



Vilniaus Universitetas

Regresija įvykių skaičiui

Laboratorinis darbas

Darbą atliko:

Vainius Gataveckas, Matas Gaulia, Dovydas Martinkus

Duomenų Mokslas

3 kursas 2 gr.

Vilnius, 2022

Turinys

Naudoti metodai	3
Duomenys ir jų šaltiniai.....	4
Tikslas ir uždaviniai	5
Atliktos analizės aprašymas	6
1. Naudojant R	6
Prognozuojant testavimo aibėje esančias reikšmes gaunami minimaliai prastesni rezultatai lyginant su kvazi-Puasono modeliu (RMSE=92, MAE=73).....	20
2. Naudojant Python.....	22

Naudoti metodai

Darbas atliktas naudojant R ir Python.

Naudoti R paketai:

tidyverse

AER

IMASS

rsample

corrplot

effects

yardstick

VGAM

Naudoti Python paketai:

numpy

pandas

matplotlib

seaborn

statsmodels

Duomenys ir jų šaltiniai

Išnuomotų dviračių kiekio pagal dienos ir oro sąlygų duomenys.

Duomenų šaltinis – Kaggle. Prieiga per internetą:

<https://www.kaggle.com/brajeshmohapatra/bike-count-prediction-data-set?select=train.csv>.

“Datetime” – data ir laikas.

“Season” – metų laikas (1 – pavasaris, 2 – vasara, 3- rudenį, 4 - žiema).

“Holiday” – ar diena yra šventė.

“Workingday” – ar diena yra darbo.

“Weather” – koduotas oro sąlygų kintamasis.

“Temp” – temperatūra Celsijaus laipsniais.

“Atemp” – jutiminė temperatūra Celsijaus laipsniais.

“Humidity” – oro drėgnumas.

“Windspeed” – vėjo greitis.

“Casual” – neregistruotų vartotojų išsinuomotų dviračių kiekis.

“Register” - registruotų vartotojų išsinuomotų dviračių kiekis.

“Counts” – bendras išsinuomotų dviračių kiekis (atsako kintamasis).

Tikslas ir uždaviniai

Tikslas: Sudaryti regresijos modelį išnuomotų dviračių skaičiui, įvertinti kokią įtaką tam tikri požymiai daro dviračių nuomos paklausai, panaudoti sudarytą modelį prognozuoti dviračių paklausą esant tam tikroms sąlygoms.

Uždaviniai:

Sudaryti įvokių skaičiaus regresijos modelius turimai duomenų aibei.

Atlikti sudarytų modelių tinkamumo analizę.

Tinkamiausio modelio parinkimas.

Modelio koeficientų interpretacija.

Modelio panaudojimas prognozuoti dviračių nuomos paklausą esant tam tikroms sąlygoms.

Atliktos analizės aprašymas

1. Naudojant R

Duomenų aibę apskritai sudaro 17379 stebėjimai. Duomenų aibėje nėra praleistų reikšmių. Pasirinkta duomenų aibę sumažinti iki stebėjimų gautų vidurdienį (12-14 valandomis) ir sudaryti modelį dviračių nuomos paklausai tik esant tam dienos laikotarpiui. Duomenys padalinti į mokymo ir testavimo aibes naudojant 90-10 santykį.

```
# Duomenys
# https://www.kaggle.com/datasets/brajeshmohapatra/bike-count-prediction-data-set?select=train.csv

library(tidyverse)
library(AER)
library(MASS)

tr <- read.csv("train.csv")
te <- read.csv("test.csv")
te$count <- te$casual + te$registered
full <- rbind(tr, te)

full <- full %>%
  dplyr::select(-c(casual, registered)) %>%
  mutate(
    hour = lubridate::hour(datetime)
  )

full <- full %>%
  filter(hour %in% c(12,13,14)) %>%
  dplyr::select(-c(hour,datetime)) %>%
  mutate(
    season = factor(season),
    holiday = factor(holiday),
    workingday = factor(workingday),
    weather = factor(weather)
  )

head(full)

##   season holiday workingday weather  temp  atemp humidity windspeed count
## 1      1       0          0       1 17.22 21.210      77    19.0012     84
## 2      1       0          0       2 18.86 22.725      72    19.9995     94
## 3      1       0          0       2 18.86 22.725      72    19.0012    106
## 4      1       0          0       2 14.76 16.665      66    19.9995     93
## 5      1       0          0       2 14.76 17.425      66      8.9981     75
## 6      1       0          0       3 14.76 17.425      76    12.9980     59

# Perdaromi mokymo ir testavimo duomenų rinkiniai
library(rsample)
full_split <- initial_split(full, prop = 0.9)
train <- training(full_split)
test <- testing(full_split)

write_csv(train, "train_from_R.csv")
write_csv(test, "test_from_R.csv")
```

```

# Siekiant gauti tokius pat rezultatus duomenys nuskaitomi iš failų
train <- read_csv("train_from_R.csv")

test <- read_csv("test_from_R.csv")

train <- train %>%
  mutate(
    season = factor(season),
    holiday = factor(holiday),
    workingday = factor(workingday),
    weather = factor(weather)
  )

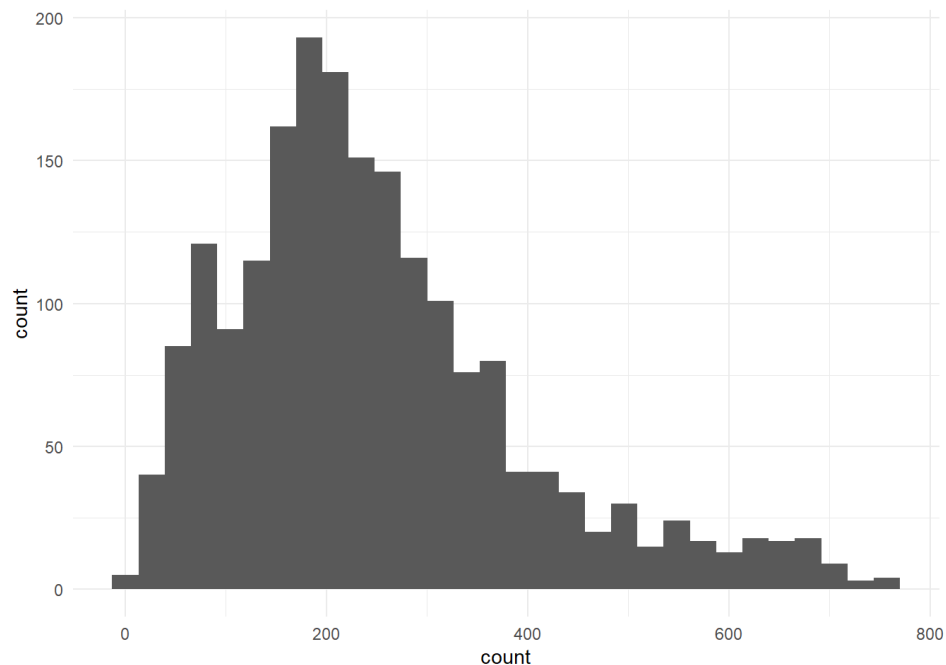
test <- test %>%
  mutate(
    season = factor(season),
    holiday = factor(holiday),
    workingday = factor(workingday),
    weather = factor(weather)
  )

min(test$count)

## [1] 12

ggplot(train, aes(x = count)) +
  geom_histogram() +
  theme_minimal()

```



```

# Dispersija didesnė už vidurkį
mean(train$count)

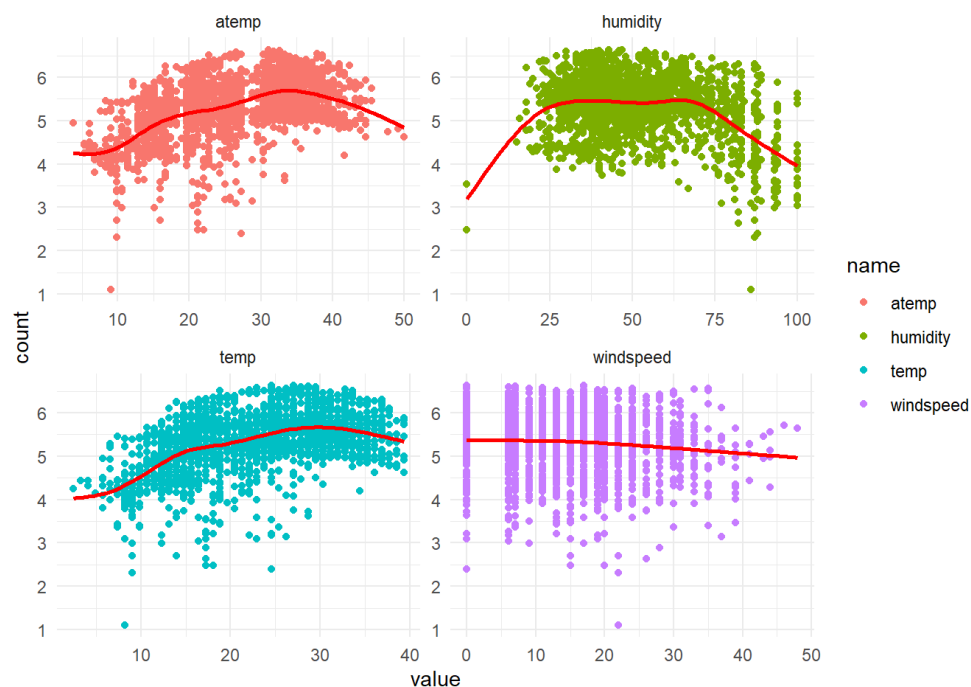
## [1] 251.3503

```

```
var(train$count)
## [1] 22014.63
```

Duomenų aibėje esanti mažiausia išnuomotų dviračių reikšmė yra lygi 12. Iš atsako kintamojo histogramos matoma, kad turimi duomenys su dešiniąja asimetriją. Apskaičiavus aprašomosios statistikos charakteristikas rasta, kad išnuomotų dviračių skaičiaus dispersija stipriai didesnė už vidurkį. Dėl šios priežasties daroma prielaida, kad Puasono regresijos modelis duomenims nebus tinkamas.

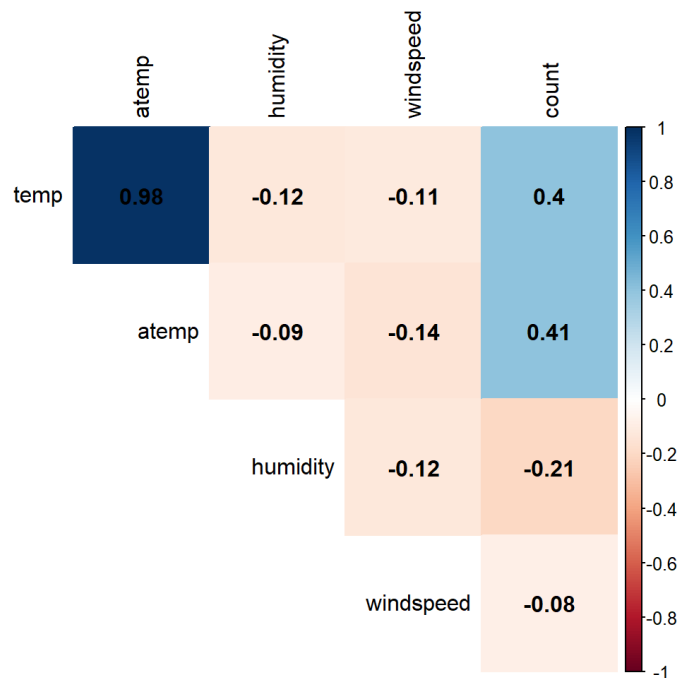
```
train %>%
  mutate(count = log(count)) %>%
  dplyr::select(c(temp, atemp, windspeed, humidity, count)) %>%
  pivot_longer(-count) %>%
  ggplot(aes(x = value, y = count, colour = name)) +
  geom_point() +
  geom_smooth(se = F, color = "red") +
  facet_wrap(~name, scales = "free") +
  theme_minimal()
```



```
# Tarpusavio koreliacijos
library(corrplot)

correlation_matrix <- train %>%
  dplyr::select(where(is.numeric)) %>%
  cor()

corrplot(correlation_matrix, order = "original", method = "color", type = "upper", diag =
FALSE, tl.col = "black", addCoef.col = "black")
```

Rasti pakankamai tiesiniai ryšiai tarp skaitinių požymių ir išnuomotų dviračių skaičiaus logaritmo. Apskaičiavus koreliacijas tarp skaitinių duomenų aibės požymių rasta beveik visiškai tiesinis ryšys tarp oro temperatūros ir jutiminės oro temperatūros ($r=0.98$). Dėl šios priežasties pasirinkta sudarant modelius kaip kovariantę įtraukti tik jutiminę oro temperatūrą.

```
train <- train %>% dplyr::select(-c(temp))

name <- full %>%
  dplyr::select(where(is.factor)) %>%
  names()

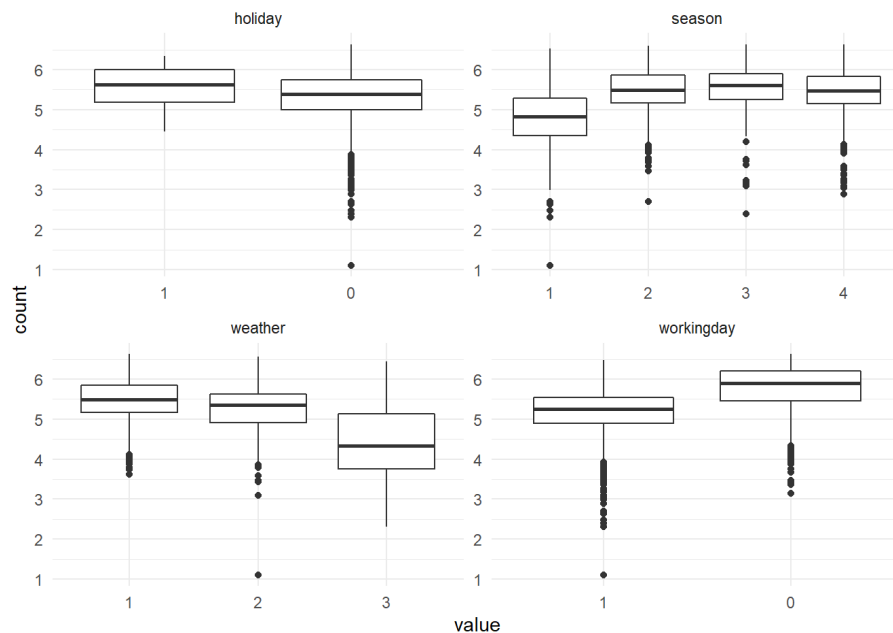
group <- function(x) {
  full %>%
    group_by(!sym(x)) %>%
    summarize(mean = mean(count), var = var(count), n = n())
}

purrr::map(name, group)

## [[1]]
## # A tibble: 4 x 4
##   season mean    var    n
##   <fct> <dbl> <dbl> <int>
## 1 1      152.  9830.  540
## 2 2      276. 22899.  552
## 3 3      293. 19199.  564
## 4 4      275. 21671.  530
##
## [[2]]
## # A tibble: 2 x 4
##   holiday mean    var    n
```

```
## <fct> <dbl> <dbl> <int>
## 1 0      248. 21573. 2123
## 2 1      291. 20152. 63
##
## [[3]]
## # A tibble: 2 x 4
##   workingday mean    var    n
##   <fct>      <dbl> <dbl> <int>
## 1 0          368. 30576. 693
## 2 1          194. 7851. 1493
##
## [[4]]
## # A tibble: 3 x 4
##   weather mean    var    n
##   <fct>      <dbl> <dbl> <int>
## 1 1          278. 22609. 1380
## 2 2          221. 14266. 632
## 3 3          125. 14994. 174

train %>%
  mutate(count = log(count)) %>%
  dplyr::select(c(season, holiday, workingday, weather, count)) %>%
  pivot_longer(-count) %>%
  ggplot(aes(x = value, y = count, group = value)) +
  geom_boxplot() +
  facet_wrap(~name, scales = "free") +
  theme_minimal()
```



Apskaičiuotas išnuomotų dviračių skaičiaus vidurkis esant skirtingiems kategorinių kintamųjų lygmenims, pasiskirstymas pavaizduotas stačiakampėmis diagramomis.

```
# Puasono modelis
model_1 <- glm(count ~ ., family = "poisson", data = train)
summary(model_1)

##
## Call:
```

```

## glm(formula = count ~ ., family = "poisson", data = train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -19.329   -4.418   -0.585    3.526   21.294
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.4093591  0.0085698  631.21 < 2e-16 ***
## season2      0.3525854  0.0056173   62.77 < 2e-16 ***
## season3      0.2529494  0.0067368   37.55 < 2e-16 ***
## season4      0.4679862  0.0049358   94.81 < 2e-16 ***
## holiday1     -0.2959043  0.0079517  -37.21 < 2e-16 ***
## workingday1  -0.6773281  0.0029401 -230.38 < 2e-16 ***
## weather2     -0.0152219  0.0037963   -4.01 6.08e-05 ***
## weather3     -0.4720235  0.0084025  -56.18 < 2e-16 ***
## atemp        0.0213013  0.0002731   78.00 < 2e-16 ***
## humidity     -0.0050001  0.0001071  -46.67 < 2e-16 ***
## windspeed    -0.0031659  0.0001686  -18.78 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 164971  on 1966  degrees of freedom
## Residual deviance:  61911  on 1956  degrees of freedom
## AIC: 76041
##
## Number of Fisher Scoring iterations: 5

cat("Deviacija padalinta iš laisvės laipsnių: ", model_1$deviance / model_1$df.residual, "\n")

## Deviacija padalinta iš laisvės laipsnių:  31.6518

cat("Siekama, kad būtų tarp 0.7 ir 1.3")

## Siekiama, kad būtų tarp 0.7 ir 1.3

# Tikrinama hipotezė, kad modelis nėra per didelės dispersijos (tiesinės dispersijos funkcijos
alternatyva)
dispersiontest(model_1, trafo = 1)

##
## Overdispersion test
##
## data:  model_1
## z = 26.659, p-value < 2.2e-16
## alternative hypothesis: true alpha is greater than 0
## sample estimates:
##      alpha
## 30.50793

```

Sudarytas Puasono regresijos modelis, naudojantis visas duomenyse esančias kovariantes, ir įvertintas pasitelkiant nykščio taisyklę, teigiančią, kad deviacija, padalinta iš jos laisvės laipsnių turi priklausyti intervalui [0.7,1.3]. Gauta reikšmė nepatenka į šį intervalą.

Hipotezė, kad modelis nėra per didelės dispersijos esant tiesinės dispersijos funkcijos alternatyvai atmesta. Todėl sudarytas kvazi-Puasono modelis.

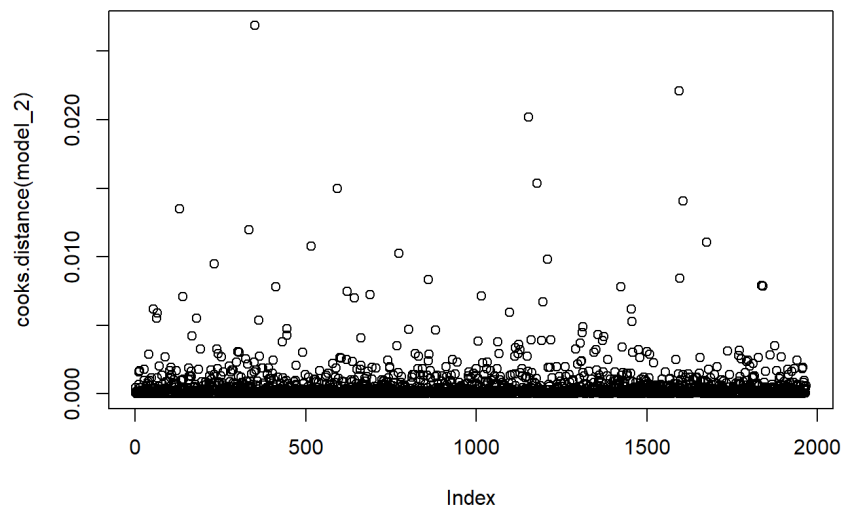
```

model_2 <- glm(count ~ ., family = quasipoisson(), data = train)
summary(model_2)

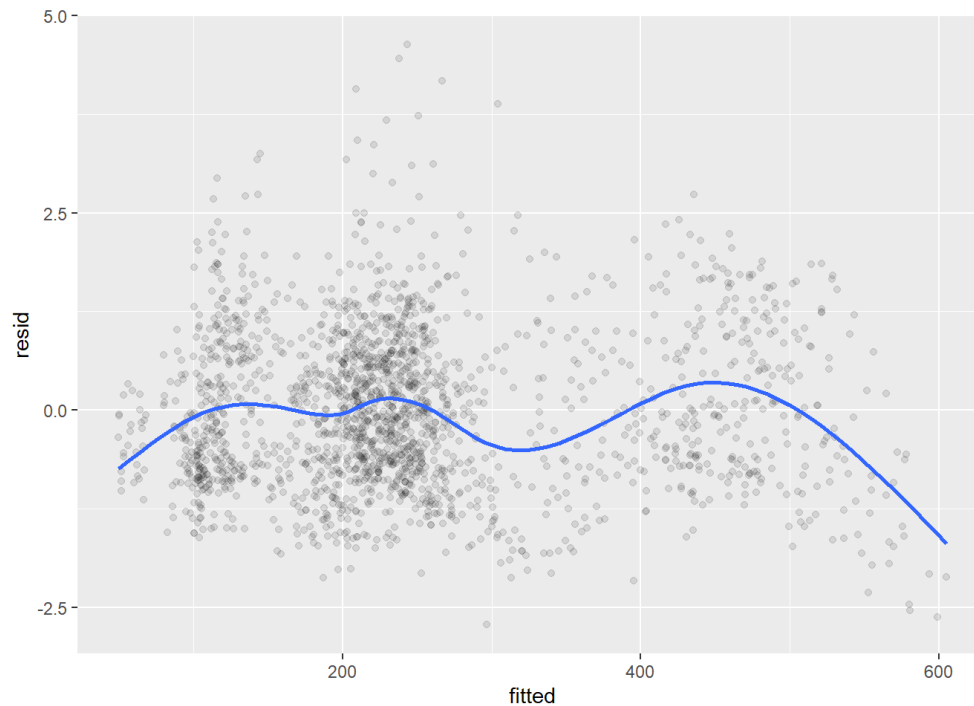
##
## Call:
## glm(formula = count ~ ., family = quasipoisson(), data = train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -19.329   -4.418   -0.585    3.526   21.294
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.4093591  0.0482365 112.142 < 2e-16 ***
## season2      0.3525854  0.0316178  11.151 < 2e-16 ***
## season3      0.2529494  0.0379188   6.671 3.3e-11 ***
## season4      0.4679862  0.0277817  16.845 < 2e-16 ***
## holiday1     -0.2959043  0.0447570  -6.611 4.9e-11 ***
## workingday1  -0.6773281  0.0165488 -40.929 < 2e-16 ***
## weather2     -0.0152219  0.0213681  -0.712 0.476325
## weather3     -0.4720235  0.0472948  -9.980 < 2e-16 ***
## atemp        0.0213013  0.0015372  13.857 < 2e-16 ***
## humidity     -0.0050001  0.0006030  -8.292 < 2e-16 ***
## windspeed    -0.0031659  0.0009489  -3.336 0.000865 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 31.68144)
##
##      Null deviance: 164971  on 1966  degrees of freedom
## Residual deviance:  61911  on 1956  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5

plot(cooks.distance(model_2))

```



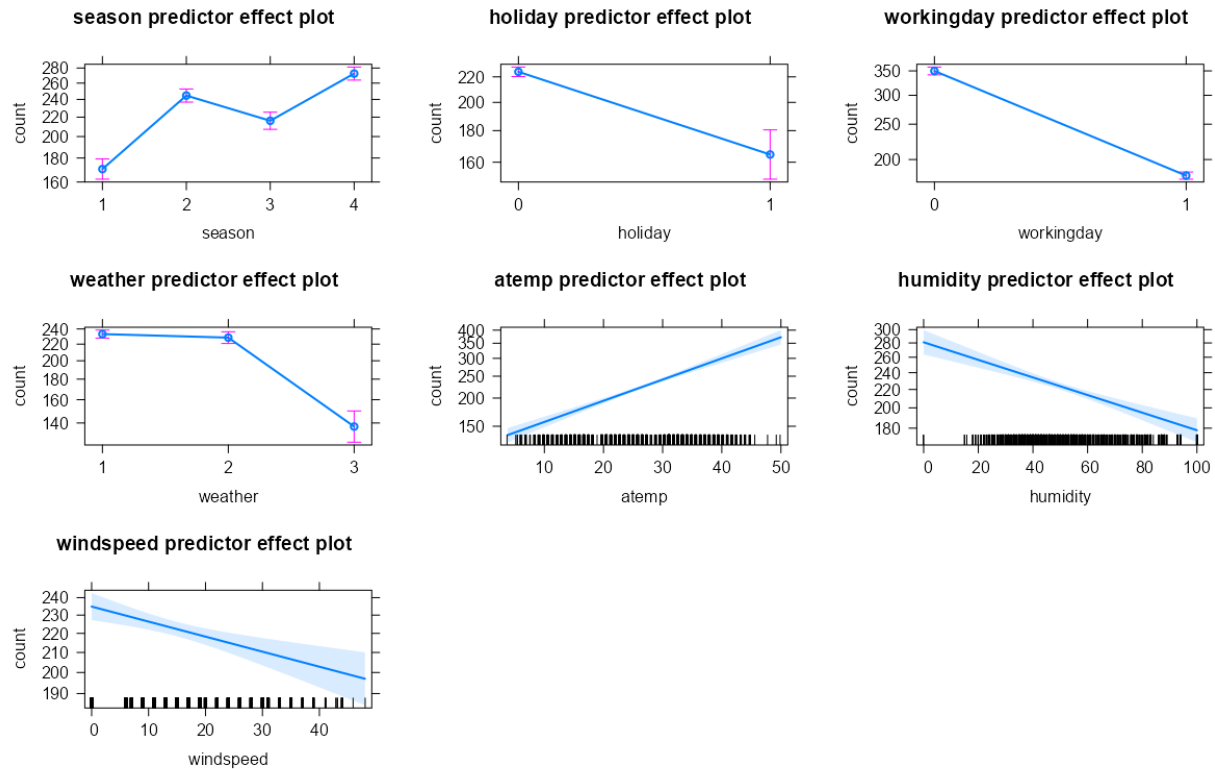
```
tibble(fitted = model_2$fitted.values, resid = resid(model_2, "pearson") / sqrt(31.8)) %>%
  ggplot(aes(fitted, resid)) +
  geom_point(alpha = 0.1) +
  geom_smooth(se = F)
```



```
# Modelio koeficientų reikšmės
est <- cbind(Estimate = exp(coef(model_2)), exp(confint(model_2)))
est
```

##	Estimate	2.5 %	97.5 %
## (Intercept)	223.4883055	203.2891364	245.6024019
## season2	1.4227412	1.3373440	1.5138075
## season3	1.2878182	1.1956824	1.3872969
## season4	1.5967753	1.5122960	1.6862980
## holiday1	0.7438586	0.6806378	0.8111957
## workingday1	0.5079724	0.4917668	0.5247259
## weather2	0.9848934	0.9444102	1.0269243
## weather3	0.6237388	0.5680586	0.6837840
## atemp	1.0215298	1.0184580	1.0246135
## humidity	0.9950124	0.9938362	0.9961880
## windspeed	0.9968391	0.9949859	0.9986938

```
library(effects)
plot(predictorEffects(model_2))
```



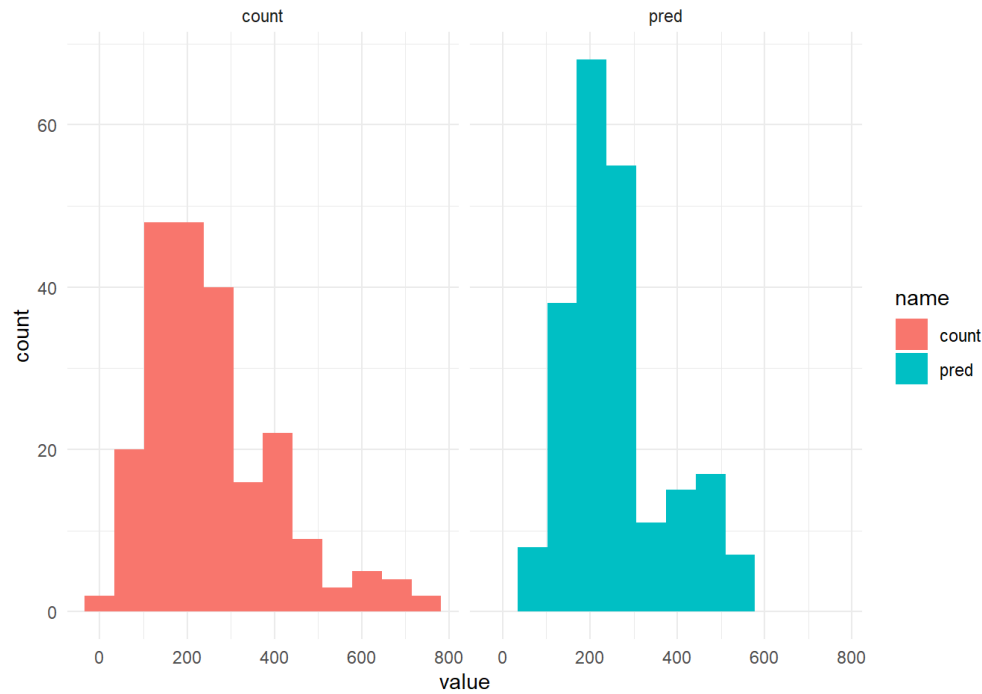
Gautame modelyje visos kovariantės reikšmingos.

Modelio koeficientų interpretacija įprasta modeliams, naudojantiems logaritminę jungties funkciją: koeficientų reikšmės atitinka kiekybinį atsako vidurkio logaritmo pokytį tą koeficientą atitinkančiai kovariantei pakitus vienetu, o likusioms kovariantėms esant fiksuotoms. Eksponencijuojant šiuos koeficientus gaunama kiek kartų padidėja atsako vidurkis.

Modelio gauti koeficientai panaudoti interpretuoti duomenyse esančių kovariačių ir atsako ryšį: Šventinėmis dienomis išnuomojama 26% mažiau dviračių negu įprastomis. Darbo dienomis išnuomojamų dviračių skaičius 50% procentų mažesnis negu nedarbo dienomis. Didėjant temperatūrai ir mažėjant oro drėgnumui išnuomojama daugiau dviračių. Didesnis vėjo greitis neigiamai veikia išnuomojamų dviračių skaičių.

```
# Modelio panaudojimas prognozėms naudojant testavimo duomenų aibę
test_with_pred <- test %>% mutate(count = count, pred = predict(model_2, test, type =
"response"))

test_with_pred %>%
  dplyr::select(c(count, pred)) %>%
  pivot_longer(everything()) %>%
  ggplot(aes(x = value, fill = name)) +
  geom_histogram(bins = 12) +
  theme_minimal() +
  facet_wrap(vars(name))
```



```
library(yardstick)

rmse(test_with_pred, count, pred)

## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>      <dbl>
## 1 rmse    standard      91.0

mae(test_with_pred, count, pred)

## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>      <dbl>
## 1 mae     standard      73.2
```

Modelis panaudotas atlikti prognozes naudojant testavimo aibės duomenis (RMSE=91, MAE=73). Iš prognozuotų ir stebėtų reikšmių histogramos pastebima, kad gautas modelis nelinkęs prognozuoti ekstremalias reikšmes, esančias testavimo aibėje. Laikant, kad būtent tokių reikšmių prognozavimas gali būti svarbiausias, modelio negalima laikyti tinkamu prognozuoti dviračių nuomos paklausą.

Hipotezė, kad modelis nėra per didelės dispersijos esant kvadratinės dispersijos funkcijos alternatyvai nebuvo atmesta, tačiau kaip alternatyvą pasirinkta sudaryti neigiamo binominio skirstinio regresijos modelį.

```

# Palyginimui sudaromas neigiamas binominis modelis (kvadratinė dispersijos funkcija)
dispersiontest(model_1, trafo = 2)

##
## Overdispersion test
##
## data: model_1
## z = 25.533, p-value < 2.2e-16
## alternative hypothesis: true alpha is greater than 0
## sample estimates:
##      alpha
## 0.1064244

model_3 <- glm.nb(count ~ ., data = train)
summary(model_3)

##
## Call:
## glm.nb(formula = count ~ ., data = train, init.theta = 6.754799939,
##      link = log)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -4.7468  -0.7857  -0.1109   0.5737   3.1238
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.3435252  0.0509210 104.938 < 2e-16 ***
## season2      0.3113537  0.0315978   9.854 < 2e-16 ***
## season3      0.2065184  0.0391877   5.270 1.36e-07 ***
## season4      0.4420221  0.0277014  15.957 < 2e-16 ***
## holiday1     -0.2916371  0.0534122  -5.460 4.76e-08 ***
## workingday1  -0.6702809  0.0196329 -34.141 < 2e-16 ***
## weather2      0.0001540  0.0228405   0.007  0.99462
## weather3     -0.5215176  0.0420013 -12.417 < 2e-16 ***
## atemp         0.0259243  0.0016459  15.751 < 2e-16 ***
## humidity     -0.0057120  0.0006397  -8.929 < 2e-16 ***
## windspeed    -0.0031533  0.0010323  -3.055  0.00225 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(6.7548) family taken to be 1)
##
##      Null deviance: 4933.9  on 1966  degrees of freedom
## Residual deviance: 2038.1  on 1956  degrees of freedom
## AIC: 23032
##
## Number of Fisher Scoring iterations: 1
##
##
##              Theta: 6.755
##            Std. Err.: 0.221
##
## 2 x log-likelihood: -23007.668

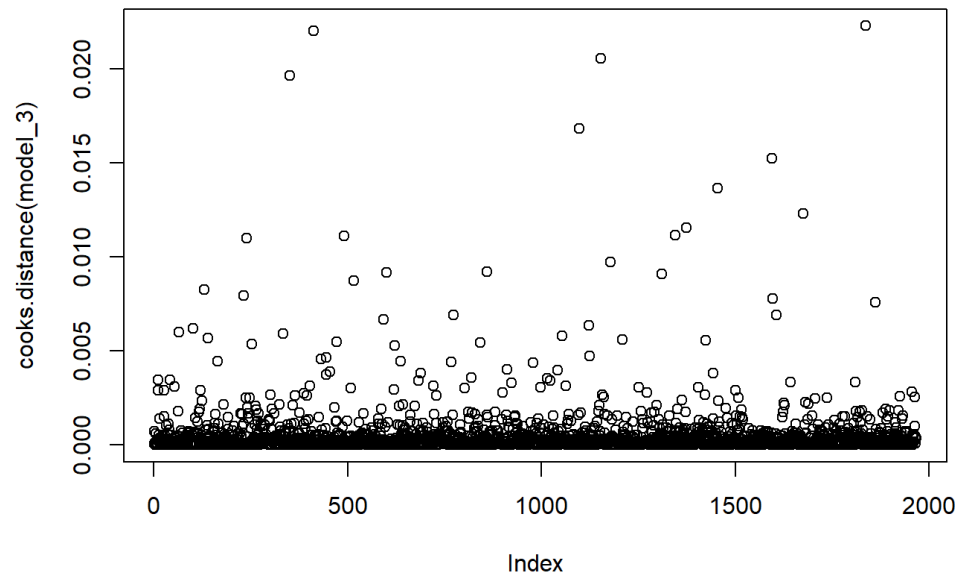
cat("Deviacija padalinta iš laisvės laipsnių: ", model_3$deviance / model_3$df.residual, "\n")

## Deviacija padalinta iš laisvės laipsnių: 1.041966

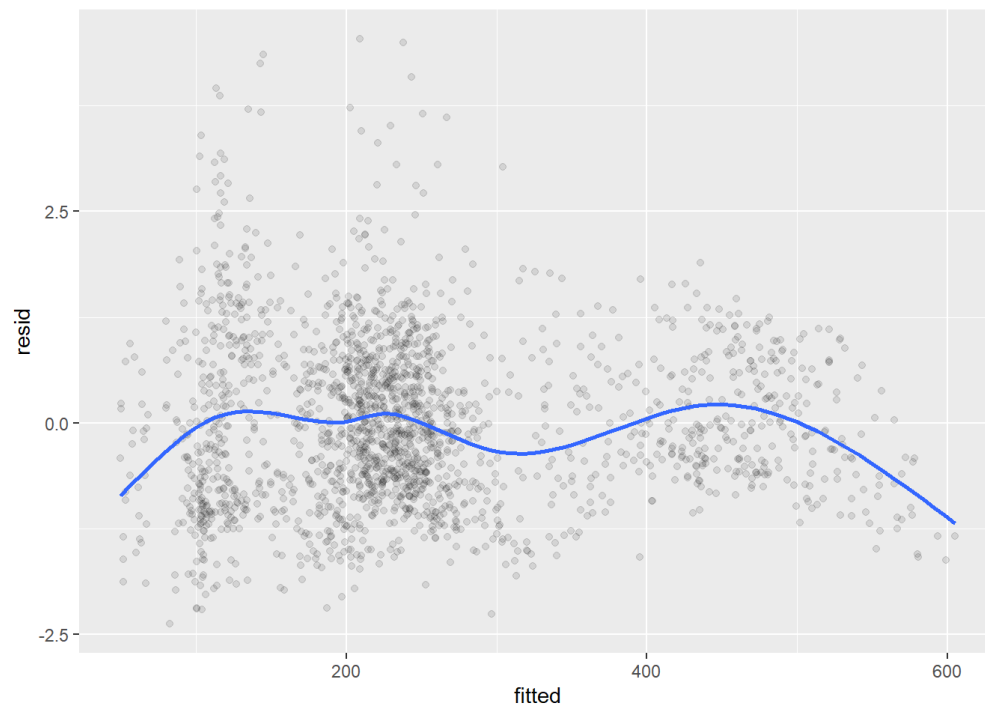
```



```
plot(cooks.distance(model_3))
```



```
tibble(fitted = model_2$fitted.values, resid = resid(model_3, "pearson")) %>%  
  ggplot(aes(fitted, resid)) +  
  geom_point(alpha = 0.1) +  
  geom_smooth(se = F)
```



```

# Galima pažingsninė regresija
model_3_step <- stepAIC(model_3)

## Start: AIC=23029.67
## count ~ season + holiday + workingday + weather + atemp + humidity +
##      windspeed
##
##           Df   AIC
## <none>      23030
## - windspeed  1 23037
## - holiday    1 23055
## - humidity   1 23102
## - weather    2 23194
## - atemp      1 23255
## - season     3 23309
## - workingday 1 24000

# Gaunamas lygiai toks pat modelis
anova(model_3, model_3_step)

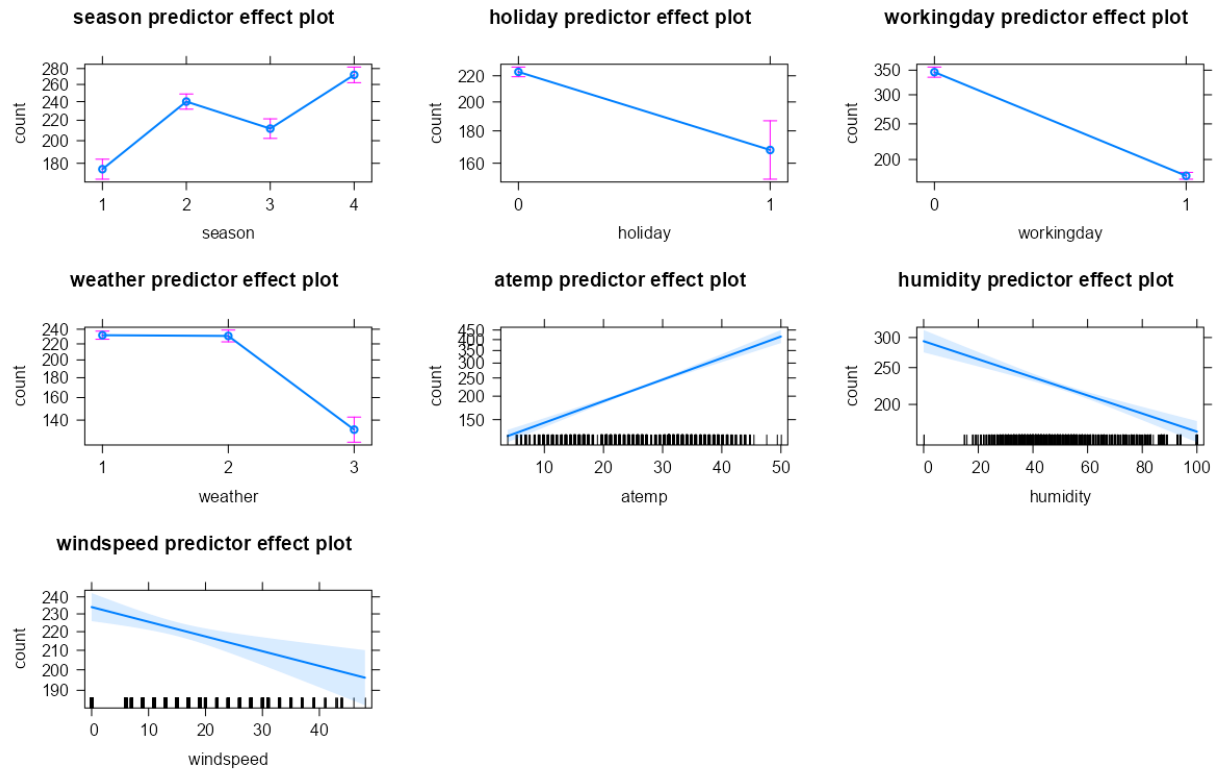
## Likelihood ratio tests of Negative Binomial Models
##
## Response: count
##                                     Model  theta
## 1 season + holiday + workingday + weather + atemp + humidity + windspeed 6.7548
## 2 season + holiday + workingday + weather + atemp + humidity + windspeed 6.7548
##   Resid. df    2 x log-lik.   Test    df LR stat. Pr(Chi)
## 1      1956      -23007.67
## 2      1956      -23007.67 1 vs 2     0        0        1

est <- cbind(Estimate = exp(coef(model_3)), exp(confint(model_3)))
est

##           Estimate      2.5 %      97.5 %
## (Intercept) 209.2490618 188.7297456 232.0759517
## season2     1.3652720   1.2849954   1.4505574
## season3     1.2293903   1.1418605   1.3235689
## season4     1.5558501   1.4749961   1.6411527
## holiday1    0.7470396   0.6737462   0.8306305
## workingday1 0.5115648   0.4922404   0.5315539
## weather2    1.0001540   0.9566608   1.0457851
## weather3    0.5936190   0.5478114   0.6437314
## atemp       1.0262633   1.0229109   1.0296238
## humidity    0.9943043   0.9930314   0.9955812
## windspeed   0.9968517   0.9948164   0.9988934

plot(predictorEffects(model_3))

```

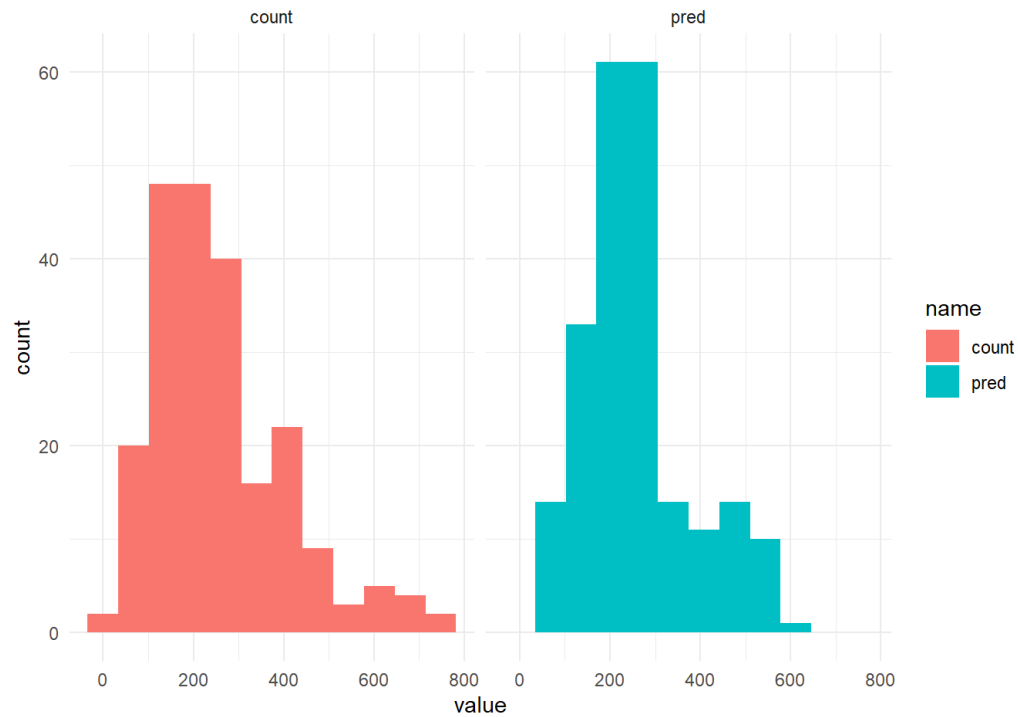


Priešingai negu kvazi-Puasono modelis, neigiamo binominio modelis leidžia pasitelkti su tikėtinumu susijusius kriterijus. Pasitelkiant anksčiau aprašytą nykščio taisyklę gauta reikšmė – 1.04. Naudojant pažingsninę regresiją pagal AIC iš modelio nepašalinta jokia kovariantė.

Gauti modelio koeficientai minimaliai skiriasi nuo prieš tai gauto kvazi-Puasono modelio.

```
test_with_pred <- test %>% mutate(count = count, pred = predict(model_3, test, type =
"response"))

test_with_pred %>%
  dplyr::select(c(count, pred)) %>%
  pivot_longer(everything()) %>%
  ggplot(aes(x = value, fill = name)) +
  geom_histogram(bins = 12) +
  theme_minimal() +
  facet_wrap(vars(name))
```



```
library(yardstick)

rmse(test_with_pred, count, pred)

## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>      <dbl>
## 1 rmse    standard      92.3

mae(test_with_pred, count, pred)

## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>      <dbl>
## 1 mae     standard      73.4
```

Prognozuojant testavimo aibėje esančias reikšmes gaunami minimaliai prastesni rezultatai lyginant su kvazi-Puasono modeliu (RMSE=92, MAE=73).

Išvados:

Duomenų aibė sumažinta iki stebėjimų gautų nuo 12 iki 14 valandos ir pasirinkta sudaryti regresijos modelį išnuomotų dviračių skaičiui tik esant šioms valandoms.

Laikyta, kad Puasono regresijos modelis yra netinkamas dėl per didelės atsako dispersijos palyginus su vidurkiu. Hipotezė, kad modelis nėra per didelės dispersijos esant tiesinės dispersijos funkcijos alternatyvai atmesta, todėl sudarytas kvazi-Puasono modelis naudojantis visas duomenyse esančias kovariantės. Sudarytame modelyje nereikšmingų kovariančių nerasta.

Modelis panaudotas siekiant interpretuoti duomenyse esančių kovariančių ir atsako ryšį: Šventinėmis dienomis ir darbo dienomis išnuomojama atitinkamai 26% ir 50% mažiau dviračių. Didesnė oro temperatūra teigiamai įtakoja dviračių nuomos paklausą, oro drėgnumas ir vėjo greitis – neigiamai.

Naudojant modelį prognozuoti išnuomotų dviračių kiekį testavimo aibėje rasta, kad modelis prastai prognozuoja ekstremalias išnuomotų dviračių skaičiaus reikšmes. Laikant, kad būtent tokių reikšmių prognozavimas gali būti svarbiausias, modelio negalima laikyti tinkamu prognozuoti dviračių nuomos paklausą.

Palyginimui sudarytas ir neigiamas binominis modelis išnuomotų dviračių skaičiui. Gautame modelyje visos duomenyse esančios kovariantės taip pat reikšmingos. Modelis panaudotas prognozuoti išnuomotų dviračių kiekį testavimo aibėje tačiau geresni rezultatai negauti.

2. Naudojant Python

Atlikta analizė pakartotinai atlikta naudojant Python.