PENINGKATAN PERFORMA PENDETEKSIAN GPS FAKE DRIVER GO-JEK MENGGUNAKAN METODE ENSEMBLE LEARNING

Rahmat Hartono
Email: rahmathartono261@gmail.com
Prodi Sistem Informasi
Universitas Pamulang
Jl. Raya Puspitek No.46, Buaran, Serpong, Tangerang Selatan, Banten

ABSTRAK

Fake GPS membawa dampak negatif pada stabilitas sistem dan keadilan di antara pengguna layanan lainnya dan terjadinya pemalsuan GPS sendiri karena Sifatnya terbuka, dari struktur sinyal GPS membuatnya rentan terhadap spoofing (pemalsuan) GPS, yang dapat dilakukan secara terang-terangan atau terselubung menjadi masalah yang sangat berpengaruh terhadap aktifitas bisnis bahkan memberi dampak yang sangat merugikan bagi Gojek, Dataset yang digunakan adalah data aktivitas Go-Ride dan GO-Food. Dalam penelitian ini berbasis esembel learning menggunakan Algoritma Random Forest, adaboost dan XGBoost. Ansambel learning mampu meningkatkan performa pendeteksian FAKE GPS dengan sangat baik.

Kata Kunci : Fake GPS, Driver, Random Forest, Adaboost , XGBoost dan deteksi.

PENDAHULUAN

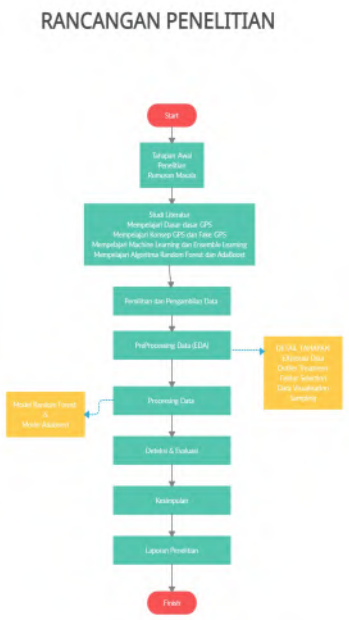
GO-JEK adalah startup pertama asal Indonesia yang diklasifikasikan sebagai perusahaan sistem transportasi. Driver merupakan tulang punggung bagi keluarga Go-Jek, di mana Driver bekerja dengan penuh semangat membantu jutaan warga Indonesia dalam aktivitas sehari-hari. Karena itu, perusahaan selalu berusaha untuk mendengarkan aspirasi Driver Go-Jek untuk terus memberikan kenyamanan dalam bekerja. Salah satu masalah yang sering dikeluhkan Driver adalah maraknya penggunaan aplikasi fake GPS. Penggunaan fake GPS ini tidak adil bagi pengemudi GO-JEK lain yang bekerja dengan jujur, ini menjadi salah satu kerugian yang bisa dialami jika mitra driver menggunakan fake GPS, dan sangat merugikan dari sudut pandang end user baik dari segi kenyamanan dan waktu Karenanya, peneliti ingin menerapkan model pembelajaran mesin untuk mengklasifikasikan apakah perjalanan dilakukan menggunakan fake GPS atau tidak berdasarkan perilaku PING mereka.

METODE

Berdasarkan hal tersebut penelitian ini bermaksud untuk melakukan penerapan metode Ensemble learning penelitian ini menggunaka algoritma Random Forest, Adaboost dan XGBoost. diharapkan mampu memberikan peningkatan performa pendeteksian terhadap fake GPS dengan pencapaian yang lebih baik dan mampu mendukung tingkat akurasi, stabilitas dan menangani underfit atau overfit problem dari penggunaan fake GPS an oleh driver. Dalam penelitian ini menggunakan tools data scientis yaitu Rstudio memiliki Versi gratis, RStudio membuat bahasa R jauh lebih mudah menggunakan dan memfasilitasi pengembangan skrip R lanjutan. RStudio menyertakan editor kode dengan alat debugging dan visualisasi canggih (Krotov, 2017).

HASIL DAN PEMBAHASAN

Berdasarkan studi literatur yang dilakukan dalam penelitian ini, maka penulis melakukan penyusunan tahap-tahap penelitian dalam bentuk kerangka pemikiran sebagai berikut :



Gambar Skema Penelitian

Pada gambar diatas, penulis mencoba merumuskan bagaimana melakukan proses deteksi penggunaan FAKE GPS yang dilakukan oleh pengemudi Go-Jek pada layanan go-ride dan go-food. Gambar diatas dapat menggambarkan proses input-proses dan output pada penelitian yang dilakukan ini

Preprocessing Data

Pemrosesan awal data biasanya dilakukan dalam studi prediksi turnover karyawan karena kumpulan data biasanya berisi entri yang hilang, berbagai tingkat kebisingan, dan perbedaan besar dalam skala per fitur (Shinde et al., 2017).

Data Preprocessing yaitu dilakukan pembersihan dan persiapan data untuk menghilangkan kosistensi data, data tidak lengkap dan redundant data yang terdapat pada data awal. Data Preprocessing juga melakukan pengubahan status barang yang semula true/false menjadi 1/0, yang akan digunakan untuk proses asosiasi.(Triyanto, 2014).

Persiapan data terdiri dari teknik-teknik yang berkaitan dengan analisis data mentah untuk menghasilkan data yang berkualitas, terutama meliputi pengumpulan data, integrasi data, transformasi data, pembersihan data, reduksi data, dan diskritisasi data.(Zhang et al., 2003).

Langkah penting dalam proses penambangan data. Pengumpulan data biasanya merupakan proses yang dikendalikan secara longgar, sehingga berada di luar jangkauan nilai, Menganalisis data yang belum disaring dapat membuat masalah di hasil akhir karena hasil akhir tidak sesuai dengan ekspektasi pada pengolahan data. Dengan demikian, representasi dan kualitas data adalah yang pertama dan terpenting sebelum menjalankan analisis. meliputi persiapan data, ditambah dengan integrasi, pembersihan, normalisasi dan transformasi data; dan tugas pengurangan data. Hasil yang diharapkan Setelah rangkaian tugas pemrosesan data yang andal adalah dataset final (.(Sanjaya et al., 2020).

Import dan Collect data

Tahapan ini merupakan proses awal dalam mempersiapkan library yang dibutuhkan dan melakukan pengambilan data dari directory ke tools Rstudio atau disebut load dataset berikut ini saya tampilkan perintah untuk melakukan import beberapa library yang dibutuhkan dalam proses ini, Selanjut nya melakukan load dataset dari direktori yang telah ditentukan, pada tahapan ini peneliti melakukan load dataset dari keseluruhan dataset yang telah disiapkan, sebagai berikut:

```
gps_train <- read_csv("train.csv")
gps_test <- read_csv("test.csv")
gps <- bind_rows(train = gps_train, test = gps_test, .id = "data")
glimpse(gps)
```

Perintah Load dataset

Dengan perintah glimple(gps) akan menghasilkan dan menampilkan struktur data dan nilai yang terdapat didalam data. Sebagai berikut:

```
## Observations: 648,879
## Variables: 12
## $ data              <chr> "train", "train", "train", "train", "train", …
## $ order_id          <chr> "RB193", "RB193", "RB193", "RB193", "RB193", …
## $ service_type      <chr> "GO_RIDE", "GO_RIDE", "GO_RIDE", "GO_RIDE", "…
## $ driver_status     <chr> "UNAVAILABLE", "AVAILABLE", "AVAILABLE", "AVA…
## $ date              <date> 2018-02-05, 2018-02-05, 2018-02-05, 2018-02-…
## $ hour              <dbl> 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, …
## $ seconds           <dbl> 1548890667, 1548890680, 1548890690, 154889070…
## $ latitude          <dbl> -6.922910, -6.923039, -6.923039, -6.923048, -…
## $ longitude         <dbl> 107.6313, 107.6312, 107.6312, 107.6312, 107.6…
## $ altitude_in_meters <dbl> NA, 712, 712, 713, 713, 713, 713, 713, 713, 7…
## $ accuracy_in_meters <dbl> 23.027, 9.577, 9.577, 8.139, 7.029, 7.029, 3.…
## $ label             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …
```

Tampilan Struktur Data

Kemudian tahapan selanjutnya menyesuaikan dengan deskripsi data yang telah dijelaskan pada bab 3, peneliti melakukan pengkodean ulang untuk beberapa variable sebagai factor. Dimana variable order_id dapat mengelompokan dari observasi "PING" dari setiap perjalanan yang unik.

```
gps$order_id <- as.factor(gps$order_id)
gps$service_type <- as.factor(gps$service_type)
gps$driver_status <- as.factor(gps$driver_status)
gps$label <- as.factor(gps$label)

gps %>%
  count(order_id, name = "nb_of_ping") %>%
  head()
```

```
## # A tibble: 6 x 2
##   order_id nb_of_ping
##   <fct>         <int>
## 1 F0              222
## 2 F1              180
## 3 F10             337
## 4 F100            293
## 5 F1000           117
## 6 F1001           118
```

Perintah Memfaktorkan variable dan grouping

Cek data missing

Nilai yang hilang (missing value) diperhitungkan untuk menjamin bahwa semua algoritme dapat menanganinya. Namun demikian, beberapa algoritme dapat menangani nilai yang hilang secara otomatis tanpa imputasi, seperti XGBoost. Untuk membatasi kompleksitas perbandingan, nilai yang hilang diperhitungkan berdasarkan tipe datanya. Untuk tipe data numerik, entri yang hilang diganti dengan nilai median dari entri lengkap. Untuk data kategoris, entri yang hilang digantikan oleh nilai mode entri lengkap. Sedangkan Nilai yang hilang dalam masalah regresi (yang memiliki variabel respons kuantitatif) lebih mudah ditangani daripada masalah klasifikasi (Xu, 2013).

```
sapply(gps, function(x) sum(is.na(x)))
```

```
##               data           order_id       service_type
##                  0                  0                  0
##      driver_status               date               hour
##                  0                  0                  0
##            seconds           latitude          longitude
##                  0                  0                  0
## altitude_in_meters accuracy_in_meters              label
##             175606                  0              81334
```

Cek NA Value/missing value

Dari gambar 4.5 diatas menunjukan terdapat baris yang mepunyai nilai NA yaitu Label sebanyak 81334 dan altitude_in_meters sebanyak 175606.

Outlier/Anomali Treatment

Dalam memperbaiki outlier dibutuhkan pengecekan terlibih dahulu terhadap outlier menggunakan diagram box, agar dapat dengan jelas dianalis. Dari data yang sudah dilakukan pengecekan terhadap data yang miss dari data gojek final.

Yang terlihat sebagai outlier bukanlah kesalahan pelaporan aktual sementara salah satu fitur penentunya malah dilaporkan secara keliru (yaitu salah satu variabel independen yang memberi makan model). Dengan kata lain, ketika menemukan pengamatan yang jauh, kita tidak dapat berasumsi bahwa itu adalah kesalahan yang sebenarnya sampai kita memeriksa apakah ada variabel lain yang menghasilkannya dilaporkan salah atau tidak.(Benatti & Central Bank, 2018).

Pada proses ini peneliti akan menggunakan library tidyverse dibantu dengan anomalize bahas apemrograman r, dimana Fungsi anomalize() digunakan untuk mendeteksi outlier dalam distribusi tanpa tren atau musiman hadiah. Dibutuhkan output dari time_decompose(), yang telah diturunkan trennya dan menerapkan anomaly metode deteksi untuk mengidentifikasi outlier.(Matt et al., 2020) .

Feature Selection

Metode pemilihan fitur sering digunakan untuk lebih meningkatkan kemampuan prediktif classifier dengan memilih atribut yang

relevan.(Meng et al., 2019). Pada tahapan ini pun akan dideteksi annomali terhadap data, seperti yang telah dilakukan oleh sudiyarno, tahun 2021 menyatakan untuk melakukan deteksi anomali menggunakan feature selection dimana dapat mengurangi redundant data dan mengurangi waktu proses klafikasi serta peningkatan nilai Sebuah proses siklus variable adalah salah satu di mana serangkaian peristiwa terjadi lagi dan lagi dalam urutan yang sama dalam sebuah dataset .pada tahapan ini, Tujuannya adalah untuk mengkodekan ulang variabel sebagai "siklus" (untuk menjaga variabilitas waktu). Memang, jika kita menggunakan penyandian 1-7 untuk hari dalam seminggu, kita memberi tahu model bahwa hari 4 dan 5 sangat mirip, sedangkan hari 1 dan 7 sangat berbeda. Faktanya, hari 1 dan 7 sama seperti hari 4 dan 5. Pengodean ulang menjadi nilai siklus akan dilakukan menggunakan transformasi cos/sin.

Basic statistics variable

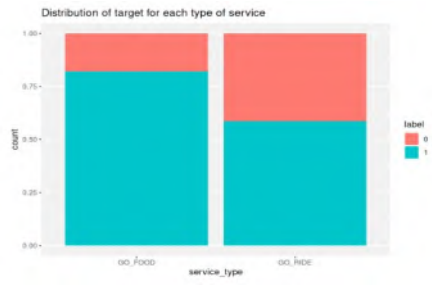
Penggunaan statistik dasar yang tidak memadai adalah penyebab utama kesalahan interpretasi artikel ilmiah. Tujuan dari artikel ulasan ini adalah untuk meninjau beberapa topik statistik dasar untuk mengingatkan penulis Pembaca tentang pentingnya pelaporan statistik dasar yang tepat(Rodrigues et al., 2017)
Pada tahapan ini variabel terakhir yang diperlakukan adalah variabel akurasi_in_meters, dari mana kita akan mengekstrak statistik dasar.

Peneliti telah mengubah kumpulan data asli dengan dimensi 648879 baris x 12 kolom ke kumpulan data baru ini dengan dimensi 4000 baris x 77 kolom, sehingga Ini merupakan sebagai dataset final

Visualisasi Data

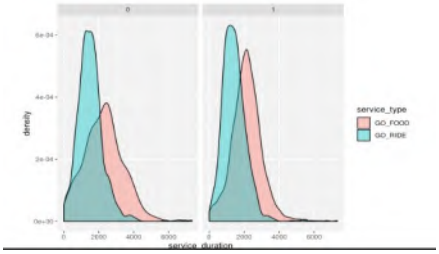
akurasi.(Sudiyarno et al., 2021). Tahapan yang dilakukan menggunakan Fungsi AnomalyDetectionTs dipanggil untuk mendeteksi satu atau lebih anomali yang signifikan secara statistik dalam deret waktu input.

Cyclical variables



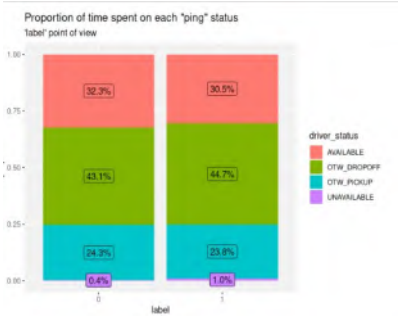
univariate bar chart label(ping service)

Dari char bar diatas menunjukan aktivitas go-food yang lebih mendominasi dari distribusi service disbanding kan go-ride

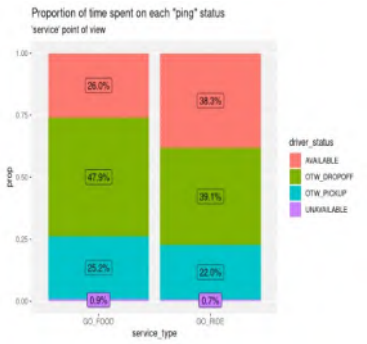


Distribusi ping terhadpa service

Dari char diatas menunjukan aktivitas ping terbanyak dilakukan oleh service go-ride dan begitu pula dengan service dari go-food.



bar char status driver



grafik bar status ping service status driver

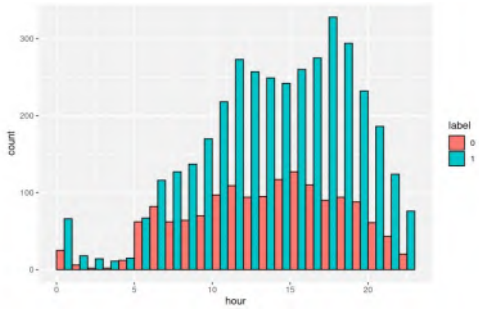
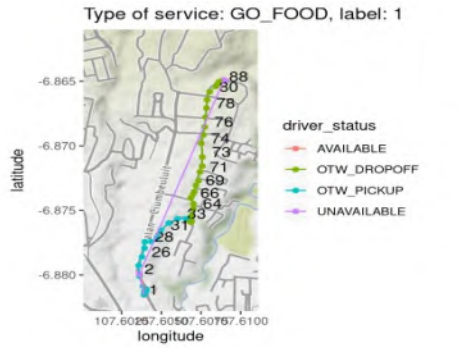


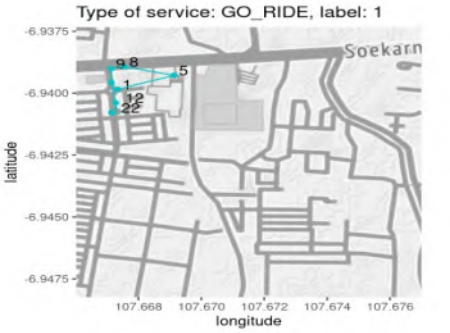
Diagram Bar Count Ping Per Jam

Plot Maps service GO-FOOD label :0



Plot Maps service GO-FOOD label :1

Plot Maps service GO-RIDE label :0



Plot Maps Service GO-RIDE label: 1

Spliting data menjadi dataset Training dan Testing

Untuk membangun algoritma menggunakan dataset train dari gps_final dimana fitur label telah di identifikasikan dan dataset test merupakan yang harus diikut sertakan yang nantinya digunakan pada tahapan evaluasi.

```
gps_train <- gps_final %>%
  filter(data == "train") %>%
  select(-data) %>%
  mutate_if(is.factor, droplevels)

gps_test <- gps_final %>%
  filter(data == "test") %>%
  select(-data) %>%
  mutate_if(is.factor, droplevels)
```

Perintah recreate data train dan test

Dari dataset gps_train, Langkah selanjutnya memisahkan data, Untuk menghindari bias dalam pengujian menggunakan gps_train, pemisahan uji-latihan harus dilakukan sebelum (sebagian besar) langkah persiapan data. Menggunakan set gps_train kemudian membagi data secara acak menjadi 80% di set pelatihan dan 20% di set pengujian.

Berdasarkan penelitian sebelumnya yang memberikan reverensi untuk pembagian jumlah persentasi data dari data set training dan test oleh M. Forest thuan 2020. Yang menyakan bahwa Dataset partion, melakukan pembagian data menjadi dua bagian, yaitu data pelatihan dan data pengujian. Training dataset, data pelatihan sebanyak 80%, sedangkan data pengujian sebanyak 20%. Data pelatihan terdiri dari pengembangan model menggunakan machine learning dan penentuan kinerja model (output data pelatihan). Data pengujian terdiri dari penilaian kinerja model untuk data uji yang tidak diketahui dan tabulasi keakuratan masing-masing algoritma (output data pengujian).(Forest, 2020) .

```
# Random sample indexes
set.seed(314)
index <- sample(x = 1:nrow(gps_train), size = 0.8 * nrow(gps_train))

# Build training and testing sets
train <- gps_train[index, colnames(gps_train) != "order_id"]
test <- gps_train[-index, colnames(gps_train) != "order_id"]
```

Gambar 4. 31 Perintah Random Sample

Perhatikan bahwa kami akan menghapus variabel order_id dari variabel dependen X_train dan X_test, karena variabel ini hanya pengidentifikasi dan dapat menyebabkan beberapa kebocoran dalam algoritme.

Data Processing

Pemodelan

Pada tahapan ini merupakan dimana dataset sudah siap untuk digunakan pada implementasi algoritma dalam ensemble learning.

Peneliti telah mengubah kumpulan data asli dengan dimensi 648879 baris x 12 kolom ke kumpulan data baru ini dengan dimensi 4000 baris x 77 kolom.

Pemodelan Random Forest

Implementasi referensi algoritma hutan acak untuk regresi dan klasifikasi tersedia dalam paket randomForest . Paket ipred memiliki bagging untuk regresi, klasifikasi dan analisis kelangsungan hidup serta bundling, kombinasi beberapa model melalui pembelajaran ensemble. Selain itu, varian hutan acak untuk variabel respons yang diukur pada skala arbitrer berdasarkan pohon inferensi bersyarat diimplementasikan dalam pesta paket .

randomForestSRC menerapkan perlakuan terpadu dari hutan acak Breiman untuk masalah kelangsungan hidup, regresi dan klasifikasi. Hutan regresi kuantil quantregForest memungkinkan untuk meregresi kuantil respons numerik pada variabel eksplorasi melalui pendekatan hutan acak. Untuk data biner, paket varSelRF dan Boruta fokus pada pemilihan variabel melalui algoritma hutan acak. Selain itu, paket ranger dan Rborist menawarkan antarmuka R untuk implementasi C++ cepat dari hutan acak. Pohon Pembelajaran Penguatan, yang menampilkan pemisahan variabel yang akan menjadi penting di bawah pohon, diimplementasikan dalam paket RLT . wsrf mengimplementasikan metode pembobotan variabel alternatif untuk pemilihan subruang variabel sebagai pengganti pengambilan sampel variabel acak tradisional. Paket RGF adalah antarmuka untuk implementasi Phyton dari prosedur yang disebut hutan serakah yang diatur. Hutan acak untuk model parametrik, termasuk hutan untuk estimasi distribusi prediktif, tersedia dalam paket trtf (hutan transformasi prediktif, mungkin di bawah sensor dan pemotongan) dan grf (implementasi hutan acak umum).(Eckstrand et al., 2021).

Salah satu fungsi yang dapat digunakan untuk menjalankan algoritma Random Forest di R adalah randomForest() yang tersedia pada package randomForest.Nilai kualitas yang semula berada pada selang 0-10 dikelompokkan terlbeih dahulu menjadi dua kelas yaitu yang nilainya 0 dan 1 (ping).Pada program di bawah ini, replikasi bootstrap yang dilakukan adalah 1000 kali, yang berarti ada 1000 kali pengambilan sampel ulang dari data dan 1000 kali pembuatan pohon klasifikasi dari setiap sampel tersebut. Pohon yang dibuat sedemikian rupa dengan setting mtry=3 yang berarti bahwa pada proses splitting pemebntukan pohon klasifikasinya, hanya 3 variabel secara acak yang

dicek/dibandingkan untuk menentukan splitting terbaik. Algoritma dijalankan dan model yang diperoleh disimpan pada objek dengan nama model_rf, sebagai berikut:

```
model_rf <- randomForest(as.factor(sales) ~ .,
data=train, ntree = 1000, mtry=3)

model_rf
```

berikut tampilan hasil dari model yang diperoleh :

```
##
## Call:
##  randomForest(formula = as.factor(quality) ~ ., data = data, ntree = 1000,      ntry = 3)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 3
##
##        OOB estimate of  error rate: 8.19%
## Confusion matrix:
##      0   1 class.error
## 0 1346  36   0.0260492
## 1  95 122   0.4377880
```

model Random Forest

Terlihat dari hasil di atas bahwa berdasarkan amatan OOB (out-of-bag), tingkat kesalahan klasifikasi yang diperoleh adalah sekitar 8.2% atau akurasi sebesar 91.8%. akurasi ini cenderung meningkat lebih tinggi jika kita melihat hasil pengerjaan yang sama menggunakan Decision tree.

Pemodelan Adaboost

Ini mengimplementasikan algoritma Adaboost.M1 Freund dan Schapire dan algoritma Bagging Breiman menggunakan pohon klasifikasi sebagai pengklasifikasi individu. Setelah pengklasifikasi ini dilatih, mereka dapat digunakan untuk memprediksi data baru. Juga, estimasi validasi silang dari kesalahan dapat dilakukan. Sejak versi 2.0, fungsi margins() tersedia untuk menghitung margin untuk pengklasifikasi ini. Juga fleksibilitas yang lebih tinggi dicapai dengan memberikan akses ke argumen rpart.control() dari 'rpart'. Empat fitur baru yang penting diperkenalkan pada versi 3.0, AdaBoost-SAMME (Zhu et al., 2009) diimplementasikan dan fungsi baru errorevol() menunjukkan kesalahan ansambel sebagai fungsi dari jumlah iterasi. Selain itu, ansambel dapat dipangkas menggunakan opsi 'newmfinal' di predict.bagging() dan predict. boosting() dan probabilitas posterior setiap kelas untuk observasi dapat diperoleh (Alfaro, 2018)

```
## [1] 0.842518
```

Pemodelan XGBoost

peneliti akan menggunakan kerangka kerja h2o untuk membangun beberapa model. Lebih khusus lagi, proses pembelajaran mesin otomatis automl() sangat praktis untuk membuat model dan secara otomatis menyetel hyperparameter dari setiap algoritme.

```
h2o.init()

##
## H2O is not running yet, starting it now...
##
## Note:  In case of errors look at the following log files:
##     /tmp/RtmpxgNiz/h2o_alex_started_from_r.out
##     /tmp/RtmpxgNiz/h2o_alex_started_from_r.err
##
##
## Starting H2O JVM and connecting: . Connection successful!
##
## R is connected to the H2O cluster:
##     H2O cluster uptime:         2 seconds 260 milliseconds
##     H2O cluster timezone:       Europe/Paris
##     H2O data parsing timezone:  UTC
##     H2O cluster version:        3.26.0.2
##     H2O cluster version age:    2 months and 16 days
##     H2O cluster name:           H2O_started_from_R_alex_tkq024
##     H2O cluster total nodes:    1
##     H2O cluster total memory:   1.73 GB
##     H2O cluster total cores:    2
##     H2O cluster allowed cores:  2
##     H2O cluster healthy:        TRUE
##     H2O Connection ip:          localhost
##     H2O Connection port:        54321
##     H2O Connection proxy:       NA
##     H2O Internal Security:      FALSE
##     H2O API Extensions:         Amazon S3, XGBoost, Algos, AutoML, Core V3, Core V4
##     R Version:                  R version 3.6.1 (2019-07-05)
```

```
train_h2o <- as.h2o(x = train)
test_h2o  <- as.h2o(x = test)

h2o.describe(train_h2o) %>%
  head()

##             Label Type Missing Zeros PosInf NegInf        Min      Max
## 1           label enum       0   843      0      0  0.000000 1.000000
## 2      is_goride enum       0  1392      0      0  0.000000 1.000000
## 3         day_cos real       0     0      0      0 -1.247655 1.442489
## 4      is_weekend enum       0  2083      0      0  0.000000 1.000000
## 5 is_two_day_ride enum       0  2784      0      0  0.000000 1.000000
## 6     is_biz_hour enum       0   415      0      0  0.000000 1.000000
##             Mean        Sigma Cardinality
## 1 6.989286e-01 0.45880563           2
## 2 5.028571e-01 0.50008114           2
## 3 8.895879e-17 1.00000000          NA
## 4 2.560714e-01 0.43653970           2
## 5 5.714286e-03 0.07539007           2
## 6 8.517857e-01 0.35537574           2

# Identify predictors and response
y <- "label"
x <- setdiff(names(train_h2o), y)
```

Pendeteksian Fake GPS

Random Forest

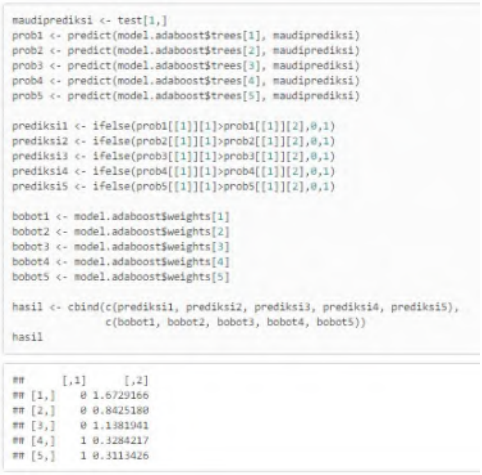
Jika model sudah didapatkan, maka melakukan prediksi dapat dilakukan dengan mudah menggunakan fungsi predict() terhadap amatan yang tersimpan pada suatu dataframe.

Adaboost

Untuk mengilustrasikan bagaimana proses prediksi final menggunakan 5 pohon, program berikut ini melakukan secara berurutan hal-hal berikut:

- menentukan amatan yang mau diprediksi (diambil dari salah satu wine merah yang ada pada data testing)
- memprediksi menggunakan pohon/stump pertama, kedua, hingga yang kelima
- menjumlahkan bobot untuk prediksi kategori 0
- menjumlahkan bobot untuk prediksi kategori 1
- menentukan kelas mana yang suaranya paling tinggi, dan itu adalah kelas prediksinya

Sebagai pembanding, dilakukan juga prediksi langsung menggunakan model hasil boosting.



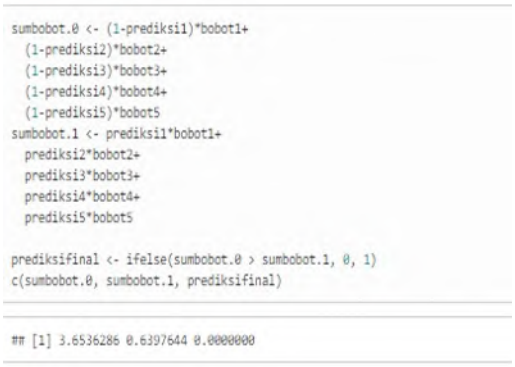
```
maudiprediksi <- test[1,]
prob1 <- predict(model.adaboost$trees[1], maudiprediksi)
prob2 <- predict(model.adaboost$trees[2], maudiprediksi)
prob3 <- predict(model.adaboost$trees[3], maudiprediksi)
prob4 <- predict(model.adaboost$trees[4], maudiprediksi)
prob5 <- predict(model.adaboost$trees[5], maudiprediksi)

prediksi1 <- ifelse(prob1[[1]][1]>prob1[[1]][2],0,1)
prediksi2 <- ifelse(prob2[[1]][1]>prob2[[1]][2],0,1)
prediksi3 <- ifelse(prob3[[1]][1]>prob3[[1]][2],0,1)
prediksi4 <- ifelse(prob4[[1]][1]>prob4[[1]][2],0,1)
prediksi5 <- ifelse(prob5[[1]][1]>prob5[[1]][2],0,1)

bobot1 <- model.adaboost$weights[1]
bobot2 <- model.adaboost$weights[2]
bobot3 <- model.adaboost$weights[3]
bobot4 <- model.adaboost$weights[4]
bobot5 <- model.adaboost$weights[5]

hasil <- cbind(c(prediksi1, prediksi2, prediksi3, prediksi4, prediksi5),
               c(bobot1, bobot2, bobot3, bobot4, bobot5))
hasil
```

```
##      [,1]     [,2]
## [1,]    0 1.6729166
## [2,]    0 0.8425180
## [3,]    0 1.1381941
## [4,]    1 0.3284217
## [5,]    1 0.3113426
```

hasil deteksi menggunakan adaboost

Terlihat bahwa stump pertama, kedua, dan ketiga memberikan prediksi bahwa anggur merah tersebut termasuk kelas kurang (kelas 0), sedangkan stump keempat dan kelima prediksinya masuk ke kelas baik (kelas 1). Kolom kedua menampilkan bobot dari masing-masing stump.

Berikut ini perhitungan jumlah bobot dari stump yang memberikan prediksi 0 dan stump yang memberikan prediksi 1.



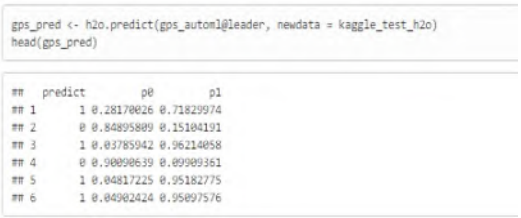
```
sumbobot.0 <- (1-prediksi1)*bobot1+
  (1-prediksi2)*bobot2+
  (1-prediksi3)*bobot3+
  (1-prediksi4)*bobot4+
  (1-prediksi5)*bobot5
sumbobot.1 <- prediksi1*bobot1+
  prediksi2*bobot2+
  prediksi3*bobot3+
  prediksi4*bobot4+
  prediksi5*bobot5

prediksifinal <- ifelse(sumbobot.0 > sumbobot.1, 0, 1)
c(sumbobot.0, sumbobot.1, prediksifinal)
```

```
## [1] 3.6536286 0.6397644 0.0000000
```

Bobot Stump hasil deteksi

Jumlah bobot suara dari stump yang memberikan prediksi 0 adalah 3.6 sedangkan jumlah bobot suara stump yang memberikan prediksi 1 adalah 0.6. Karena lebih besar yang 0, maka kelas prediksi final/akhir adalah 0.

Extra Gradiens Boost(XGBoost)



```
gps_pred <- h2o.predict(gps_automl@leader, newdata = kaggle_test_h2o)
head(gps_pred)
```

```
##   predict        p0        p1
## 1       1 0.28170026 0.71829974
## 2       0 0.84895889 0.15104191
## 3       1 0.03785942 0.96214058
## 4       0 0.90090639 0.09909361
## 5       1 0.04817225 0.95182775
## 6       1 0.04902424 0.95097576
```

perintah membuat prediksi xgboots

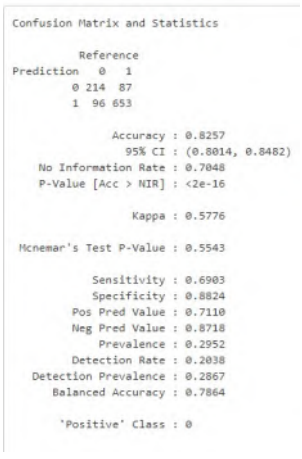
```
# Create final csv file
final <- tibble(order_id = gps_test$order_id,
                label = as.vector(as.numeric(gps_pred$predict)))
head(final)
```

```
## # A tibble: 6 x 2
##   order_id label
##      <fct>  <int>
## 1 F1770      1
## 2 M83125     0
## 3 M8714      1
## 4 M8610      0
## 5 F1432      1
## 6 F100       1
```

hasil prediksi xgboost berdasarkan order_id

Evaluasi

Random Forest



```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 214  87
         1  96 653

               Accuracy : 0.8257
                 95% CI : (0.8014, 0.8482)
    No Information Rate : 0.7048
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.5776

 Mcnemar's Test P-Value : 0.5543

            Sensitivity : 0.6903
            Specificity : 0.8824
         Pos Pred Value : 0.7110
         Neg Pred Value : 0.8718
             Prevalence : 0.2952
         Detection Rate : 0.2038
   Detection Prevalence : 0.2867
      Balanced Accuracy : 0.7864

       'Positive' Class : 0
```

Confusion matrik random forest

Akurasi keseluruhan dari model 500 pohon kami adalah sekitar 82,5%, yang cukup bagus tetapi tidak jauh lebih baik dari model awal kami.

Adaboost

Peneliti dapat menggunakan data test untuk melakukan proses pemeriksaan performa prediksi model. Tahapannya adalah: pertama amatan pada data test kita prediksi menggunakan model adaboost, selanjutnya hasil prediksi kita bandingkan dengan kelas aslinya.

fungsi confusionMatrix() kita gunakan untuk menampilkan berbagai ukuran ketepatan klasifikasi.

```
prediksi.boost <- predict(model.adaboost, test)$class
confusionMatrix(as.factor(prediksi.boost),
                test$quality, positive = "1")
```

```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction   0   1
##          0 398  34
##          1  16  31
##
##               Accuracy : 0.8856
##                 95% CI : (0.8647, 0.9045)
##    No Information Rate : 0.8643
##    P-Value [Acc > NIR] : 0.02158
##
##                  Kappa : 0.4962
## Mcnemar's Test P-Value : 0.03621
##
##            Sensitivity : 0.47692
##            Specificity : 0.96135
##         Pos Pred Value : 0.65957
##         Neg Pred Value : 0.92138
##             Prevalence : 0.13570
##         Detection Rate : 0.06472
##   Detection Prevalence : 0.09812
##      Balanced Accuracy : 0.71914
##
##       'Positive' Class : 1
##
```

confusin matrix adaboost

Secara umum kita bisa lihat bahwa akurasinya sekitar 89,5% meningkat dari sebelumnya pada random forest. dapat mengubah banyaknya iterasi pada proses boosting (gunakan opsi gps_final) dengan mengubah ke nilai yang jauh lebih besar dari 5. Default dari fungsi ini adalah 100.

Extra Gradiens Boost (XGBoost)

Dalam model terbaik kami, akurasi GPS tampaknya menjadi prediktor terbaik. fungsi confusionMatrix() penelitti gunakan untuk menampilkan berbagai ukuran ketepatan klasifikasi. peneliti juga dapat melihat confusion matrik pada set pengujian.

```
##                                          model_id       auc  logloss
## 1    StackedEnsemble_AllModels_AutoML_20191015_153643 0.9007076 0.3604215
## 2  StackedEnsemble_BestOfFamily_AutoML_20191015_153643 0.9004798 0.3599897
## 3                    XGBoost_1_AutoML_20191015_153643 0.9001660 0.3512131
## 4                        GBM_2_AutoML_20191015_153643 0.8963142 0.3623349
## 5                        GBM_3_AutoML_20191015_153643 0.8937885 0.3650331
## 6                        GBM_5_AutoML_20191015_153643 0.8935506 0.3630208
## 7                    XGBoost_3_AutoML_20191015_153643 0.8925534 0.3637106
## 8                    XGBoost_2_AutoML_20191015_153643 0.8911008 0.3677750
## 9                        GBM_4_AutoML_20191015_153643 0.8890407 0.3779071
## 10                       GBM_1_AutoML_20191015_153643 0.8854116 0.3801981
##    mean_per_class_error      rmse       mse
## 1             0.2295918 0.3392682 0.1151029
## 2             0.2062075 0.3390417 0.1149492
## 3             0.2174036 0.3382877 0.1144386
## 4             0.2434807 0.3424413 0.1172660
## 5             0.2158447 0.3449120 0.1189643
## 6             0.2406463 0.3446288 0.1187690
## 7             0.2560941 0.3451568 0.1191332
## 8             0.1965703 0.3458641 0.1196220
## 9             0.2356859 0.3491716 0.1219208
## 10            0.2633220 0.3519303 0.1238549
```

```
## Confusion Matrix (vertical: actual; across: predicted)  for max f1 @ threshold = 0.366185574791611:
##        0   1   Error     Rate
## 0    128  76 0.387755  =76/196
## 1     36 468 0.071429  =36/504
## Totals 156 544 0.180000  =112/700
```

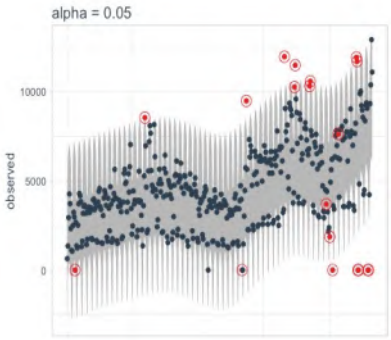
confusion matrix xgboost

Dapat dilihat dari confusion metrix yang diperoleh menunjukan peningkatan yang cukup tinggi xgboost 1 mencapai 90%, xgboost 2 mencapai 89,6% dan xgboost ke 3 mencapai 89,3%.

4.2 Pembahasan

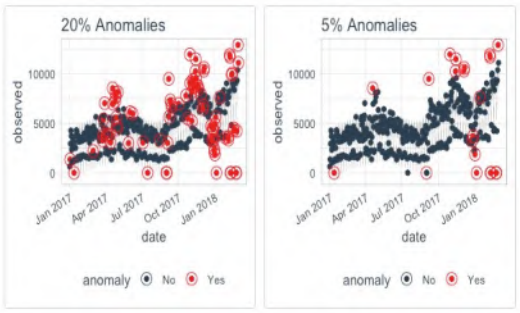
4.2.1 Pendeteksian Anomali

Parameter alpha dan max_anoms adalah dua parameter yang, mengontrol anomalize() fungsi. Penulis melakukan penyesuaian alpha, yang diatur ke 0,05 secara default. Secara default, pita hanya menutupi bagian luar jangkauan.



alfa o,5 pada dua parameter

Untuk menyesuaikan alpha = 0.3 sehingga hampir semua hal adalah outlier. Peneliti mencoba perbandingan antara max_anoms = 0.2(20% anomali diperbolehkan) dan max_anoms = 0.05(5% anomali diperbolehkan).



timeseries detection anomaly

Pendeteksian fake GPS Go-Jek dan Evaluasi

Dari pembahasan sebelumnya diperoleh hasil dari algoritma yang digunakan yang sudah dipastikan menggunakan fasilitas fake gps, berdasarkan algoritma yang digunakan adalah sebagai berikut:

- random forest 653 aktivitas servce yang terdekteksi posistif
- adaboost 31 aktifivitas service yang terdeteksi positif
- xgboost 468 actifitas service yang terdeteksi positif

Analisa Dari Literatur Sebelumnya

Dari Studi literature journal yang dilakukan dalam penelitian ini, penelitian yang telah dilakukan oleh (Farhanna Mar'i dan gusti pangestu. 2021) dengan judul "Classification of Fake GPS in GOJEK Application using Logistic Regression" menunjukan hasil akurasi dari metode yang digunakan sebagai berikut: Berdasarkan hasil penelitian, penerapan Regresi Logistik untuk klasifikasi GPS palsu pada aplikasi GOJEK diperoleh nilai akurasi maksimum sebesar 74,7%, Precision 74,7%, Recall 99,4%, dan skor F1 sebesar 85,5%.(Mar'i & Pangestu, 2021)

Sedangkan hasil penelitian yang telah ddilakukan oleh peneliti saat ini mendapatkan hasil akurasi deteksi yang cukup baik dari penelitian sebelumnya, baik algoritma Random Forest 82,5%, adaboost 89,5%, dan XGBoost dengan rata-rata akurasi mencapai 99%.

Implikasi Penelitian

Ensemble Learning menggunakan algoritma Random Forest, Adaboost dan XG Boost dapat digunakan untuk melakukan Deteksi terhadap fake gps aplikasi driver Go-Jek baik Go-Food maupun Go-Ride, yang mana dari ke-tiga algoritma baik Random Forest, Adaboost, dan XGBoost (Extra Gradien Boost) memiliki tingkat akurasi yang mendekati akurat, sehingga dapat menjadi referensi, terhadap penelitian yang akan dilakukan selanjutnya dan untuk permasalahan pengklasifikasian lainnya.

KESIMPULAN DAN SARAN

Kesimpulan

Dalam penelitian ini menyajikan kumpulan dataset dari fake gps gojek, dengan hasil klasifikasi biner menggunakan pembelajaran mesin dan pembelajaran mendalam.

- Penggunaan Paket library tidyverse R dapat mendeteksi Anomali FAKE GPS driver Go-Jek dapat membantu melakukan deteksi dengan baik.
- Penerapan algoritma Random forest, adaboost, XGboost dapat digunakan untuk melakukan pendeteksian fake gps pada go-jek
- Dari hasil pengujian yang dilakukan terhadap performa model dengan akurasi deteksi menggunakan algoritma Random Forest 82%, Adaboost 89% dan XGBoost 90% dalam Melakukan Deteksi FAKE GPS driver GO-JEK, sangat baik dan mendekati sempurna.

Saran

- Ensemble Learning menggunakan algoritma Random Forest, Adaboost dan XG Boost dapat digunakan untuk melakukan Deteksi terhadap fake gps aplikasi driver Go-Jek baik Go-Food maupun Go-Ride, yang mana dari ke-tiga algoritma baik Random Forest, Adaboost, dan XGBoost (Extra Gradien Boost) memiliki tingkat akurasi yang mendekati akurat, sehingga dapat menjadi referensi, terhadap

penelitian yang akan dilakukan selanjutnya dan untuk permasalahan pengklasifikasian lainnya

- Pada penggunaan H2o library pada bahsa pemrograman R harus diperhatikan terkait versi java jdk dari 8-11, dan hanya dapat digunakan pada OS windos 64 bit.
- Harap menjadi perhatian pada proses preparation atau pre-processing data, karena proses ini merupakan tahapan yang krusial dan critis, untuk mendapatkan pemodelan yang baik. .
- Pada saat melakukan deteksi anomaly diharapkan untuk mendapatkan hasil yang maksimal, selain menggunakan library tidyverse, juga menggunakan library anomalize agar dapat melihat timeseries anomaly yang terjadi berdasarkan tanggal
-

DAFTAR PUSTAKA

Aberson, C. L. (2015). Statistical Power Analysis. Emerging Trends in the Social and Behavioral Sciences, 1–14. https://doi.org/10.1002/9781118900772.etrds0319

Alfaro, A. (2018). Package ‘ adabag .’

Benatti, N., & Central Bank, E. (2018). A machine learning approach to outlier detection and imputation of missing data. August, 30–31.

Boinee, P., Angelis, A. De, & Foresti, G. L. (2008). Meta Random Forests. World Academy of Science, Engineering and Technology, 18(6), 1148–1157.

Budiawan, T., Santoso, I., Zahra, A. A., Elektro, J. T., Teknik, F., & Diponegoro, U. (n.d.). Mobile tracking gps (global positioning system) melalui media sms (short message service).

Chang, Y. (2018). Fake GPS Defender : A Server-side Solution to Detect Fake GPS. c, 36–41.

Eckstrand, E., Hill, B., Vidrio, S., Wang, A., & Peck, R. (2021). R topics documented :

Forest, M. D. (2020). MultiClass Decision Forest Machine Learning Artificial Intelligence. 4(1), 1–7.

Gao, X., Wen, J., & Zhang, C. (2019). An Improved Random Forest Algorithm for Predicting Employee Turnover. Mathematical Problems in Engineering, 2019. https://doi.org/10.1155/2019/4140707

Hijmans, R. J., Williams, E., & Vennes, C. (2019). Spherical Trigonometry: Package “geosphere.” Cran, 1–45. http://www.movable-type.co.uk/scripts/latlong.html

Krotov, V. (2017). A Quick Introduction to R and RStudio ® Tutorial. November. https://doi.org/10.13140/RG.2.2.10401.92009

Mar’i, F., & Pangestu, G. (2021). Classification of Fake GPS in GOJEK Application using Logistic Regression.

Matt, A., Vaughan, D., & Dancho, M. M. (2020). Package ‘ anomalize .’ 1.

Meng, Q., Hsu, L. T., Xu, B., Luo, X., & El-Mowafy, A. (2019). A GPS spoofing generator using an open sourced vector tracking-based receiver. Sensors (Switzerland), 19(18). https://doi.org/10.3390/s19183993

Rodrigues, C. F. de S., Lima, F. J. C. de, & Barbosa, F. T. (2017). Importance of using basic statistics adequately in clinical research. Brazilian Journal of Anesthesiology (English Edition), 67(6), 619–625. https://doi.org/10.1016/j.bjane.2017.01.011

Sanjaya, J., Renata, E., Budiman, V. E., Anderson, F., & Ayub, M. (2020). Prediksi Kelalaian Pinjaman Bank Menggunakan

Random Forest dan Adaptive Boosting. Jurnal Teknik Informatika Dan Sistem Informasi, 6(1), 50–60. https://doi.org/10.28932/jutisi.v6i1.2313

Setiawan, A. (2015). Pengantar Teori Probabilitas. June 2015, 215. https://www.researchgate.net/publication/301202850

Sharma, S., & Berger, P. D. (2017). a Predictive Workforce-Analytics Model for Voluntary Employee Turnover in the Banking/ Financial-Service Industry. Global Journal of Human Resource Management, 5(1), 47–59. www.eajournals.org

Shinde, P. P., Oza, K. S., & Kamat, R. K. (2017). Big data predictive analysis: Using R analytical tool. Proceedings of the International Conference on IoT in Social, Mobile, Analytics and Cloud, I-SMAC 2017, February, 839–842. https://doi.org/10.1109/I-SMAC.2017.8058297

statisctical power.pdf. (n.d.).

Sudiyarno, R., Setyanto, A., & Luthfi, E. T. (2021). Peningkatan Performa Pendeteksian Anomali Menggunakan Ensemble Learning dan Feature Selection. Creative Information Technology Journal, 7(1), 1. https://doi.org/10.24076/citec.2020v7i1.238

Szapkiw, R.-. (2013). Statistics Guide.

Triyanto, W. A. (2014). Association Rule Mining Untuk Penentuan Rekomendasi Promosi Produk. Journal SIMETRIS, Vol.5(No.2), 121–126.

Wang, X., & Lu, X. (2020). A host-based anomaly detection framework using XGBoost and LSTM for IoT devices. Wireless Communications and Mobile Computing, 2020. https://doi.org/10.1155/2020/8838571

Xu, R. (2013). Improvements to random forest methodology. Dissertation (Doctor of Philosophy) Iowa State University, 1–88. http://lib.dr.iastate.edu/etd/13052/

Yedida, R., Reddy, R., Vahi, R., Jana, R., GV, A., & Kulkarni, D. (2018). Employee Attrition Prediction. http://arxiv.org/abs/1806.10480

Zhang, S., Zhang, C., & Yang, Q. (2003). Data preparation for data mining. Applied Artificial Intelligence, 17(5–6), 375–381. https://doi.org/10.1080/713827180

Zhao, Y., Hryniewicki, M. K., Cheng, F., Fu, B., & Zhu, X. (2018). Employee turnover prediction with machine learning: A reliable approach. In Advances in Intelligent Systems and Computing (Vol. 869). Springer International Publishing. https://doi.org/10.1007/978-3-030-01057-7_56