme_hc(base_size = baseSize) +
  theme(strip.background = element_rect(fill = alpha(bgColor, 0.4))) +
  facet_wrap( ~ "AUC improvement of the HMM framework over alternative recommenders",
              strip.position = "top") +

  scale_color_manual(values = c(baseColor, baseColor, baseColor)) +
  scale_shape_manual(name = "",
                     values = c(baseShape, baseShape, baseShape)) +
  theme(legend.position = "none") +
  geom_abline(slope = 0,
              intercept = 0,
              linetype = 'dashed') +
  theme(axis.title = element_text())
```



```r
ggsave(
  file = paste(
    "../../../../../../../../Apps/Overleaf/gbu/ms/r2/figures/",
    dataset,
    "auc",
    ".pdf",
    sep = ""
  ),
  width = 14,
  height = 5,
  dpi = 300
)
```

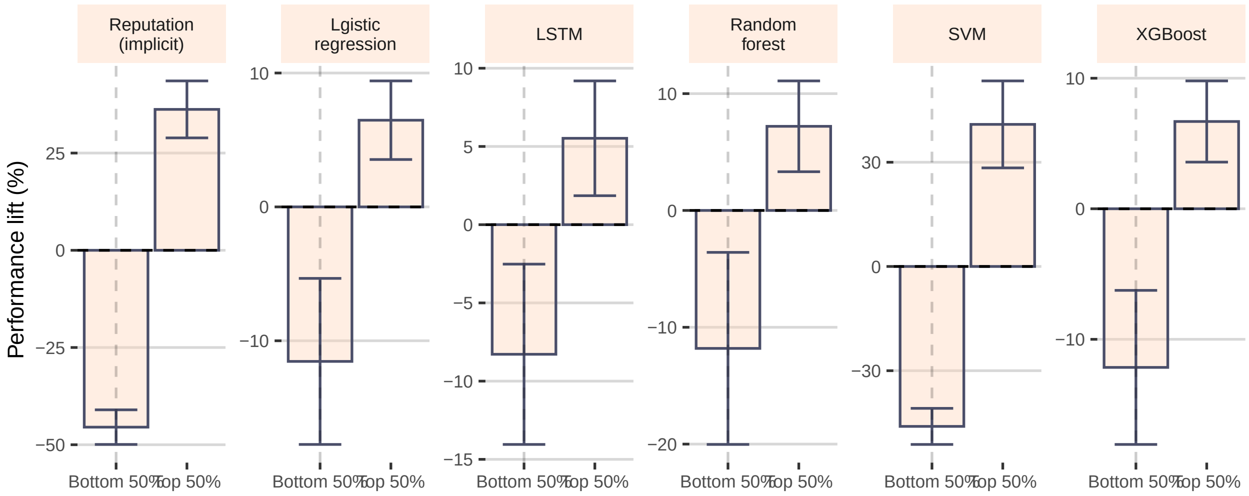
7 Figure 8: AUC-n comparison

```r
p = get_auc_n_plot()
ggsave(
  file = paste(
    "../../../../../../../../Apps/Overleaf/gbu/ms/r2/figures/",
    dataset,
    "aucn.pdf",
    sep = ""
  ),
  width = 20,
  height = 6,
  dpi = 300
)
p
```



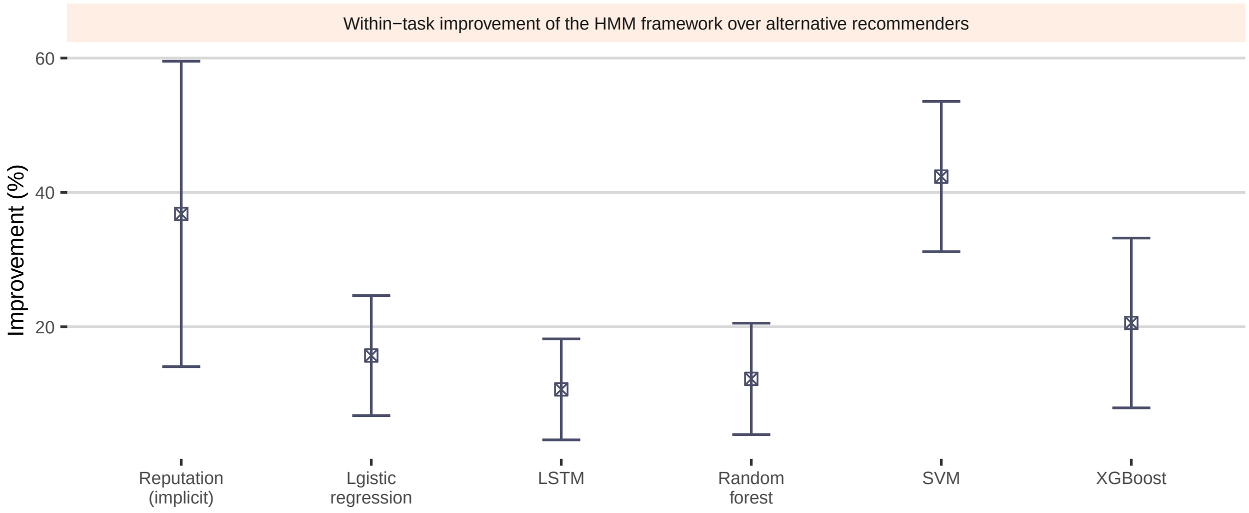
8 Figure 9: Performance lift

```
p = get_ranking_performance_plot()
ggsave(
  file = paste(
    "../../../../../../../Apps/Overleaf/gbu/ms/r2/figures/",
    dataset,
    "ranking_performance.pdf",
    sep = ""
  ),
  width = 18,
  height = 6,
  dpi = 300
)
p
```



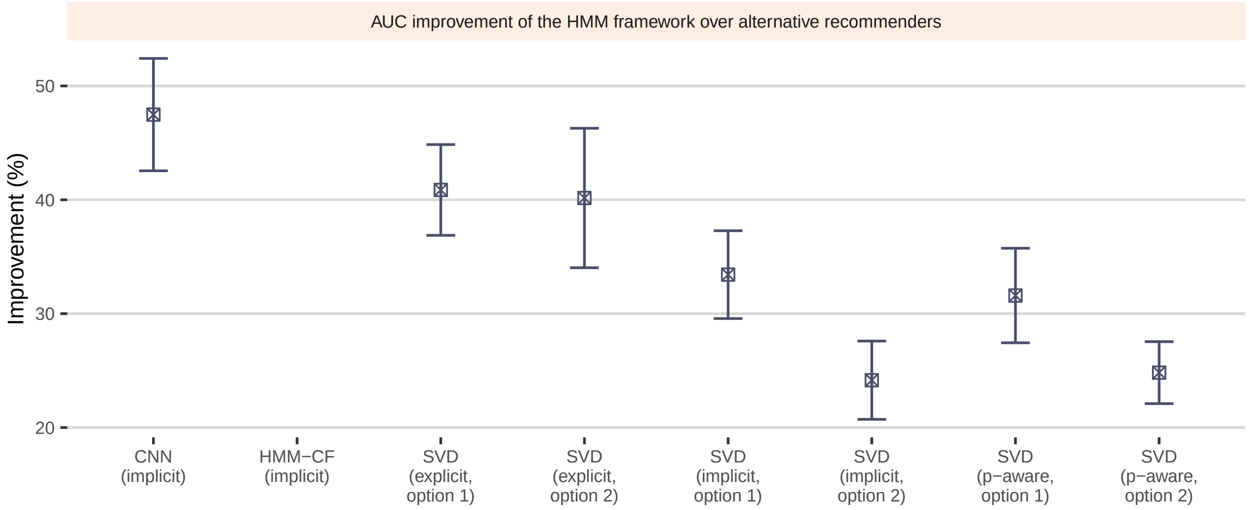
9 Figure 10: Performance within tasks

```
p = get_within_opening_perf_plot()
ggsave(
  file = paste(
    "../../../../../../../Apps/Overleaf/gbu/ms/r2/figures/",
    dataset,
    "withinOpenings.pdf",
    sep = ""
  ),
  width = 14,
  height = 5,
  dpi = 300
)
p
```



10 Figure 11: Many-assessment

```
d = read_csv(paste("../../data/evaluation_results/many_assessment.csv", sep =
                     ""))
get_focal_imprvs = function(algorithm) {
  new_d = get_imprvement(d, algorithm) %>% bind_cols(d %>% select(fold))
  new_d %>% rename(improvement = algorithm) %>% mutate(model = algorithm) %>%
    group_by(model) %>% summarise(avg = mean(improvement),
                                  se = serror(improvement)) -> new_d
  return(new_d)
}
baselines = unique(d$algorithm)
baselines = baselines[baselines != 'hmm']
d %>% select(fold, algorithm, score) %>%  pivot_wider(names_from = algorithm,
                                                      values_from = score) ->
  d
baselines %>% map_dfr(get_focal_imprvs) -> r


r %>% update_model_names %>% mutate(model = fct_relevel(model,sort)) %>%
  ggplot(aes(
  x = model,
  y = avg,
  ymin = avg - 1.645 * se,
  ymax = avg + 1.645 * se,
  color = 'r',
  shape = 'r'
)) +
  geom_point(size = shape_size) +
  geom_errorbar(width = 0.2) +
  ylab("Improvement (%)") +
  xlab(expression(paste(""))) +
  geom_vline(xintercept = c(10.5), linetype = "dashed") +
  theme_hc(base_size = baseSize) +
  theme(strip.background = element_rect(fill = alpha(bgColor, 0.4))) +
  facet_wrap( ~ "AUC improvement of the HMM framework over alternative recommenders",
             strip.position = "top") +

  scale_color_manual(values = c(baseColor, baseColor, baseColor)) +
  scale_shape_manual(name = "",
                     values = c(baseShape, baseShape, baseShape)) +
  theme(legend.position = "none") +
  geom_abline(slope = 0,
              intercept = 0,
              linetype = 'dashed') +
  theme(axis.title = element_text())
```

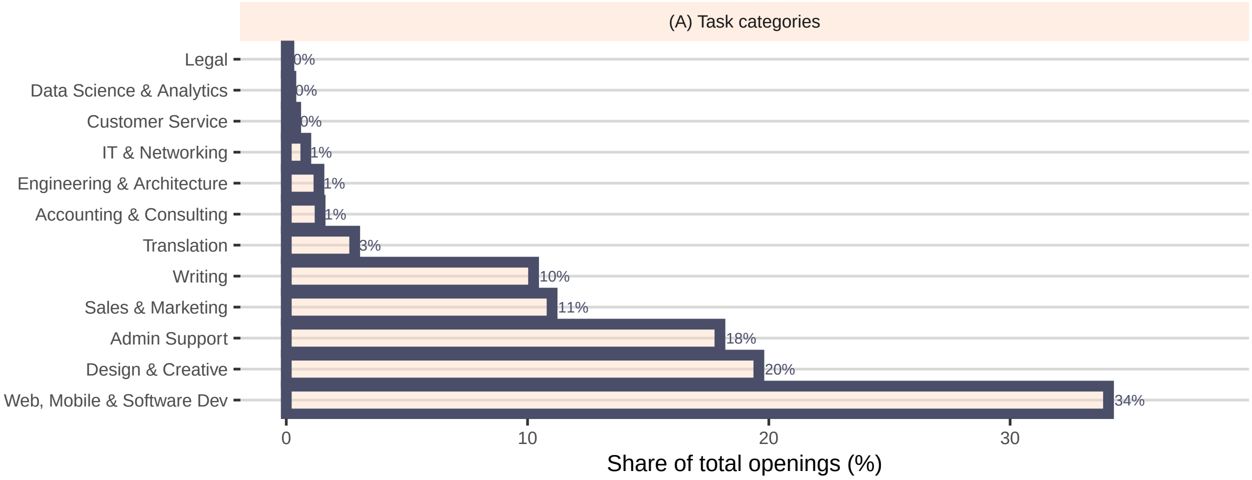


```
ggsave(
  file = paste(
    "../../../../../../../Apps/Overleaf/gbu/ms/r2/figures/many_assessment_auc.pdf",
    sep = ""
  ),
  width = 14,
  height = 5,
  dpi = 300
)
```

11 Figure 17: Market description

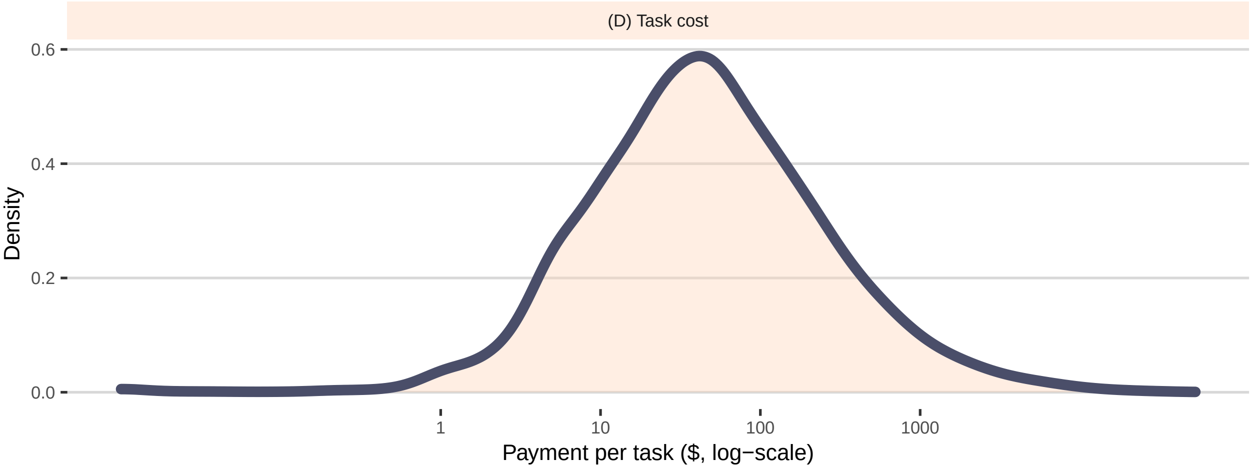
```
d <- read.csv("../../data/final_main_ncv.csv")
d %>% group_by(task_category) %>% summarize(count = n() / nrow(d)) %>%
  ggplot(aes(
    x = reorder(task_category,-count),
    y = count * 100,
    color = 'r',
    fill = 'r'
  )) +
  geom_bar(alpha = 0.4, size = 2, stat = "identity") +
  coord_flip() +
  scale_colour_manual(values = c(baseColor), name = "") +
  scale_fill_manual(values = c(bgColor), name = "") +
  theme_hc(base_size = baseSize) +
```

```r
  theme(strip.background = element_rect(fill = alpha(bgColor, 0.4))) +
  theme(legend.position = "None") +
  xlab("") +
  ylab("Share of total openings (%)") +
  scale_y_continuous(limits = c(0, 38)) +
  geom_text(
    hjust = -0.2,
    aes(label = paste(round(count * 100), "%", sep = "")),
    size = annot_size,
    color = baseColor
  ) +
  theme(plot.margin = unit(c(1, 0, 0, 0), "cm")) +
  theme(axis.title = element_text()) +
  facet_wrap( ~ "(A) Task categories", strip.position = "top") -> p1
p1
```



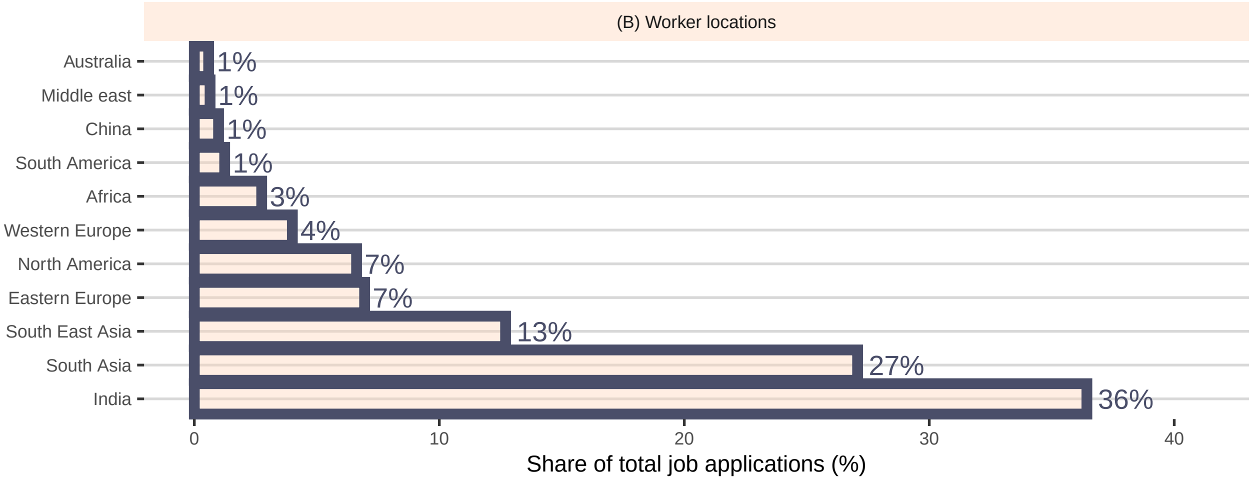
```r
#filtering hires from applications:
d %>% filter(totalCharge > 0) %>% select(totalCharge) %>%
  ggplot(aes(x = totalCharge, colour = 'r', fill = 'r')) +
  geom_density(adjust = 2, alpha = 0.4, size = 2) +
  scale_x_log10(breaks=c(1,10,100,1000)) +
  scale_fill_manual(values = c(bgColor)) +
  scale_color_manual(values = c(baseColor)) +
  theme_hc(base_size=baseSize)+
    theme(strip.background =element_rect(fill=alpha(bgColor,0.4)))+
  xlab("Payment per task ($, log-scale)") +
  ylab("Density") +
  theme(legend.position = "none") +
  theme(plot.margin = unit(c(1, 0, 0, 0), "cm"))+
  theme(axis.title = element_text())+
  facet_wrap(~"(D) Task cost",strip.position = "top")-> p2
```

```
p2
```



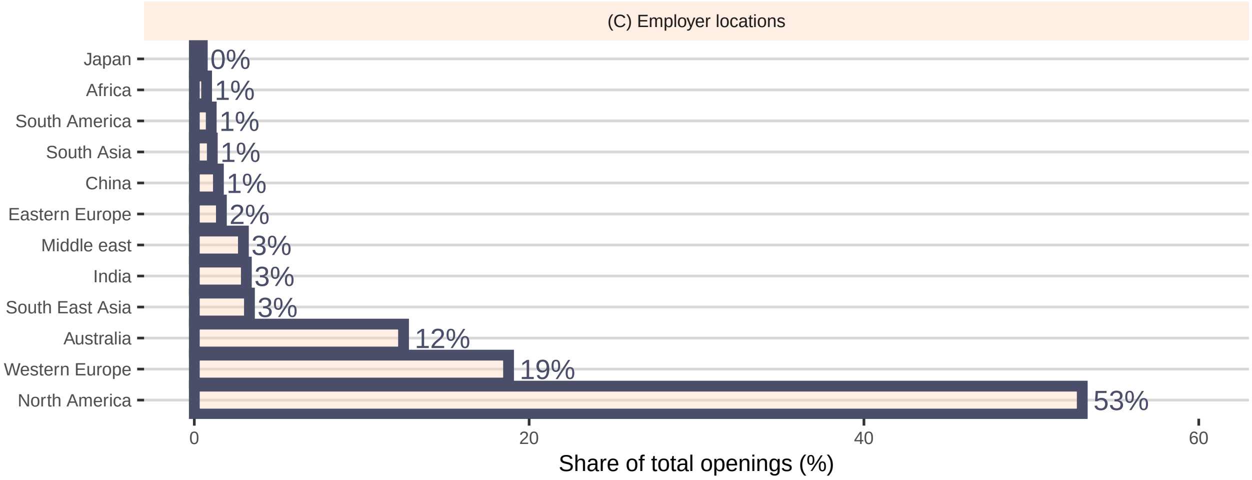
```
dr <- read.csv("../../../r0/data/dataRaw.csv")
SouthAsia <- c('Bangladesh', 'Pakistan', 'Sri Lanka', 'Nepal')
SouthEastAsia <-
  c('Philippines',
    'Indonesia',
    'Vietnam',
    'Malaysia',
    'Thailand',
    'Singapore')
NorthAm <- c('United States', 'Canada')
eastEurope <-
  c(
    'Ukraine',
    'Russia',
    'Armenia',
    'Croatia',
    'Belarus',
    'Bosnia and Herzegovina',
    'Serbia',
    'Bulgaria',
    'Poland',
    'Macedonia',
    'Romania'
  )
westEurope <- c(
  'United Kingdom',
  'Germany',
  'Spain',
  'France',
  'Italy',
  'Portugal',
  'Netherlands',
  'Greece',
  'Czech Republic',
  'Sweden',
  'Switzerland',
  'Norway',
  'Ireland',
  'Denmark',
  'Belgium'
)
Africa <-
  c('Kenya',
    'Egypt',
    'Tunisia',
    'Morocco',
    'Algeria',
    'South Africa',
    'Nigeria',
    'Kenya')
China <- c('China', 'Hong Kong', 'Taiwan')
Australia <- c('New Zealand', 'Australia')
southAmerica <-
  c('Jamaica',
    'Argentina',
    'Brazil',
    'Mexico',
    'Venezuela',
    'Colombia')
middleEast <-
  c(
    'Turkey',
    'United Arab Emirates',
    'Saudi Arabia',
    'Lebanon',
    'Jordan',
    'Israel',
    'Egypt'
  )

dr$contractorCurCountry <- as.character(dr$contractorCurCountry)
dr$contractorCurCountry <-
```

```r
  ifelse(dr$contractorCurCountry %in% SouthAsia,
         'South Asia',
         dr$contractorCurCountry)
dr$contractorCurCountry <-
  ifelse(
    dr$contractorCurCountry %in% SouthEastAsia,
    'South East Asia',
    dr$contractorCurCountry
  )
dr$contractorCurCountry <-
  ifelse(dr$contractorCurCountry %in% NorthAm,
         'North America',
         dr$contractorCurCountry)
dr$contractorCurCountry <-
  ifelse(dr$contractorCurCountry %in% eastEurope,
         'Eastern Europe',
         dr$contractorCurCountry)
dr$contractorCurCountry <-
  ifelse(dr$contractorCurCountry %in% westEurope,
         'Western Europe',
         dr$contractorCurCountry)
dr$contractorCurCountry <-
  ifelse(dr$contractorCurCountry %in% Africa,
         'Africa',
         dr$contractorCurCountry)
dr$contractorCurCountry <-
  ifelse(dr$contractorCurCountry %in% China,
         'China',
         dr$contractorCurCountry)
dr$contractorCurCountry <-
  ifelse(dr$contractorCurCountry %in% southAmerica,
         'South America',
         dr$contractorCurCountry)
dr$contractorCurCountry <-
  ifelse(dr$contractorCurCountry %in% middleEast,
         'Middle east',
         dr$contractorCurCountry)
dr$contractorCurCountry <-
  ifelse(dr$contractorCurCountry %in% Australia,
         'Australia',
         dr$contractorCurCountry)

dr %>% count(contractorCurCountry) %>% filter(n > 4000) %>% mutate(count = n /
                                                              sum(n)) %>%
  ggplot(aes(
    x = reorder(contractorCurCountry,-count),
    y = count * 100,
    color = 'r',
    fill = 'r'
  )) +
  geom_bar(alpha = 0.4, size = 2, stat = "identity") +
  coord_flip() +
  scale_colour_manual(values = c(baseColor), name = "") +
  scale_fill_manual(values = c(bgColor), name = "") +
  theme_hc(base_size = baseSize) +
  theme(strip.background = element_rect(fill = alpha(bgColor, 0.4))) +
  theme(legend.position = "None") +
  xlab("") +
  ylab("Share of total job applications (%)") +
  scale_y_continuous(limits = c(0, 41)) +
  geom_text(
    hjust = -0.2,
    aes(label = paste(round(count * 100), "%", sep = "")),
    size = 4,
    color = baseColor
  ) +
  theme(plot.margin = unit(c(1, 0, 0, 0), "cm")) +
  theme(axis.title = element_text()) +
  facet_wrap( ~ "(B) Worker locations", strip.position = "top") -> p3
p3
```



```
dr$clientCurCountry <- as.character(dr$clientCurCountry)
dr$clientCurCountry <-
  ifelse(dr$clientCurCountry %in% SouthAsia,
```

```r
         'South Asia',
         dr$clientCurCountry)
dr$clientCurCountry <-
  ifelse(dr$clientCurCountry %in% SouthEastAsia,
         'South East Asia',
         dr$clientCurCountry)
dr$clientCurCountry <-
  ifelse(dr$clientCurCountry %in% NorthAm,
         'North America',
         dr$clientCurCountry)
dr$clientCurCountry <-
  ifelse(dr$clientCurCountry %in% eastEurope,
         'Eastern Europe',
         dr$clientCurCountry)
dr$clientCurCountry <-
  ifelse(dr$clientCurCountry %in% westEurope,
         'Western Europe',
         dr$clientCurCountry)
dr$clientCurCountry <-
  ifelse(dr$clientCurCountry %in% Africa, 'Africa', dr$clientCurCountry)
dr$clientCurCountry <-
  ifelse(dr$clientCurCountry %in% China, 'China', dr$clientCurCountry)
dr$clientCurCountry <-
  ifelse(dr$clientCurCountry %in% southAmerica,
         'South America',
         dr$clientCurCountry)
dr$clientCurCountry <-
  ifelse(dr$clientCurCountry %in% middleEast,
         'Middle east',
         dr$clientCurCountry)
dr$clientCurCountry <-
  ifelse(dr$clientCurCountry %in% Australia,
         'Australia',
         dr$clientCurCountry)



dr %>% count(clientCurCountry) %>% filter(n > 4000) %>% mutate(count = n /
                                                                 sum(n)) %>%
  ggplot(aes(
    x = reorder(clientCurCountry,-count),
    y = count * 100,
    color = 'r',
    fill = 'r'
  )) +
  geom_bar(alpha = 0.4, size = 2, stat = "identity") +
  coord_flip() +
  scale_colour_manual(values = c(baseColor), name = "") +
  scale_fill_manual(values = c(bgColor), name = "") +
  theme_hc(base_size = baseSize) +
  theme(strip.background = element_rect(fill = alpha(bgColor, 0.4))) +
  theme(legend.position = "None") +
  xlab("") +
  ylab("Share of total openings (%)") +
  scale_y_continuous(limits = c(0, 60)) +
  geom_text(
    hjust = -0.2,
    aes(label = paste(round(count * 100), "%", sep = "")),
    size = 4,
    color = baseColor
  ) +
  theme(plot.margin = unit(c(1, 0, 0, 0), "cm")) +
  theme(axis.title = element_text()) +
  facet_wrap( ~ "(C) Employer locations", strip.position = "top") -> p4


p4
```

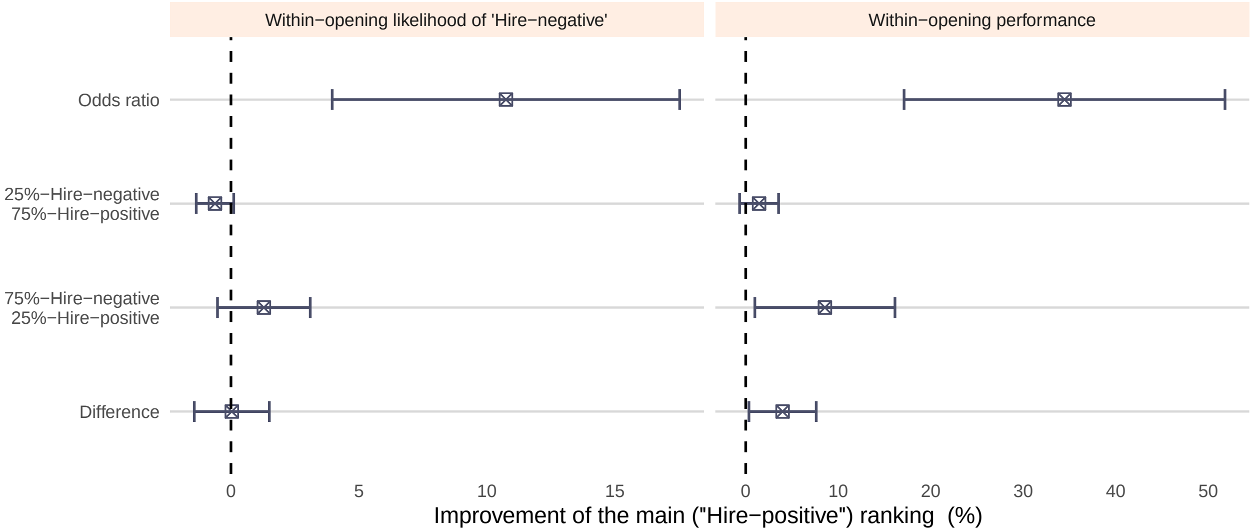


```
top_row <-
  plot_grid(
    p1,
    p3,
    p4,
    p2,
    label_size = 14,
    ncol = 2,
    rel_widths = c(1, 1)
  )
plot <- plot_grid(top_row, ncol = 1)
ggsave(
  file = paste(
    "../../../../../../../Apps/Overleaf/gbu/ms/r2/figures/market.pdf",
    sep = ""
  ),
  width = 18,
  height = 9,
  dpi = 300
)
```

12 Figure 18: Alternative rankings

```
d <- read.csv("../../data/evaluation_results/alt_ranks.csv")

baselines = unique(d$algorithm)
baselines = baselines[baselines != 'hmm']
d %>% select(fold, algorithm, score, metric) %>%  pivot_wider(names_from = algorithm,
                                                              values_from = score) ->
  d
get_focal_imprvs = function(algorithm) {
  new_d = get_imprvement(d, algorithm) %>% bind_cols(d %>% select(fold, metric))
  new_d %>% rename(improvement = algorithm) %>% mutate(model = algorithm) %>%
    group_by(model, metric) %>% summarise(avg = mean(improvement),
                                          se = serror(improvement)) -> new_d
  return(new_d)
}

baselines %>% map_dfr(get_focal_imprvs) -> r
r %>% update_model_names %>% ggplot(aes(
  x = model,
  y = avg,
  ymin = avg - 1.645 * se,
  ymax = avg + 1.645 * se,
  color = 'r',
  shape = 'r'
)) +
  geom_point(size = shape_size) +
```

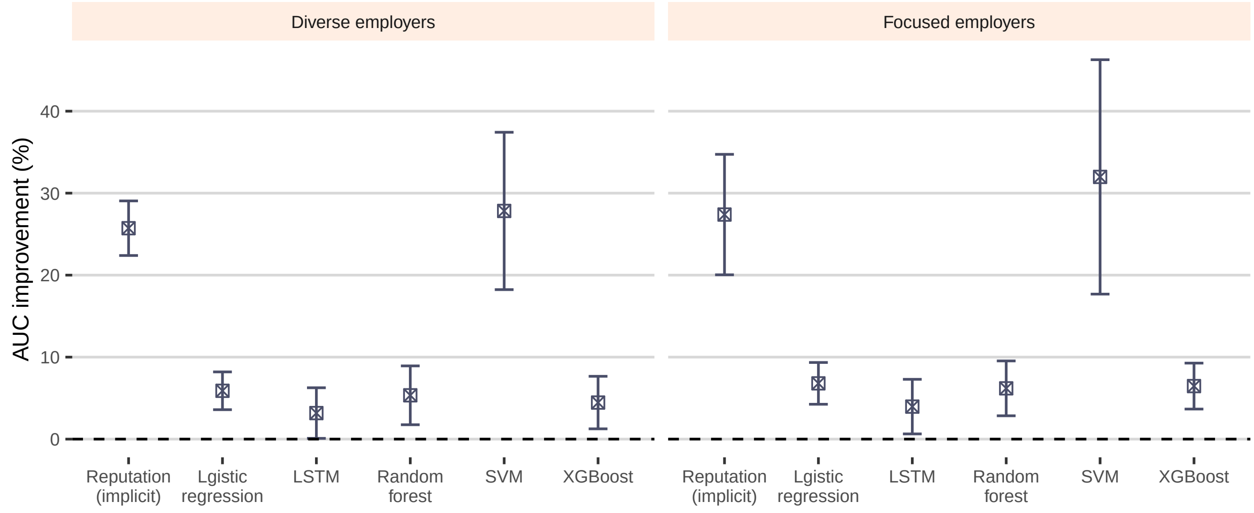
```r
  facet_wrap( ~ metric, scales = "free_x") +
  geom_errorbar(width = 0.2) +

  ylab("Improvement of the main (''Hire-positive'') ranking  (%)") +
  xlab(expression(paste(""))) +
  geom_vline(xintercept = c(10.5), linetype = "dashed") +
  theme_minimal(base_size = baseSize) +
  scale_color_manual(values = c(baseColor, baseColor, baseColor)) +
  scale_shape_manual(name = "",
                     values = c(baseShape, baseShape, baseShape)) +
  theme_hc(base_size = baseSize) +
  theme(strip.background = element_rect(fill = alpha(bgColor, 0.4))) +

  theme(legend.position = "none") +
  geom_abline(slope = 0,
              intercept = 0,
              linetype = 'dashed') +
  coord_flip()
```



```r
ggsave(
  file = paste(
    "../../../../../../../Apps/Overleaf/gbu/ms/r2/figures/altRankings.pdf",
    sep = ""
  ),
  width = 16,
  height = 5,
  dpi = 300
)
```

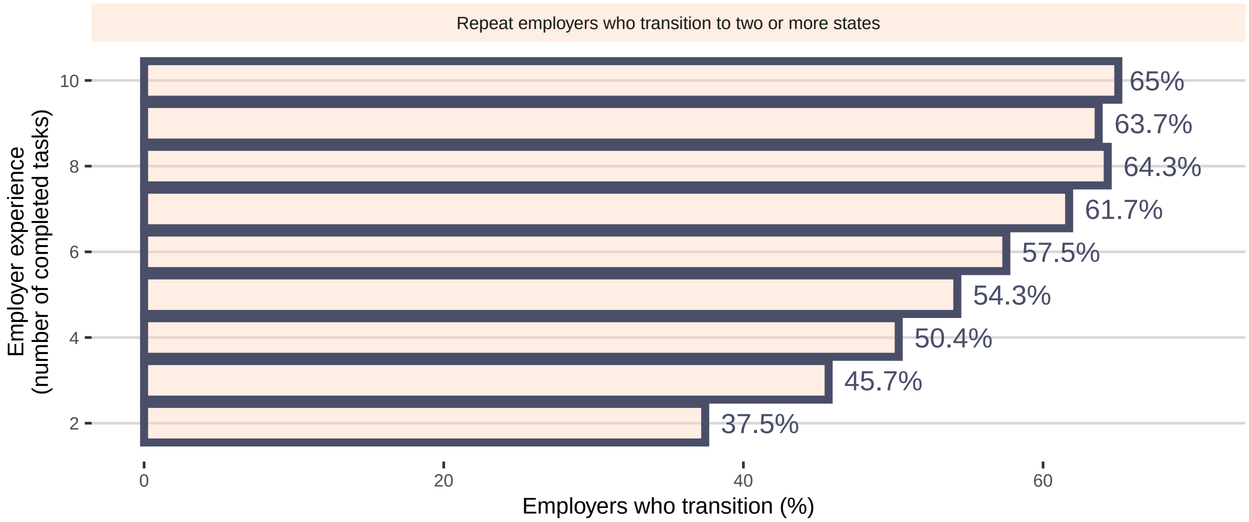
13 Figure 21: Diverse vs. focused employers

```
d = read_csv(paste(
  "../../data/evaluation_results/auc_n_heterogeneity.csv",
  sep = ""
))
get_focal_imprvs = function(algorithm) {
  new_d = get_imprvement(d, algorithm) %>% bind_cols(d %>% select(fold, cluster))
  new_d %>% rename(improvement = algorithm) %>% mutate(model = algorithm) %>%
    group_by(model, cluster) %>% summarise(avg = mean(improvement),
                                           se = serror(improvement)) -> new_d
  return(new_d)
}
baselines = unique(d$algorithm)
baselines = baselines[baselines != 'hmm']
d$cluster[d$cluster == "diverse"] = "Diverse employers"
d$cluster[d$cluster == "focused"] <- "Focused employers"
d %>% select(fold, algorithm, score, cluster) %>%  pivot_wider(names_from = algorithm,
                                                               values_from = score) ->
  d
baselines %>% map_dfr(get_focal_imprvs) -> r
r %>% update_model_names %>% ggplot(aes(
  x = model,
  y = avg,
  ymin = avg - 1.645 * se,
  ymax = avg + 1.645 * se,
  color = 'r',
  shape = 'r'
)) +
  facet_wrap( ~ cluster) +
  geom_point(size = shape_size) +
  geom_errorbar(width = 0.2) +
  ylab("AUC improvement (%)") +
  xlab(expression(paste(""))) +
  geom_vline(xintercept = c(10.5), linetype = "dashed") +
  theme_hc(base_size = baseSize) +
  theme(strip.background = element_rect(fill = alpha(bgColor, 0.4))) +
  scale_color_manual(values = c(baseColor, baseColor, baseColor)) +
  scale_shape_manual(name = "",
                     values = c(baseShape, baseShape, baseShape)) +
  theme(legend.position = "none") +
  geom_abline(slope = 0,
              intercept = 0,
              linetype = 'dashed') -> p


ggsave(
  file = paste(
    "../../../../../../../Apps/Overleaf/gbu/ms/r2/figures/heterogeneous.pdf",
    sep = ""
  ),
  width = 16,
  height = 5,
  dpi = 300
)
p
```



14 Figure 22: Transitions

```
d = read_csv("../../data/evaluation_results/transitions.csv")
d  %>%  ggplot(aes(
  x = completed_tasks,
  y = two_or_more_states * 100,
  color = 'r',
  fill = 'r'
)) + geom_col(alpha = 0.4, size = 1.5) +
  ylab("Employers who transition (%)") +
  xlab("Employer experience\n(number of completed tasks)") +
  theme_hc(base_size = baseSize) +
  theme(strip.background = element_rect(fill = alpha(bgColor, 0.4))) +
  scale_x_continuous(breaks = c(2, 4, 6, 8, 10)) +
  ylim(c(0, 70)) +
  coord_flip() +
  geom_text(
    hjust = -0.2,
    aes(label = paste(
      round(two_or_more_states * 100, 1), "%", sep = ""
    )),
    size = 4,
    color = baseColor
  ) +
  scale_color_manual(values = c(baseColor, baseColor, baseColor)) +
  scale_fill_manual(name = "",
                    values = c(bgColor, baseColor, baseColor)) +
  theme(legend.position = "none") +
  facet_wrap(~ "Repeat employers who transition to two or more states",
             strip.position = "top") +


  ggsave(
    file = "/Users/mkokkodi/Dropbox/Apps/Overleaf/gbu//ms/r2/figures/transitions.pdf",
    width = 10,
    height = 5,
    dpi = 300
  )
```



15 Table 3: Absolute performance tables

```
dataset = ""
d = read_csv(paste(
  "../../data/",
  dataset,
  "/evaluation_results/auc_n",
  dataset,
  ".csv",
  sep = ""
)) %>% filter(n == 0)
dataset = "70"
r1 = read_csv(paste(
  "../../data/",
  dataset,
  "/evaluation_results/auc_n",
  dataset,
  ".csv",
  sep = ""
)) %>% filter(n == 0)
dataset = "90"
r2 = read_csv(paste(
  "../../data/",
  dataset,
  "/evaluation_results/auc_n",
  dataset,
  ".csv",
  sep = ""
)) %>% filter(n == 0)
d %>% mutate(model = algorithm) %>% update_model_names -> d
r1 %>% mutate(model = algorithm) %>% update_model_names -> r1
r2 %>% mutate(model = algorithm) %>% update_model_names -> r2
models = unique(d$model)
d = as.data.frame(d)
r1 = as.data.frame(r1)
r2 = as.data.frame(r2)
d %>% group_by(model) %>% summarize(mean = mean(score)) -> d1
r1 %>% group_by(model) %>% summarize(mean = mean(score)) -> r12
r2 %>% group_by(model) %>% summarize(mean = mean(score)) -> r22
d1 = as.data.frame(d1)
r12 = as.data.frame(r12)
r22 = as.data.frame(r22)
cat("\\addlinespace\nAverage")
for (algo in models) {
  #algo = as.character(d1[i,'model'])
  cur_score = d1[d1$model == algo, 'mean']
  cur_score_70 = r12[r12$model == algo, 'mean']
  cur_score_90 = r22[r22$model == algo, 'mean']
  hl = ifelse(algo == 'HMM', "\\cellcolor{lightColor}", "")
  cat("&&",
      paste(
        str_replace_all(algo, "\n", " "),
        paste(hl, round(cur_score, 3)),
        round(cur_score_70, 3),
        round(cur_score_90, 3),
        sep = "&"
      ),
      "\\\\\n")
}

cat("\\cmidrule(r){3-5}\n\\addlinespace\nPer fold")

for (fold in 0:9) {
  for (algo in models) {
    cur_score = d[d$fold == fold & d$model == algo, 'score']
    cur_score_70 = r1[r1$fold == fold & r1$model == algo, 'score']
    cur_score_90 = r2[r2$fold == fold & r2$model == algo, 'score']
    hl = ifelse(algo == 'HMM', "\\cellcolor{lightColor}", "")
    cat("&",
        paste(
          fold,
          str_replace_all(algo, "\n", " "),
          paste(hl, round(cur_score, 3)),
          round(cur_score_70, 3),
          round(cur_score_90, 3),
          sep = "&"
        ),
        "\\\\\n")
  }
}
```

```
## \addlinespace
## Average&& HMM&\cellcolor{lightColor} 0.71&0.68&0.681 \\
## && Lgistic regression& 0.668&0.661&0.642 \\
## && Random forest& 0.673&0.664&0.643 \\
## && SVM& 0.561&0.548&0.543 \\
## && XGBoost& 0.677&0.66&0.657 \\
## && LSTM& 0.688&0.667&0.647 \\
## && Reputation (implicit)& 0.564&0.555&0.569 \\
## \cmidrule(r){3-5}
## \addlinespace
## Per fold& 0&HMM&\cellcolor{lightColor} 0.683&0.621&0.646 \\
## & 0&Lgistic regression& 0.584&0.576&0.5 \\
## & 0&Random forest& 0.579&0.553&0.562 \\
## & 0&SVM& 0.523&0.521&0.353 \\
## & 0&XGBoost& 0.603&0.595&0.587 \\
## & 0&LSTM& 0.578&0.608&0.563 \\
## & 0&Reputation (implicit)& 0.547&0.541&0.567 \\
## & 1&HMM&\cellcolor{lightColor} 0.704&0.689&0.649 \\
## & 1&Lgistic regression& 0.65&0.655&0.566 \\
## & 1&Random forest& 0.638&0.625&0.607 \\
## & 1&SVM& 0.463&0.556&0.499 \\
## & 1&XGBoost& 0.629&0.662&0.621 \\
## & 1&LSTM& 0.693&0.652&0.616 \\
## & 1&Reputation (implicit)& 0.6&0.585&0.591 \\
## & 2&HMM&\cellcolor{lightColor} 0.712&0.675&0.683 \\
## & 2&Lgistic regression& 0.67&0.68&0.683 \\
## & 2&Random forest& 0.656&0.667&0.624 \\
## & 2&SVM& 0.598&0.497&0.524 \\
## & 2&XGBoost& 0.684&0.663&0.633 \\
## & 2&LSTM& 0.698&0.69&0.494 \\
## & 2&Reputation (implicit)& 0.554&0.536&0.561 \\
## & 3&HMM&\cellcolor{lightColor} 0.732&0.719&0.719 \\
## & 3&Lgistic regression& 0.696&0.677&0.685 \\
## & 3&Random forest& 0.723&0.704&0.699 \\
## & 3&SVM& 0.63&0.542&0.566 \\
## & 3&XGBoost& 0.727&0.683&0.701 \\
## & 3&LSTM& 0.705&0.66&0.658 \\
## & 3&Reputation (implicit)& 0.603&0.603&0.624 \\
## & 4&HMM&\cellcolor{lightColor} 0.717&0.666&0.682 \\
## & 4&Lgistic regression& 0.676&0.655&0.664 \\
## & 4&Random forest& 0.688&0.653&0.644 \\
## & 4&SVM& 0.483&0.548&0.606 \\
## & 4&XGBoost& 0.68&0.649&0.64 \\
## & 4&LSTM& 0.698&0.631&0.688 \\
## & 4&Reputation (implicit)& 0.566&0.561&0.561 \\
## & 5&HMM&\cellcolor{lightColor} 0.696&0.669&0.679 \\
## & 5&Lgistic regression& 0.656&0.656&0.65 \\
## & 5&Random forest& 0.666&0.672&0.638 \\
## & 5&SVM& 0.557&0.557&0.56 \\
## & 5&XGBoost& 0.665&0.64&0.668 \\
## & 5&LSTM& 0.678&0.691&0.676 \\
## & 5&Reputation (implicit)& 0.557&0.54&0.553 \\
## & 6&HMM&\cellcolor{lightColor} 0.714&0.688&0.676 \\
## & 6&Lgistic regression& 0.692&0.663&0.628 \\
## & 6&Random forest& 0.654&0.649&0.594 \\
## & 6&SVM& 0.579&0.609&0.597 \\
## & 6&XGBoost& 0.678&0.656&0.673 \\
## & 6&LSTM& 0.699&0.682&0.642 \\
## & 6&Reputation (implicit)& 0.563&0.557&0.573 \\
## & 7&HMM&\cellcolor{lightColor} 0.708&0.687&0.673 \\
## & 7&Lgistic regression& 0.691&0.668&0.677 \\
## & 7&Random forest& 0.701&0.705&0.66 \\
## & 7&SVM& 0.582&0.561&0.618 \\
## & 7&XGBoost& 0.691&0.678&0.65 \\
## & 7&LSTM& 0.708&0.675&0.695 \\
## & 7&Reputation (implicit)& 0.594&0.561&0.598 \\
## & 8&HMM&\cellcolor{lightColor} 0.71&0.688&0.695 \\
## & 8&Lgistic regression& 0.682&0.682&0.682 \\
## & 8&Random forest& 0.722&0.714&0.691 \\
## & 8&SVM& 0.552&0.525&0.542 \\
## & 8&XGBoost& 0.705&0.699&0.703 \\
## & 8&LSTM& 0.693&0.672&0.696 \\
## & 8&Reputation (implicit)& 0.559&0.563&0.561 \\
## & 9&HMM&\cellcolor{lightColor} 0.726&0.698&0.711 \\
## & 9&Lgistic regression& 0.687&0.699&0.686 \\
## & 9&Random forest& 0.706&0.693&0.708 \\
## & 9&SVM& 0.644&0.565&0.566 \\
## & 9&XGBoost& 0.706&0.678&0.69 \\
## & 9&LSTM& 0.732&0.71&0.74 \\
## & 9&Reputation (implicit)& 0.498&0.5&0.499 \\
```