Prediksi Risiko Perjalanan Transportasi *Online* Dari Data Telematik Menggunakan Algoritma *Support Vector Machine*

http://dx.doi.org/10.28932/jutisi.v6i2.2672

Christ Memory Sitorus ✉ #1, Adhi Rizal#2, Muhammad Jajuli#3

#*Teknik Informatika, Universitas Singaperbangsa Karawang*
Jl. HS. Ronggo Waluyo, Telukjambe Timur, Karawang.
[1]1610631170062@student.unsika.ac.id
[2]adhi.rizal@staff.unsika.ac.id
[3]mohamad.jajuli@unsika.ac.id

Abstract — **The ride-hailing service is now booming because it has been helped by internet technology, therefore many call this service online transportation. The magnitude of the potential for growth in online transportation service users also increases the risk of user satisfaction which could have declined therefore the company is increasing in its service. Both in terms of application and services provided by partners/drivers of the company. During each trip, the online transportation application will record device movement data and send it to the server. This data set is usually called telematic data. This telematics data if processed can have enormous benefits. In this study, an analysis will be conducted to predict the risk of online transportation trips using the Support Vector Machine (SVM) algorithm based on the obtained telematic data. The data obtained is telematic data so it must be processed first using feature engineering to obtain 51 features, then trained using the SVM algorithm with RBF kernel and modified C values. Every C value that is changed will be used K-Fold cross-validation first to separate the testing data and training data. The specified k value is 5. The results for each trial obtained accuracy, Receiver Operating Characteristic (ROC) and Area Under the Curves (AUC), for the best that is at C = 100 while the worst at C = 0.001.**

Keywords— **Feature Engineering; Support Vector Machine; Telematic.**

I. PENDAHULUAN

Layanan *ride-hailing* merupakan skema transportasi alternatif yang sedang populer sekarang ini. Layanan *ride-haillling* menjadi *booming* karena sudah dibantu dengan teknologi internet, oleh karena itu banyak yang menyebut layanan ini dengan transportasi *online*.

Banyak perusahaan perintis yang sukses pada layanan transportasi *online*. Contohnya Uber yang sudah sukses lebih dahulu di Amerika, kemudian Grab dari Singapura serta Gojek dari Indonesia. Sebelumnya ketiga nama besar perusahaan tersebut bersaing di ASEAN. Namun pada 2018 Uber memutuskan menghentikan semua operasinya dari ASEAN [1]. Oleh karena itu hanya tersisa dua pesaing besar yaitu Grab dan Gojek.

Besarnya potensi pada pertumbuhan pengguna layanan transportasi *online* juga meningkatkan risiko kepuasan pengguna yang bisa saja menurun. Oleh karena itu perusahaan melakukan peningkatan dalam pelayannya. Baik dalam segi aplikasi dan juga pelayanan yang diberikan oleh mitra/*driver* perusahaan tersebut. Contoh peningkatan dari segi aplikasi, perusahaan melakukan perbaikan pada kecepatan aplikasi, kemudahan penggunaan (*user experience*), bahkan gojek juga melakukan perbaikan pada algoritma untuk pencarian *driver* [2]. Sementara itu dari segi pelayanan yang diberikan oleh *driver* pada pengguna, perusahaan memberikan opsi pada pengguna untuk menilai driver menggunakan fitur rating pada aplikasi.

Fitur *rating* pada aplikasi transportasi *online* seperti Gojek dan Grab berfokus pada penilaian pelayanan dari *driver*. Pelayanan yang dimaksud yaitu keramahan, kondisi kendaraan, kemampuan mengemudi dan lain-lain. Namun walaupun pelayanan yang diberikan *driver* kurang baik seringkali pengguna merasa sungkan untuk memberi *rating* kecil. Untuk beberapa pengguna asal memberikan *rating* sehingga akurasi dari *rating drive*r patut dipertanyakan [3] terlebih dengan kaitannya dengan risiko perjalanan. Berdasarkkan pemaparan sebelumnya penggunaan sistem rating untuk menilai risiko yang berkaitan dengan berbahaya atau amannya perjalanan masih kurang tepat.

Dalam setiap perjalanan, aplikasi transportasi *online* akan merekam data pergerakan *device* dan mengirimnya ke peladen. Kumpulan data ini biasa disebut dengan data telematik. Data telematik biasanya menyimpan banyak sekali data dan atribut, untuk jenis telematik yang berhubungan dengan perjalanan atau kendaraan biasanya terdapat data *Global Positioning System* (GPS), kecepatan

akselerasi dan lain-lain. Data tersebut akan sangat banyak tergantung lamanya perjalanan. Bahkan dalam satu perjalanan saja dapat memiliki ribuan data *point* yang mencatat aktivitas/pergerakan dan perubahan pada *device* yang sudah terpasang aplikasi transportasi *online*.

Data telematik yang akan digunakan untuk penelitian ini diperoleh dari [4]. Berdasarkan data tersebut akan dibuat model untuk memprediksi apakah perjalanan berbahaya atau tidak, namun banyaknya data dalam satu perjalanan akan menyulitkan proses *modeling*. Oleh karena itu akan dilakukan proses *feature engineering* terlebih dahulu. Tahapan *feature engineering* akan meningkatkan kinerja pemodelan untuk prediksi dengan merubah ruang fiturnya [5]. Banyaknya data dalam satu perjalanan akan diwakili oleh fitur-fitur yang dibuat dari proses *feature engineering*. Sehingga hanya ada 1 data saja untuk setiap perjalanan.

Support Vector Machine merupakan algoritma berbeda dengan algoritma lainnya seperti misalnya *Neural Network* (NN), dimana SVM menggunakan data terpilih saja yang akan berkontribusi pada model. Data-data yang akan berkontribusi pada model disebut dengan *support vector*. SVM memiliki kernel yang digunakan untuk mentransformasikan inputan ke bentuk yang diperlukan. Pada penelitian prediksi risiko perjalanan dari data telematik menggunakan SVM akan digunakan kernel *Gaussian radial basis function* (RBF).

Masalah yang dikaji pada penelitian ini adalah bagaimana *feature engineering* dilakukan pada dataset telematik perjalanan Grab dan juga pemodelannya menggunakan SVM dengan kernel RBF. Tujuan dari penelitian ini adalah melakukan *feature engineering* dan mengimplementasikan data hasil *feature engineering* untuk pembuatan model menggunakan algoritma SVM dengan kernel RBF.

Hasil akhir yang diharapkan dari penelitian ini yaitu model data yang dapat digunakan untuk memprediksi risiko perjalanan apakah berbahaya atau aman berdasarkan data telematik perjalanan Grab serta hasil evaluasi model.

II. LANDASAN TEORI

A. Data Telematik

Data telematik merupakan kumpulan data yang diperoleh dari perangkat yang mendukung. Jenis data telematik sangat banyak sumbernya, salah satu sekarang yang sedang berkembang adalah yang berasal dari *smartphone*. Sistem telematik berbasis *smartphone* menjadi populer karena bersifat terbuka dan alternatif murah dari sensor dan komunikasi khusus yang mahal [6].

Selain murah juga dikarenakan *smartphone* sekarang sudah sangat canggih karena didukung dengan sensor yang cukup banyak mulai dari *accelerometer*, *gyroscope* hingga temperature sensor. Dari dataset telematik yang digunakan pada penelitian ini terdapat beberapa data sensor yaitu gps *bearing*, *acceleratio*n, *accuracy*, *gyroscope*, *second* dan *speed*.

B. Feature Engineering

Feature Engineering merupakan tugas utama dalam persiapan data untuk *machine learning* [5]. Proses ini adalah tahapan membangun *feature* yang sesuai dari *feature* yang diberikan, dan mengarah pada peningkatan kinerja prediksi atau klasifikasi. Menurut [7], *feature engineering* merupakan aktivitas mengekstraksi *feature* dari data mentah dan mengubah nya ke format yang sesuai dengan model *machine learning*.

Feature Engineering biasanya melibatkan penerapan fungsi transformasi seperti operator aritmatika dan agregat pada fitur yang diberikan untuk menghasilkan yang *feature* yang baru atau mengubah hubungan *non-linear* antara fitur dan kelas target menjadi hubungan *linear*, yang lebih mudah dipelajari.

C. Support Vector Machine

Support Vector Machine (SVM) diperkenalkan oleh Vapnik pada tahun 1992. Metode ini berakar dari teori pembelajaran statistik yang hasilnya menjanjikan untuk memberikan hasil yang lebih baik dari metode lain [8]. Dibanding dengan teknik klasifikasi di era 1980-an seperti *decision tree* dan *Artificial Neural Network* (ANN), SVM memiliki konsep yang lebih matang.

SVM akan memaksimalkan batas *hyperlane*, dengan memaksimalkan jarak antar kelas (*margin*). Algoritma ini dapat digunakan untuk masalah klasifikasi atau regresi. Namun, kebanyakan digunakan dalam menyelesaikan masalah klasifikasi. Dalam algoritma ini, setiap item data diplot sebagai titik dalam ruang n-dimensional (di mana n adalah jumlah fitur yang miliki) dengan nilai setiap fitur menjadi nilai koordinat tertentu. Kemudian, dilakukan klasifikasi dengan menemukan *hyper-plane* yang membedakan dua kelas dengan sangat baik.

.

III. PENELITIAN TERKAIT

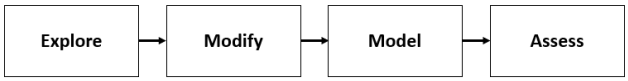
Data telematik jika diolah dapat memberikan manfaat yang sangat besar oleh karena itu penelitian terkait dengan data telematik sudah dilakukan sebelumnya yaitu analisis data telematik dari aplikasi *mobile* yang diperoleh dari perusahaan asuransi untuk memprediksi jenis gender pengemudi menggunakan *Choquet fuzzy* yang dimodifikasi [9]. Pada penelitian tersebut *feature* yang digunakan diantaranya *speed, acceleration(x,y), Yaw rate, Pitch rate, Roll rate* dan *GPS heading*. Dikarenakan data telematik sangat berbeda tergantung alat yang digunakan untuk mengumpulkannya maka wajar jika antara satu penelitian dan yang lainnya memiliki *feature* atau *atribut* yang berbeda untuk dianalisis. Selain itu penelitian lain yang telah dilakukan yaitu identifikasi pengemudi menggunakan data telematik kendaraan menggunakan algoritma *Random Forest* [10]. Dari kedua penelitian tersebut alat yang digunakan untuk mengumpulkan data merupakan alat yang dipasang pada kendaraan dan dibuat dengan tujuan pengumpulan data. Hal tersebut berbeda dengan penelitian

ini dimana pada penelitian ini data dikumpulkan menggunakan aplikasi pada *smartphone* yang digunakan oleh pengguna aplikasi.

Adapun penelitian yang menggunakan *smartphone* sebagai alat pengumpulan datanya [6], namun aplikasi yang digunakan, dibuat dengan tujuan pengumpulan data sehingga berbeda hasil pengumpulan atau pun data yang diprosesnya.

IV. METODOLOGI

Metode penelitian yang digunakan pada penelitian ini adalah adaptasi dari metode data mining yaitu SEMMA (*SAMPLE*, *EXPLORE*, *MODIFY*, *MODEL*, *ASSESS*). Adaptasi ini dilakukan karena SEMMA cocok sekali digunakan dan memberikan keleluasaan karena tahapnya yang lebih sedikit dibanding dengan model proses data mining lain seperti KDD dan CRISP-DM. Namun tahapan sample tidak akan dilakukan karena tahapan ini bersifat opsional [11]. Alurnya digambarkan pada Gambar 1.



Gambar 1. Metodologi Penelitian

A. *Explore*

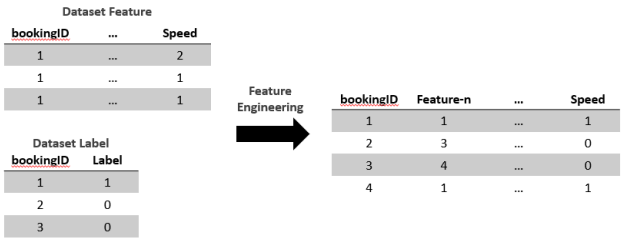
Pada tahapan ini dataset yang sudah didapatkan akan dieksplorasi dengan mencari *trend* dan anomali yang tidak terduga, tujuannya untuk mendapatkan pemahaman dan ide. Tahapan yang dilakukan untuk menjawab pertanyaan-pertanyaan seperti bagaimana distribusi jumlah data antara pengendara berbahaya dan aman, apakah dalam setiap perjalanan terdapat jumlah data telematik yang sama, bagaimana hasil *summary* statistik keseluruhan data, apakah data berbentuk *time series* dan lain-lain.

B. *Modify*

Pada tahapan ini terdiri atas modifikasi data dengan membuat, memilih, dan mentransformasikan variabel untuk memfokuskan proses modeling, tujuannya agar data lebih mudah diproses. Tahapan ini juga melakukan proses pengurangan data yang dipertimbangkan pada tahapan sebelumnya.

Beberapa hal yang dilakukan pada tahapan ini yaitu menyatukan dataset yang terpisah menjadi satu (integrasi), kemudian karena ukuran dataset terlalu besar sehingga diperlukan proses untuk kompresi ke bentuk lain sehingga proses read data semakin cepat. Selain itu pada tahapan ini juga dilakukan data cleaning dan data *preprocessing* lalu pada tahap terakhir dilakukan *feature engineering*.

1) Feature Engineering: Tahapan ini akan mengekstraksi dari beberapa kolom data dan baris pada satu perjalanan menjadi *feature* (Lihat Gambar 2). Untuk jenis *feature* yang akan digunakan diantaranya yaitu *statistical feature,* menghitung jumlah perilaku pengemudi di luar *outlier*, dan nilai dari *Risk Driving Score (RDS)*.



Gambar 2. Mekanisme *feature engineering*

C. *Model*

Pada tahapan ini dilakukan proses untuk pemodelan yaitu tahapan menentukan kombinasi *feature* yang akan digunakan bedasarkan korelasi antara feature dengan label.

Setelah kombinasi *feature* didapatkan kemudian data testing dan training akan di split menggunakan *k-fold cross validation* dengan *fold* 5. Lalu akan di training menggunakan SVM dengan kernel linier dan parameter sequential training untuk nilai parameter C nya 0,001, 0,01, 0,1, 1, 10, 50, 100 sedangkan yang lain *default.* Pengujian nilai C ini digunakan karena untuk menangani data *imbalanced* pada *dataset*. Setelah dilakukan *training* dan diuji, hasil uji berupa *confussion matrix* akan dinilai pada tahap selanjutnya yaitu *assessment*.

D. *Assess*

Tahap ini terdiri atas penilaian data dengan mengevaluasi kegunaan dan keandalan model dari proses yang telah dilakukan. Untuk evaluasi digunakan dua cara yaitu akurasi dan nilai ROC nya. Setiap fold menggunakan *k-fold cross validation* akan diamati akurasi dan nilai *True Positive Rate* (TPR) dan *False Positive Rate* (FPR) nya dari hasil *confussion matrix*.

Nilai TPR dan FPR diperoleh dari rumus 1 dan 2.

$$TPR = \frac{True\ Positives}{True\ Postives - False\ Negative} \quad (1)$$

$$FPR = \frac{False\ Positives}{False\ Positives + True\ Negatives} \quad (2)$$

Kemudian hasil TPR dan FPR dari setiap fold akan diplot untuk melihat kurva ROC-nya.

V. HASIL DAN PEMBAHASAN

A. *Explore*

Tahapan awal yang dilakukan adalah mengeksplorasi struktur dataset. Seperti yang dijelaskan pada bab sebelumnya, *dataset* yang diperoleh terdiri dari dua, yaitu *dataset feature* dan *dataset* label. *Dataset feature* terdiri dari beberapa atribut yaitu diantaranya *bookingID, Accuracy, Bearing, acceleration_x, acceleration_y, acceleration_z, gyro_x, gyro_y, gyro_z, second dan speed*. Sementara *dataset* label berisi *bookingID* dan label.

Untuk melakukan pemahaman lebih lanjut diperlukan pengetahuan terhadap setiap atribut, karena data yang diperoleh merupakan data telematik maka sangat penting mengetahui satuan fisika atribut tersebut. Dari file dataset ini, diperoleh informasi seperti yang tertera pada Tabel I.

TABEL I
KETERANGAN ATRIBUT *DATASET FEATURE*

No	Nama Atribut	Keterangan
1	bookingID	id perjalanan
2	Accuracy	Akurasi disimpulkan oleh GPS dalam meter, semakin tepat jika nilainya mendekati atau 1
3	Bearing	Bantalan GPS dalam derajat
4	acceleration_x	Percepatan dari *accelerometer* dalam sumbu x (m/s^2)
5	acceleration_y	Percepatan dari *accelerometer* dalam sumbu y (m/s^2)
6	acceleration_z	Percepatan dari *accelerometer* dalam sumbu z (m/s^2)
7	gyro_x	*Gyroscope* dalam sumbu x (rad/s)
8	gyro_y	*Gyroscope* dalam sumbu y (rad/s)
9	gyro_z	*Gyroscope* dalam sumbu z (rad/s)
10	second	Catatan waktu dalam detik
11	Speed	Kecepatan diperoleh dari GPS (m/s)

Selain mengetahui keterangan setiap atributnya, perlu juga informasi seperti jumlah observasi hingga tipe datanya. Dari Gambar 3 diketahui bahwa terdapat 16.135.561 observasi dari *dataset feature*. Selain itu juga diperoleh informasi semua atribut/variabel bertipe data dbl atau *double*.

```
Observations: 16,135,561
Variables: 11
$ bookingID      <dbl> 1202590843006, 274877907034, 884763263056, 107374182...
$ Accuracy       <dbl> 3.000, 9.293, 3.000, 3.900, 3.900, 3.900, 10.000, 3....
$ Bearing        <dbl> 353.00, 17.00, 189.00, 126.00, 50.00, 178.00, 262.18...
$ acceleration_x <dbl> 1.22887, 0.03277, 1.13967, 3.87154, -0.11288, 0.8056...
$ acceleration_y <dbl> 8.900, 8.660, 9.546, 10.386, 10.551, 9.207, -9.180, ...
$ acceleration_z <dbl> 3.98697, 4.73730, 1.95133, -0.13647, -1.56011, 2.954...
$ gyro_x         <dbl> 0.0082205, 0.0246293, -0.0068991, 0.0013439, 0.13056...
$ gyro_y         <dbl> 0.0022689, 0.0040279, -0.0150804, -0.3396014, -0.061...
$ gyro_z         <dbl> -0.009966, -0.010858, 0.001122, -0.017956, 0.161530,...
$ second         <dbl> 1362, 257, 973, 902, 820, 533, 556, 200, 115, 824, 2...
$ Speed          <dbl> 0.00000, 0.19000, 0.66706, 7.91329, 20.41941, 19.250...
```

Gambar 3. Informasi struktur *dataset feature*

Sementara itu pada dataset label terdiri dari dua atribut yaitu bookingID dan label (Lihat Tabel II). Artibut label merupakan target dari bookingID yang menunjukan apakah perjalanan aman atau berbahaya.

TABEL II
KETERANGAN ATRIBUT *DATASET LABEL*

No	Nama Atribut	Keterangan
1	bookingID	id perjalanan
2	Label	Kelas target, 1 untuk berbahaya, 0 untuk aman

Jumlah total observasi yang didapatkan dari *dataset* label berjumlah 20.018 dengan tipe data bookingID berupa *double* dan tipe data label berupa *integer* (Lihat Gambar 4).

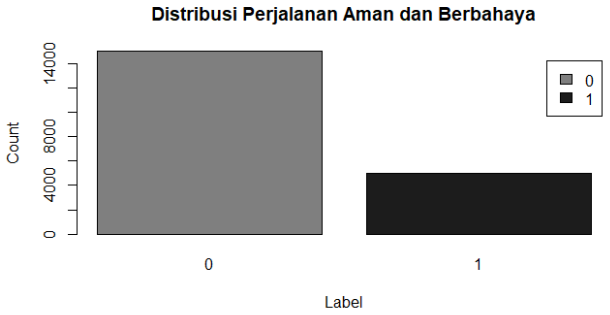
```
Observations: 20,018
Variables: 2
$ bookingID <dbl> 111669149733, 335007449205, 171798691856, 1520418422900, ...
$ label     <int> 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, ...
```

Gambar 4. Informasi struktur *dataset label*

Setiap perjalanan tersebut memiliki data telematik yang terdapat pada dataset *feature*, selanjutnya akan disebut dengan data *point*. Setiap perjalanan memiliki label aman dan berbahaya, tahapan awal pada eksplorasi atribut yaitu adalah mengatahui bagaimana distribusi antara perjalanan aman dan berbahaya dari *dataset*. Tujuannya adalah sebagai dasar pengambilan keputusan pada tahapan selanjutnya.

Pada Gambar 5, distribusi antara perjalanan berbahaya dan aman diketahui dari total keseluruhan data 75% nya merupakan perjalanan aman, sementara 25% nya perjalanan tidak aman atau berbahaya. Berdasarkan hal tersebut, jumlah data pada kelas aman dan berbahaya tidak seimbang atau disebut dengan data *imbalanced*.

Data *imbalanced* merupakan kondisi dimana dalam masalah klasfikasi terdapat kelas-kelas yang tidak terwakili secara sama. Bila tidak ditangani akibatnya prediksi hanya condong kepada satu kelas saja. Akibatnya model yang akan dibuat pada kasus ini akan sulit mengklasifikasikan perjalanan yang berbahaya. Untuk menanganinya pada tahapan *modeling* dapat dilakukan proses *random oversampling* dimana kelas yang jumlahnya sedikit akan diperbanyak hingga jumlahnya sama seperti kelas terbanyak, dengan cara menduplikasi baris data dengan kelas terkecil secara acak.



Gambar 5. Distribusi Perjalanan Aman dan Berbahaya

Setelah mengetahui bagaimana distribusi antara perjalanan aman dan berbahaya kemudian dilakukan eksplorasi secara keseluruhan terhadap *dataset* label. Tujuan eksplorasi ini yaitu mencari tahu apakah terdapat bookingID yang duplikat dan juga memiliki label yang tidak konsisten.

```
Jumlah bookingID yang duplikat :  18
```

A data.frame: 6 × 2

	Var1	Freq
	<fct>	<int>
10	13	2
2124	154618822837	2
3030	223338299461	2
5122	395136991308	2
5239	403726925929	2
5949	455266533495	2

Gambar 6. Label bookingID yang duplikat

Berdasarkan Gambar 6 terdapat 18 bookingID atau perjalanan duplikat (Pada gambar hanya ditampilkan sebagian) dimana frekuensi setiap bookingID yang duplikat berjumlah 2. Setelah melakukan pengecekan secara langsung pada setiap bookingID yang duplikat, masing-masing bookingID yang sama memiliki label yang berbeda atau tidak konsisten.

Setelah mengetahui bagaimana distribusi antara perjalanan aman dan berbahaya selanjutnya adalah mengetahui bagaimana hasil summary statistik. *Summary* statistik atau ringkasan statistik merupakan cara mendapatkan informasi statistik seperti nilai terkecil dan terbesar, kuartil 1 dan 3, mean dan median dari setiap atribut pada dataset, tujuannya untuk memperoleh informasi secara cepat. *Dataset feature* adalah dataset yang akan dilihat hasil *summary* statistiknya karena data tersebut yang akan diolah pada tahapan selanjutnya.

```
   bookingID              Accuracy       Bearing     acceleration_x
 Min.   :           0   Min.   :   1   Min.   :  0   Min.   :-78.42
 1st Qu.: 377957122222   1st Qu.:   4   1st Qu.: 78   1st Qu.: -0.51
 Median : 807453851666   Median :   4   Median :169   Median :  0.06
 Mean   : 818481856668   Mean   :  12   Mean   :169   Mean   :  0.07
 3rd Qu.:1254130450440   3rd Qu.:   8   3rd Qu.:263   3rd Qu.:  0.64
 Max.   :1709396983980   Max.   :6070   Max.   :360   Max.   : 66.87
 acceleration_y   acceleration_z       gyro_x           gyro_y
 Min.   :-72.99   Min.   :-78.45   Min.   :-48.46   Min.   :-74.89
 1st Qu.: -2.02   1st Qu.: -0.93   1st Qu.: -0.03   1st Qu.: -0.03
 Median :  9.08   Median :  0.78   Median :  0.00   Median :  0.00
 Mean   :  4.47   Mean   :  0.89   Mean   :  0.00   Mean   :  0.00
 3rd Qu.:  9.71   3rd Qu.:  2.75   3rd Qu.:  0.02   3rd Qu.:  0.03
 Max.   : 75.06   Max.   : 78.06   Max.   : 39.84   Max.   : 80.31
     gyro_z           second              Speed
 Min.   :-53.55   Min.   :         0   Min.   : -2.00
 1st Qu.: -0.02   1st Qu.:       241   1st Qu.:  1.02
 Median :  0.00   Median :       520   Median :  7.53
 Mean   :  0.00   Mean   :      3803   Mean   :  9.01
 3rd Qu.:  0.02   3rd Qu.:       863   3rd Qu.: 15.48
 Max.   : 66.30   Max.   :1495796757   Max.   :148.02
```

Gambar 7. Ringkasan Statistik Atribut *dataset feature*

Dari hasil *summary* pada Gambar 7 ditemukan beberapa informasi menarik yaitu pada atribut *accuracy*. Atribut *acuracy* dapat disebut tepat ketika nilainya 1, semakin besar nilainya maka semaikn buruk akurasinya. Dari informasi summary dibawah terdapat nilai akurasi terbesar yaitu 6070 sementara rataan nya hanya 12, dari data tersebut diindikasikan terdapat data perjalanan yang akurasinya sangat buruk dan bisa saja berdampak pada atribut lainnya karena pengumpulan data diperngaruhi oleh akurasi pada gps.

Atribut *Bearing* atau GPS Bearing, merupakan arah kompas dari posisi saat ini ke arah tujuan. GPS *Bearing* memiliki cakupan ke segala arah atau 0-360 derajat. Setiap data point pergerakan perjalanan, nilainya akan selalu berubah sesuai kondisi perjalanan, namun tujuannya tetap pada suatu titik. Pada hasil ringkasan statistik diatas, tidak ditemukan anomali pada atribut ini.

Sementara itu atribut percepatan yaitu *acceleration_x*, *acceleration_y*, dan *acceleration_z* tidak ditemukan anomali. X,Y, dan Z merupakan sumbu, karena kendaran berjalan berfokus pada satu sumbu saja kemungkinan hanya satu sumbu yang nilainya akan berubah-ubah dalam perjalanan. Pada atribut percepatan ditemukan percepatan bernilai negatif tetapi hal tersebut bukan berarti kesalahan pada pencatatan, namun berarti terjadi perlambatan.

Selain percepatan terdapat juga atribut *gyro* yang memiliki sumbu x, y dan z. Atribut ini merupakan singkatan dari *gyroscope*, dimana menampilkan hasil pengukuran orientasi setiap data point. Dari data diatas diketahui bahwa mean dan mediannya berada disekitaran nilai nol, hal ini karena berarti kebanyakan data point yang tercatat kondisi smartphone yang digunakan dalam pencatatan stabil, akan tidak stabil jika nilainya menjauhi nol diasumsikan terjadi guncangan, atau bisa saja disebabkan tanjakan atau jalan yang tidak rata

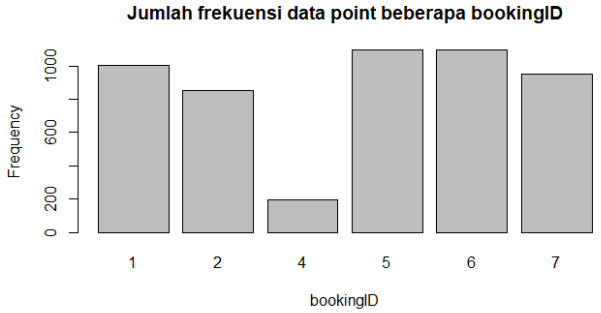
Kemudian pada atribut *second* didapatkan nilai yang sangat besar, hal ini diindikasikan bahwa terdapat perjalanan yang memiliki catatan waktu yang sangat besar hingga melebihi 10^9, berdasarkan hal tersebut harus diteliti lagi apakah terdapat perjalanan yang sangat panjang atau bisa saja terjadi kesalahan saat pengumpulan dan juga apakah ada korelasinya dengan akurasi.

Selain itu pada atribut *Speed* atau kecepatan terdapat hal yang menarik yaitu nilai minimunya adalah -2 yang berarti terdapat perjalanan yang memiliki data point Speed dengan nilai negatif sementara seharusnya kecepatan tidak dapat bernilai negatif. Hal ini diasumsikan bahwa terjadi kesalahan pada pengumpulan data dari aplikasi.

Setelah mengetahui bagaimana *summary* statistik dari setiap atribut pada *dataset feature*. Tahapan selanjutnya adalah mengetahui tentang data setiap perjalanan. Seperti yang dijelaskan sebelumnya bahwa *dataset feature* berisi data point pergerakan setiap perjalanan. Oleh karena itu perlu diketahui apakah data *point* perjalanan yang ada pada *dataset* memiliki jumlah yang sama dengan melihat frekuensi munculnya beberapa sampel perjalanan/bookingID pada *dataset feature*.

Gambar 8 menunjukkan grafik frekuensi munculnya sampel beberapa bookingID pada *dataset feature*. Dari hasil

gambar tersebut diketahui bahwa setiap perjalanan memiliki data point yang berbeda. Perbedaan data point terjadi diasumsikan karena setiap perjalanan memiliki jarak tempuh dan waktu yang berbeda. Namun pada *dataset* tidak ditemukan atribut jarak, namun ditemukan atribut waktu yaitu *second*. Oleh karena itu perbedaan tersebut bisa terjadi karena lama perjalanan masing-masing bookingID berbeda.



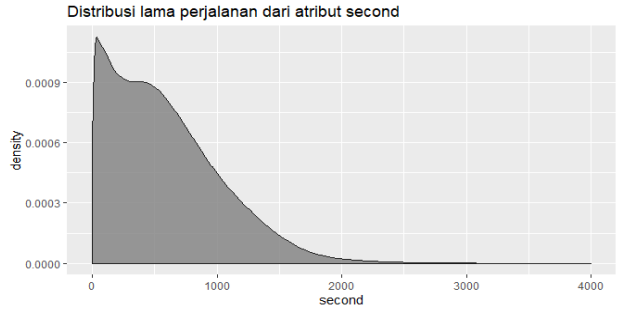
Gambar 8. Frekuensi beberapa bookingID pada dataset Feature

Pada pembahasan sebelumnya ditemukan terdapat anomali pada atribut *second*, maka akan dilakukan eksplorasi lebih lanjut dengan mengambil sampel satu bookingID kemudian menampilkan data *point* nya. Kemudian dilakukan pengurutan dari nilai terkecil hingga terbesar lalu ditampilkan 6 teratas sehingga hasilnya bisa dilihat pada Gambar 9.

bookingID	Accuracy	Bearing	acceleration_x	acceleration_y	acceleration_z	gyro_x	gyro_y	gyro_z	second	Speed
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
335007449205	3.309	205	1.1181	9.625	1.9896	-0.007551	0.257499	0.0050238	0	2.1743
335007449205	3.109	191	0.8859	9.608	-0.5315	0.026047	0.099285	0.0215171	1	1.4101
335007449205	3.000	185	0.8572	9.670	0.8260	-0.004497	-0.008227	-0.0041392	2	0.5638
335007449205	3.000	185	0.8859	9.608	0.7255	-0.002664	0.017429	-0.0090261	3	0.0000
335007449205	3.000	185	0.8643	9.357	0.9673	-0.006329	0.010710	0.0007477	4	0.0000
335007449205	3.000	185	0.8643	9.335	0.8979	-0.005718	-0.002118	-0.0029175	5	0.0000

Gambar 9. Sampel bookingID diurutkan dari terkecil

Dari sampel perjalanan yang diambil dan ditampilkan diatas, setelah diurutkan diketahui bahwa second dimulai dari 0 hingga perjalanan selesai, artinya semakin banyak *second* semakin banyak pula data point. Bisa disebut bahwa *second* sangat berhubungan dengan jumlah data *point* setiap perjalan. Data seperti ini bisa disebut dengan *time-series*.



Gambar 10. Distribusi atribut *second* < 10000

Setelah mengetahui bahwa atribut second berbentuk *time-series*, berdasarkan hasil *summary* statistik sebelumnya didapatkan informasi bahwa terdapat perjalanan dengan nilai second yang sangat besar mencapai 10^9. Itu berarti asumsinya terdapat perjalanan yang jika dikalkulasi lama perjalanannya mencapai 415 ribu jam perjalanan. Oleh karena itu diperlukan informasi bagaimana distribusi pada atribut *second*.

Berdasarkan Gambar 10 diketahui bahwa perjalanan rata-rata berkisar pada 0 sampai 1500 detik. Sementara itu diketahui dari informasi *summary* statistik terdapat perjalanan yang nilai detiknya lebih dari 109. Oleh karena itu proses eksplorasi ini dilakukan untuk mencari tahu data point yang nilai detiknya diatas 10000 detik.

	bookingID	freq
	<dbl>	<int>
1	858993459333	44
2	1108101562401	3
3	1434519076976	2
4	1460288880770	32
5	1503238553722	1

Gambar 11. Beberapa data point perjalanan second-nya melebihi 10000 detik

Dari proses yang dilakukan ditemukan total 82 data point (Gambar 11) yang nilai secondnya diatas 10000 detik, terbagi atas 5 bookingID yang jumlah masing-masingnya berbeda. Setelah dilakukan pengecekan manual, ditemukan bahwa 82 data *point* tersebut berada di akhir masing-masing perjalanan/bookingID. Pada Gambar 12 memperlihatkan second yang bernilai sangat besar dan terjadi perbedaan signiffikan dengan baris diatasnya. Sementara itu seharusnya *second* berarti waktu dalam detik yang dihitung dari mulai perjalanan hingga selesai. Namun terdapat nilai pada atribut *second* yang sangat tinggi, hal ini diindikasikan bisa saja aplikasi mengalami terjadi *error/bug* saat pencatatan pergerakan ke *server*. Oleh karena tidak konsisten, datanya berada diakhir perjalanan dan jumlahnya tidak banyak maka data ini akan dihapus pada tahapan *modify*.

bookingID	Accuracy	Bearing	acceleration_x	acceleration_y	acceleration_z	gyro_x	gyro_y	gyro_z	second	Speed
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1460288880770	5.000	283	0.80445	9.462	1.07260	0.20647	0.101404	0.048869	1407	9.423
1460288880770	5.000	284	0.11492	9.807	1.18752	-0.18815	0.000000	-0.116064	1408	9.680
1460288880770	5.000	285	0.65122	9.539	0.15323	0.03299	-0.682947	-0.035430	1409	9.939
1460288880770	4.006	0	0.57461	9.653	1.03430	-0.08674	-0.201586	-0.026878	619315319	-1.000
1460288880770	4.306	0	0.76614	9.653	0.65122	-0.01833	-0.074526	0.014661	619315340	-1.000
1460288880770	4.322	0	0.68953	9.117	0.11492	-0.07575	0.010996	-0.018326	619315361	-1.000

Gambar 12. Sampel perjalanan data point second > 10000

B. Modify

Pada tahapan ini dilakukan proses integrasi, dikarena kan data masih terpish oleh karena itu harus disatukan terlebih dahulu ke dalam satu file berekstensi *feather* agar data lebih cepat di proses oleh kode program.

Pada tahapan *cleaning* berdasarkan anomali yang ditemukan pada tahapan *explore* anomali seperti data yang duplikat dan tidak konsisten, serta data perjalanan yang memiliki data *point* dengan nilai *second* lebih dari 10000 detik akan dihapus.

Setelah melakukan tahapan pembersihan data kemudian dilakukan persiapan data yang bertujuan untuk mempercepat proses *feature engineering*. Seperti yang sudah dijelaskan pada bab sebelumnya terdapat 3 jenis feature yang akan digunakan yaitu *statistical feature*, *outlier feature* dan rds. Untuk mendapatkan nilai rds seperti yang dijelaskan yaitu jumlah dari kecepatan dikalikan dengan jumlah percepatan total, kemudian, membagi hasilnya dengan jarak setiap perjalanan. Sementara itu nilai percepatan pada dataset terdiri atas 3 sumbu yaitu x, y dan z.

Secara umum kendaraan bergerak ke satu arah hanya satu sumbu yang akan menunjukkan variasi cukup besar sementara sumbu yang lain variasinya cukup kecil. Dengan mengasumsikan bahwa pembacaan *accelerometer* diwakili dalam vektor, maka ketiga sumbu tersebut dapat diringkas dengan rumus 3.1. Dimana akan dibuat sebuah atribut baru dengan nama *acceleration*, setiap nilai *accelerometer* x,y,z dari data *point* akan dihitung dan disimpan hasilnya ke atribut yang baru. Atribut hasil dari perhitungan beberapa atribut dasar ini disebut dengan atribut kombinasi.

Setelah melalui proses *integration*, data *cleaning* dan data *preparing* jumlah data pada *dataset feature* adalah 16.116.622 dengan 13 atribut. Sementara pada dataset label, jumlah perjalanan yang tersisa yaitu 19.982.

1) *Statistical Feature*: Model yang nantinya akan dibuat adalah model yang dapat digunakan untuk membedakan perjalanan yang aman dan berbahaya. Salah satu *feature* yang biasa digunakan untuk proses feature engineering ini adalah *statistical feature*. *Feature* ini mencoba untuk mengambil nilai statistik seperti mean, median, standar deviasi, dan quartil. Pada penelitian ini akan digunakan 4 informasi statistik yaitu mean, standar deviasi, kuartil 1 dan 3. Penggunaan keempat nilai tersebut yaitu karena dianggap sudah cukup untuk mencerminkan bagaimana pesebaran nilainya. Mekanisme pembuatan *feature* dengan cara membaca setiap bookingID pada dataset label kemudian bookingID tersebut akan dicari data point nya dalam dataset, dimana setiap atribut akan dihitung dan dicari nilai informasi statistik yang dibutuhkan. Sehingga nilai atribut untuk setiap data *point* hanya akan diwakili oleh 4 nilai (Lihat Tabel III).

TABEL III
KETERANGAN ATRIBUT *DATASET LABEL*

No	Nama Atribut	Hasil *Statistical Feature*
1	Accuracy	Accuracy_Mean, Accuracy_Std, Accuracy_Q1, Accuracy_Q3
2	Bearing	Bearing_Mean, Bearing_Std, Bearing_Q1, Bearing_Q3
3	acceleration_x	acceleration_x_Mean, acceleration_x_Std, acceleration_x_Q1, acceleration_x_Q3
4	acceleration_y	acceleration_y_Mean, acceleration_y_Std, acceleration_y_Q1, acceleration_y_Q3
5	acceleration_z	acceleration_z_Mean, acceleration_z_Std, acceleration_z_Q1, acceleration_z_Q3
6	gyro_x	gyro_x_Mean, gyro_x_Std, gyro_x_Q1, gyro_x_Q3
7	gyro_y	gyro_y_Mean, gyro_y_Std, gyro_y_Q1, gyro_y_Q3
8	gyro_z	gyro_z_Mean, gyro_z_Std, gyro_z_Q1, gyro_z_Q3
9	Speed	Speed_Mean, Speed_Std, Speed_Q1, Speed_Q3

2) *Jumlah Perilaku Pengemudi diluar Outlier*: Pembuatan *feature* ini dilakukan karena diasumsikan bahwa dalam sebuah perjalanan sebuah kendaraan bisa saja memasuki area berbeda dimana komponen seperti kecepatan, percepatan dan orientasi geraknya bisa saja berbeda-beda. Contohnya batas minimum dan maksimum kecepatan didaerah kota dengan jalan tol berbeda. Di jalanan kota biasanya kecepatan minimum 40 Km perjam, sementara ketika berada di jalan tol dianggap kecepatan tersebut bisa saja membahayakan baik kendaraan tersebut ataupun kendaraan dibelakang. Jika hanya diwakilkan dengan *statistical feature* hal tersebut dianggap tidak baik karena menyamaratakan perjalananya maka digunakanlah feature ini untuk mengakomodasi hal tersebut. *Feature* ini merupakan proses menghitung berapa banyak aktivitas pengemudi yang tercatat diluar *outlier*. Aktivitas perilaku yang dimaksud yaitu kegiatan yang dilakukan pengemudi yaitu berkaitan dengan percepatan, *gryroscope* dan kecepatan. Dimana atribut yang digunakan adalah *acceleration* pada sumbu x,y dan z, kemudian *gyro* pada sumbu x, y dan z serta *speed*. Dari atribut yang telah ditentukan sebelumnya, pada setiap perjalanan/bookingID akan dihitung nilai Q1 dan Q3 nya. Kemudian dilakukan penjumlahan berapa data *point* yang berada diluar nilai tersebut yaitu kurang dari Q1 dan atau lebih dari Q3. Jika diatas Q3 maka akan diberi nama *over_* lalu diikuti dengan nama atribut yang digunakan, sedangkan untuk dibawah Q1 akan diberi nama *under_* dan diikuti dengan nama atribut. Hasilnya tertera pada Tabel IV.

TABEL IV
ATTRIBUT HASIL *FEATURE ENFINEERING* JUMLAH PERILAKU PENGEMUDI DILUAR *OUTLIER*

No	Atribut Asal	Hasil Pengolahan *Feature*
1	acceleration_x	under_acceleration_x, over_acceleration_x
2	acceleration_y	under_acceleration_y, over_acceleration_y
3	acceleration_z	under_acceleration_z, over_acceleration_z
4	gyro_x	under_gyro_x, over_gyro_x
5	gyro_y	under_gyro_y, over_gyro_y
6	gyro_z	under_gyro_z, over_gyro_z
7	Speed	under_speed, over_speed

3) *Risk Driving Score*: *Feature* terakhir yang digunakan yaitu Risk Driving Score, *feature* ini adalah adaptasi dari yang dilakukan oleh Winitthumkul [12]. *Feature* ini merupakan jumlah dari jumlah kecepatan dikalikan dengan jumlah percepatan total, kemudian, membagi hasilnya dengan jarak setiap perjalanannya. Namun percepatan pada dataset yang didapatkan terdiri atas 3 sumbu sehingga akan diringkas menggunakan rumus 3. Selain dikarenakan jarak tidak diketahui pada *dataset*, oleh karena itu untuk mendapatkannya akan dilakukan proses perhitungan terlebih dahulu. Setiap data point memiliki kecepatan dan waktu/second. Secara rumus untuk memperoleh jarak dapat dilakukan dengan mengalikan waktu dengan kecepatan. Oleh karena itu kecepatan dan *second* setiap data point dalam perjalanan akan dikalikan kemudian dijumlahkan keseluruhannya, maka akan didapatkan jarak perjalanan dapat dilihat pada Rumus 4. Kemudian untuk mendapatkan rds, hasil dari penjumlahan perkalian antara kecepatan dan percepatan setiap data point sebelumnya akan dibagi dengan jarak yang diperoleh dapat dilihat pada Rumus 5. Semakin besar nilai dari perhitungan RDS dalam satu perjalanan, perjalanaan tersebut semakin beresiko. Sementara itu kebalikannya semakin kecil nilai RDS maka perjalanan semakin tidak beresiko.

$$a_i = \sqrt{acceleration\_x_i{}^2 + acceleration\_y_i{}^2 + acceleration\_z_i{}^2} \quad (3)$$

Dimana:
- a_i= Percepatan

$$S = \Sigma(v_i \times t_i) \quad (4)$$

Dimana:
- a_i= Percepatan
- S = Jarak dalam 1 perjalanan (m)
- v = Kecepatan tiap data *point*
- t = Waktu/*Second*
- i = id data *point*

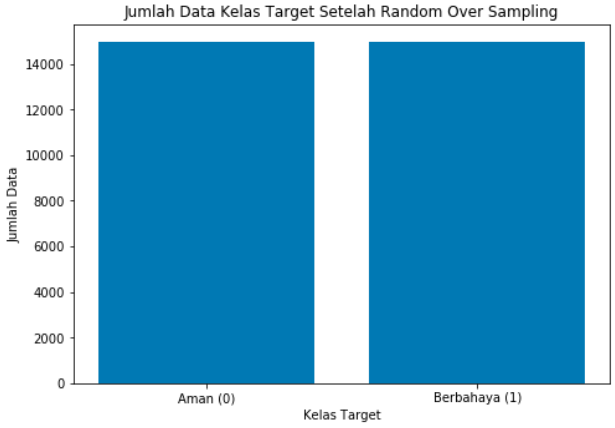
$$RDS = \frac{\sum_{i=1,2,3}^{n}(v_i \times a_i)}{S} \quad (5)$$

Dimana :
- v : Kecepatan tiap data *point* (m/s)
- a : Percepatan tiap data *point* (m/s)
- S : Jarak dalam 1 perjalanan (m)
- i : id data point

C. Model

Dari tahapan *feature engineering* yang telah dilakukan sebelumnya didapatkan total 51 *feature*. Sebelum masuk ke tahap *model*, semua *feature* tersebut akan digabungkan ke dalam satu *dataframe* beserta dengan kelas target.

Berdasarkan hasil eksplorasi sebelumnya diketahui bahwa dataset memiliki kelas target yaitu 1 untuk berbahaya dan 0 untuk aman. Dimana jumlahnya 75% untuk perjalanan aman dan 25% perjalanan berbahaya. Data ini dianggap tidak seimbang atau *imbalanced*. Oleh karena itu akan dilakukan proses *random over sampling*. Dimana perjalanan yang berbahaya akan diambil secara acak untuk diduplikat sehingga nantinya jumlah data perjalanan berbahaya jumlahnya akan sama dengan data perjalanan yang aman dapat dilihat Gambar 13. Jumlah total data setelah dilakukan *random over sampling* yaitu 29.996.



Gambar 13. Jumlah Data Kelas Target Setelah Random Over Sampling

Setelah itu dilakukan pemisahan antara data *feature* dan target/label. Karena nilai pada data cukup beragam untuk mempercepat proses, terlebih dahulu dilakukan proses *scaling* pada setiap *feature*. Proses *scaling* ini menggunakan *library* dari sckit-learn bernama *MinMaxScaler* dimana proses perhitungannya diperoleh dengan cara berikut.

$$m = \frac{(X - X_{min})}{(X_{max} - X_{min})} \quad (6)$$

Dimana:
- m merupakan nilai hasil *scaling*
- x merupakan nilai asal
- *Xmax* nilai terbesar dari kolom/*feature*
- *Xmin* nilai terkecil dari kolom/*feature*

Setelah melakukan *scaling* terhadap data, kemudian tahapan selanjutnya yaitu *training*. Sebelum melakukan training terhadap data, data akan di split menggunakan *K-Fold Cross Validation* dengan K = 5 untuk spliting data. Jumlah data setiap *fold* tertera pada Tabel V.

TABEL V
JUMLAH DATA SETIAP *FOLD*

Fold	Jumlah Data
1	6000
2	5999
3	5999
4	5999
5	5999

Kemudian dilakukan *training* terhadap data. Pengujian dilakukan dengan mengganti nilai parameter C dengan beberapa nilai yang ditentukan yaitu 0,001, 0,01, 0,1, 1, 10, 50 dan 100. Sementara itu untuk nilai yang lain dibiarkan secara default.

Seperti yang diketahui bahwa algoritma yang digunakan pada penelitian ini yaitu *Support Vector Machine* dimana kernel yang digunakan yaitu RBF. Kernel ini digunakan karena pada penelitian-penelitian sebelumnya seperti yang dilakukan Arif dkk [13], hasilnya terbilang sangat baik dan juga komputasinya cukup cepat jika dibandingan dengan kernel linier.

D. Assess

Setelah melakukan proses training, pada tahapan assess akan dilakukan proses penilaian terhadap model yang sudah dibuat. Penilaian dilakukan dengan melihat dan membandingkan hasil confussion matrix, akurasi (Tabel VI). Kemudian juga akan digunakan kurva ROC dan skor AUC-ROC (Gambar 14 - 20) untuk menilai model. Penilaian menggunakan AUC-ROC sangat baik untuk melihat seberapa baik model mampu membedakan antar kelas.

Kurva ROC diperoleh dari *false positive rate* dan *true positive rate*. Untuk mendapatkannya digunakan fungsi roc_*curve* dari *library* sklearn dimana parameter yang dibutukan adalah y_true dan y_score. Parameter y_true yaitu merupakan target kelas data testing nilainya berupa 0 dan 1. Sementara y_score merupakan hasil prediksi probabilitas positif terhadap data testing. Nilai inilah yang akan digunakan untuk mendapatkan false positive rate dan true positive rate dari setiap *threshold*. Baru kemudian semua hasilnya diplot kedalam kurva dimana true positif rate menjadi sumbu y dan false positif rate merupakan sumbu x.

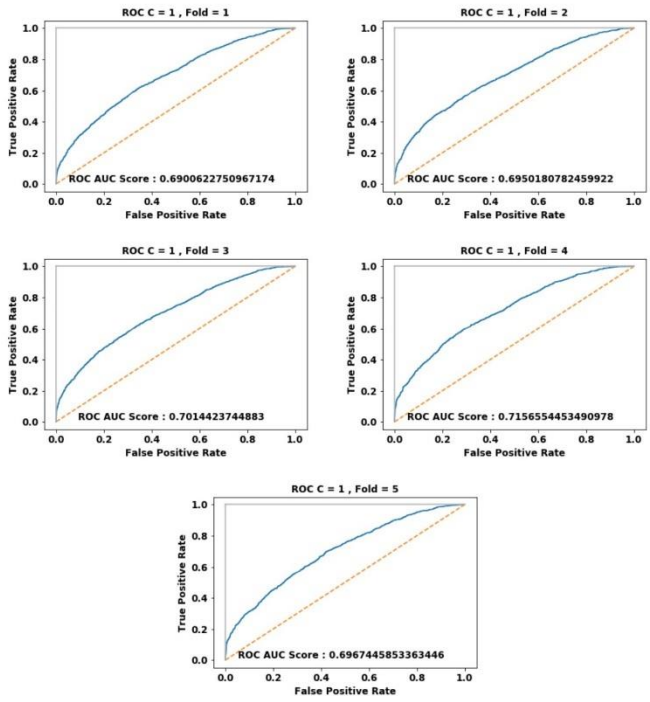
Sementara itu untuk memperoleh skor ROC-AUC, sama seperti fungsi roc_curve dibutuhkan nilai y_true dan y_score. Namun keluarannya merupakan perhitungan area dibawah kurva ROC dari nilai prediksi y_score. Tabel VI adalah hasil evaluasi confussion matrix dan akurasi, yang didapatkan untuk setiap *fold* dari setiap nilai C.

TABEL VI
TABEL HASIL *CONFUSSION MATRIX* DAN AKURASI SETIAP NILAI C

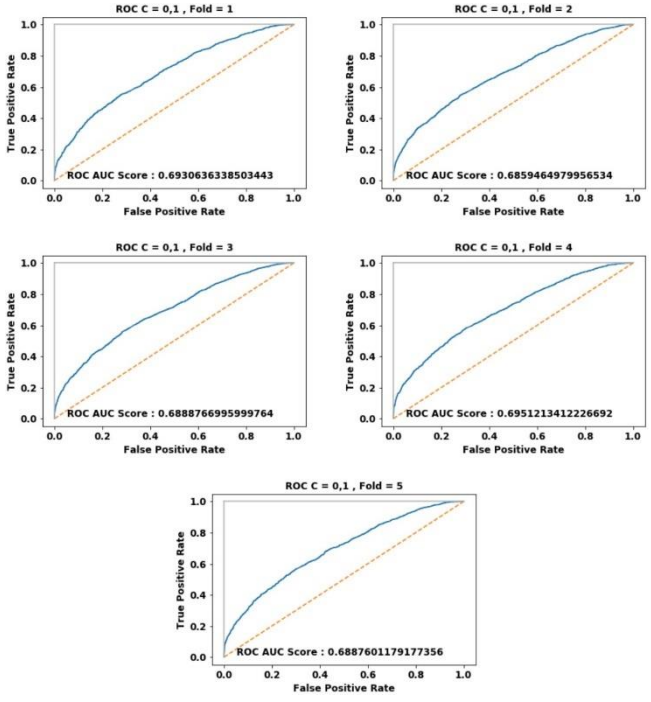
C	Fold	TN	FP	FN	TP	Akurasi
1	1	2243	781	1385	1591	63,90 %
	2	2228	795	1392	1584	63,54 %
	3	2215	779	1407	1598	63,56 %
	4	2269	754	1390	1586	64,20 %
	5	2202	732	1485	1580	63,04 %
0,1	1	2197	827	1355	1621	63,63 %
	2	2205	818	1399	1577	63,04 %
	3	2167	827	1371	1634	63,36 %
	4	2217	806	1367	1609	63,77 %
	5	2145	788	1446	1619	62,76 %
0,01	1	2215	809	1400	1576	63,18 %
	2	2230	793	1446	1530	62,67 %
	3	2209	785	1419	1586	63,26 %
	4	2230	793	1414	1562	63,21 %
	5	2198	736	1524	1541	62,32 %
0,001	1	0	3024	0	2976	49,60 %
	2	0	3023	0	2976	49,60 %
	3	2994	0	3005	0	49,90 %
	4	0	3023	0	2976	49,60 %
	5	2934	0	3065	0	48,90 %
10	1	2236	788	1333	1643	64,65 %
	2	2189	834	1308	1668	64,29 %
	3	2180	814	1348	1657	63,96 %
	4	2256	767	1352	1624	64,67 %
	5	2195	739	1441	1624	63,66 %
50	1	2237	787	1281	1695	63,53 %
	2	2196	827	1284	1692	64,81 %
	3	2159	835	1303	1702	64,36 %
	4	2201	822	1298	1678	64,66 %
	5	2169	765	1421	1644	63,56 %
100	1	2238	786	1276	1700	65,63 %
	2	2187	836	1249	1727	65,24 %
	3	2159	835	1284	1721	64,67 %
	4	2190	833	1243	1733	65,39 %
	5	2157	777	1385	1680	63,96 %

Berdasarkan Tabel VI, untuk nilai C = 1, diperoleh akurasi terbaik yaitu 63,90% pada *fold* ke 1. Untuk C = 0,1 diperoleh akurasi terbaik 64,77% tidak jauh berbeda dengan nilai akurasi pada C = 0,01 yaitu 63,26. Pada nilai C = 0,001 nilai akurasi sangat buruk yaitu dibawah 50% untuk setiap *fold* nya. Sementara itu untuk nilai C = 10, 50 dan 100 memiliki nilai akurasi terbaik setiap foldnya yaitu 64,67 %, 64,81 % dan 65,63 %.

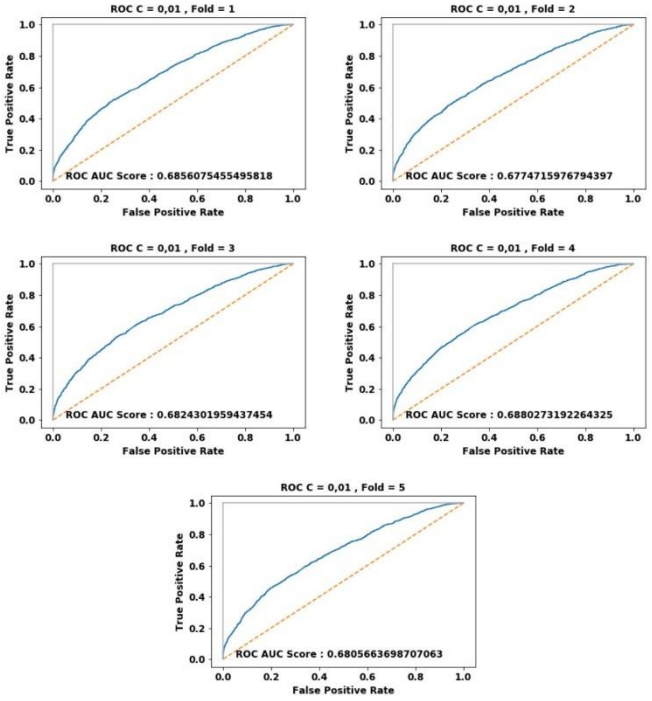
Sementara itu jika melihat hasil *confussion matrix*, untuk setiap nilai C lebih banyak perjalanan aman yang terprediksi dengan benar dibandingkan dengan perjalanan berbahaya yang terprediksi dengan benar. Oleh karena itu *model* masih belum meyakinkan, karena tujuan utamanya adalah memprediksi perjalanan yang berbahaya. Untuk mengetahui apakah model yang dibuat sudah mampu memisahkan antara dua kelas maka digunakanlah ROC/AUC *Score*.



Gambar 14. Kurva ROC dengan C = 1

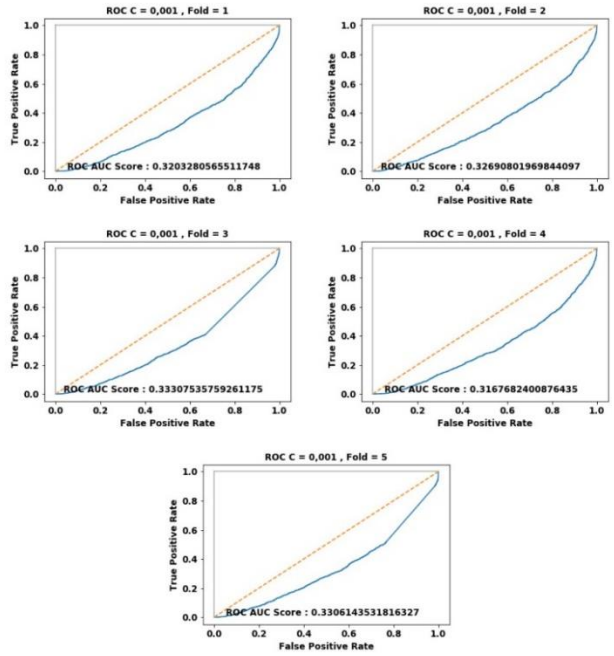


Gambar 15. Kurva ROC dengan C = 0,1



Gambar 16. Kurva ROC dengan C = 0,01

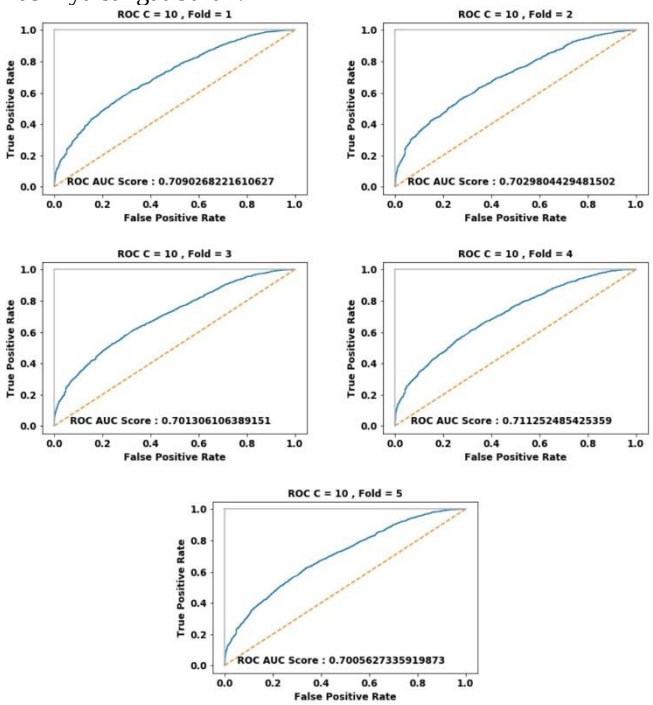
Jika kita melihat kurva ROC untuk nilai C = 1, C = 0,1 dan C = 0,01 (Gambar 14-16). Kurva mengarah ke arah sudut kiri atas ke arah *perfect classifier* dan menjauhi garis *random classifier* (garis putus-putus). *Random classifier* merupakan garis diagonal yang merepresentasikan 50% *true positive rate* dan *false positive rate* [14] sehingga ketika kurva mengarah ke sudut kiri atas artinya kualitas *model* lebih baik.



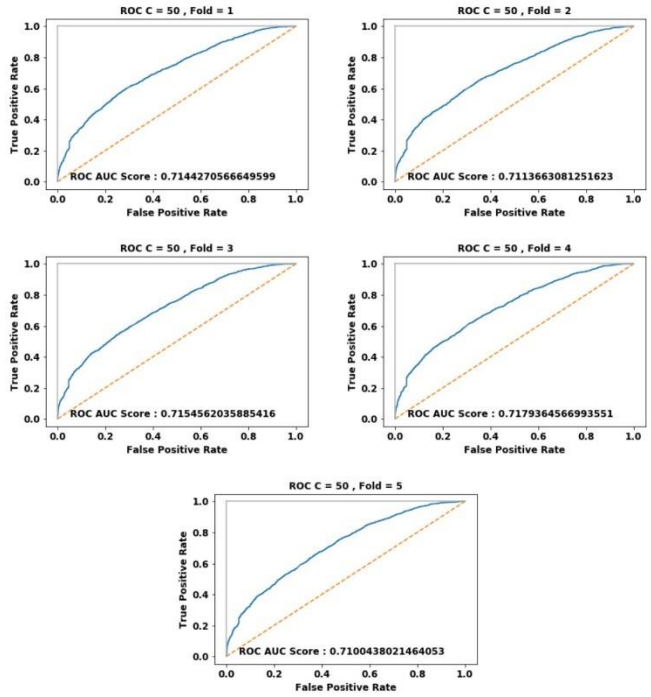
p-ISSN : 2443-2210
e-ISSN : 2443-2229

Gambar 17. Kurva ROC dengan C = 0,001

Sementara itu kurva dengan nilai = 0,001, bentuk kurva mengarah ke sudut kanan bawah yang berarti hasil *model* kurang baik. Hal ini sejalan dengan hasil akurasi dan *conffusion matrix* (Tabe; VI) dimana pada nilai C = 0,001 hasilnya sangat buruk.

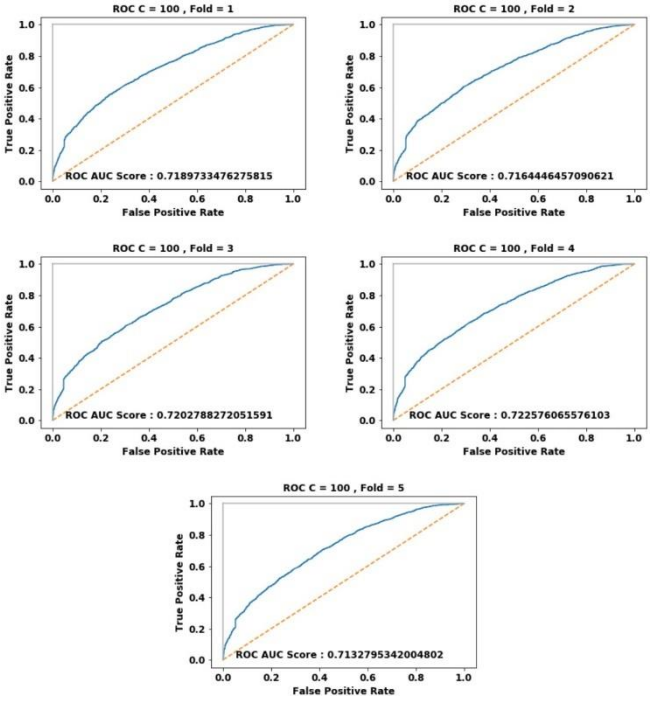


Gambar 18. Kurva ROC dengan C = 10



Gambar 19. Kurva ROC dengan C = 50

Jika kita melihat kurva dengan nilai C = 1, C = 0,1, C = 0,01, C = 10, C = 50 dan C = 100, bentuknya hampir sama, tidak jauh berbeda. Untuk menentukan mana *model* terbaik selain dari bentuk kurva, juga dapat digunakan skor AUC, dimana akan didapatkan hasil yang lebih detail lagi yaitu C = 100 merupakan *model* dengan skor AUC terbaik.



Gambar 20. Kurva ROC dengan C = 100

Setelah melakukan pengujian menggunakan *cross validation* untuk setiap nilai C yang diubah, hasilnya mempengaruhi akurasi setiap fold. Dari 5 *fold* yang diujikan untuk setiap nilai C, rata-rata hasil akurasi terbaik yaitu pada *fold* ke 4. Sementara itu jika melihat keseluruhan hasil akurasi terbaik yang didapatkan yaitu 65,63% yaitu dengan C = 100 pada *fold* ke 1.

Jika melihat dari penelitian serupa [9], pada penelitian tersebut kelas targetnya adalah apakah pengemudi ber-*gender* laki-laki atau perempuan, pada penelitian tersebut digunakan skor ROC-AUC sebagai salah satu penilaian algoritma yang digunakan. Hasilnya nilai AUC/skor ROC-AUC nya bernilai 71.67. Jika dibandingkan dengan nilai AUC pada penelitian ini, skor ROC-AUC terbaik yaitu 71.79. Hasilnya tidak jauh beda bahkan lebih baik. Selain itu jika membandingkan jumlah observasi datanya, penelitian ini memiliki data yang lebih banyak.

Berdasarkan dari pengujian yang telah dilakukan skor ROC-AUC dari setiap nilai C yang diubah, nilai terbaik konsisten pada *fold* 4. Hal ini menyatakan bahwa akurasi dan skor ROC-AUC tidak sama dalam menguji model. Akurasi hanya melihat dari perjalanan yang diprediksi dengan benar kemudian dibagi total perjalanan yang diprediksi. Sementara skor ROC-AUC menilai lebih baik karena membandingkan antara kelas target dengan

probabilitas positif kelas target. Kemudian dihitung nilai *false positive rate* dan *true positive rate* untuk setiap *threshold*-nya.

Dari hasil penelitian ditemukan bahwa hasil klasifikasi menggunakan SVM terburuk pada penelitian ini yaitu pada nilai C = 0,001 (Gambar 17), dimana kurva ROC setiap *fold* nya berada di bawah garis random classifier/mengarah ke sudut kanan bawah kurva. Selain itu rata-rata akurasinya dibawah 50% dan kelas model hanya bisa memprediksi satu kelas saja.

Selain itu hal menarik yang dapat disimpulkan yaitu nilai C mempengaruhi besar atau kecilnya nilai akurasi, skor ROC-AUC bahkan kurva ROC. Semakin kecil nilai C maka semakin kecil juga hasil akurasi dan juga skor ROC-AUC yang didapatkan. Begitu juga sebaliknya semakin besar nilai C, semakin besar juga nilai akurasi dan skor ROC-AUC nya. Nilai C disini berfungsi untuk mengontrol trade off antara antara *margin* dan *error* klasifikasi [13], dan juga mengklasifikasikan data pelatihan dengan benar. Semakin besar nilai C, *erro*r klasifikasi yang didapatkan semakin kecil sehingga akurasi semakin mengingkat.

VI. KESIMPULAN

Berdasartkan dataset yang didapatkan diketahui bahwa dalam satu perjalanan terdapat banyak sekali data point. *Feature Engineering* dilakukan untuk menyederhanakan banyaknya data telematik dalam satu perjalanan sehingga diwakilkan dalam beberapa *feature.* Pada penelitian ini berhasil dibuat 3 jenis feature yaitu *statistical feature*, jumlah perilaku pengemudi diluar outlier dan nilai *risk driving score*. *Statistical feature* diperoleh dari nilai mean, standar deviasi, kuartil 1 dan 3. Nilai *statistical feature* sangat baik untuk mengetahui bagaimana pesebaran data seperti kecepatan, percepatan dll dalam satu perjalanan, namun dalam suatu perjalanan sebuah kendaraan bisa saja memasuki area berbeda dimana komponen seperti kecepatan, percepatan dan orientasi geraknya bisa saja berbeda-beda. Oleh karena itu dibutuhkan *feature* jumlah perilaku pengemudi diluar *outlier*, perilaku yang dimaksud yaitu berkaitan dengan percepatan, *gryroscope* dan kecepatan. Sementara itu untuk feature RDS merupakan *feature* yang merupakan implementasi dari penelitian sebelumnya dimana semakin besar nilai RDS dapat menyatakan perjalanan tersebut semakin beresiko atau tidaknya.

Implementasi algoritma SVM pada penelitian ini dilakukan dengan menggunakan kernel RBF, dimana nilai parameter C yang akan diamati hasilnya. Selain itu juga sebelumnya dilakukan split data menggunakan *K-Fold Cross Validation*, dimana didapatkan hasil yang cukup baik dalam akurasi. Dari hasil tahapan assess didapatkan model dengan akurasi terbaik yaitu dengan parameter C = 100, pada fold 1 dengan akurasi 65,63%. Sementara itu yang menarik adalah pada *fold* 4 setiap parameter C yang diubah, nilai ROC-AUC konsisten menjadi yang terbaik dari percobaan yang telah dilakukan.

Pada penelitian ini juga didapatkan bahwa parameter C mempengaruhi hasil akurasi, skor ROC-AUC dan juga kurva ROC. Berdasarkan penelitian [9] skor ROC-AUC merupakan salah satu komponen penilaian model, jika dibandingkan dengan penelitian ini, penelitian yang telah dilakukan lebih baik, selain itu jumlah data yang digunakan lebih banyak.

DAFTAR PUSTAKA

[1] J. Davis. (2018) The Real Story Behind Uber's Exit from Southeast Asia *INSEAD Knowledge*. [Online]. Tersedia : https://knowledge.insead.edu/entrepreneurship/the-real-story-behind-ubers-exit-from-southeast-asia-10096

[2] A. Venkatesan. (2018) How GOJEK Manages 1 Million Drivers With 12 Engineers (Part 2). [Online]. Tersedia : https://blog.gojekengineering.com/how-go-jek-manages-1-million-drivers-with-12-engineers-part-2-35f6a27a0faf

[3] H. Ghadially. (2018) How we do driver ratings at GO-JEK - Gojek Product Tech *Gojek Product Tech*. [Online]. Tersedia : https://blog.gojekengineering.com/driver-ratings-338c2f3ffeb2

[4] Grab Aiforesa.com. (2019). AI For SEA: Safety. [Online]. Tersedia : https://www.kaggle.com/dataset/1e0ae2b890510f8f2540277583a52b4ced5ae5dfb3c4be71ccd456a7b85c2512.

[5] F. Nargesian, H. Samulowitz, U. Khurana, E. B. Khalil, & D. Turaga, "Learning feature engineering for classification," *Prosiding IJCAI'17,*2017, p. 2529.

[6] T. Osafune, T. Takahashi, N. Kiyama, T. Sobue, H. Yamaguchi, dan T. Higashino, "Analysis of Accident Risks from Driving Behaviors," *Int. J. ITS Res.*, vol. 15, no. 3, pp. 192–202, 2017.

[7] A. Zheng dan A. Casari, *Feature Engineering for Machine Learning*. Sebastopol: O'Reilly Media, Inc., 2018.

[8] E. Prasetyo, *Data Mining Mengolah Data Menjadi Informasi Menggunakan Matlab*. Yogyakarta: Andi Offset, 2014.

[9] M. Siami, M. Naderpour, dan J. Lu, "A Choquet Fuzzy Integral Vertical Bagging Classifier for Mobile Telematics Data Analysis," *Conferention FUZZ-IEEE`19*, 2019, pp. 1-6.

[10] B. Wang, S. Panigrahi, M. Narsude, dan A. Mohanty, "Driver Identification Using Vehicle Telematics Data," *SAE Technical Paper.*, pp. 1 – 8, Mar. 2017.

[11] A. Azevedo dan M. F. Santos, "KDD , SEMMA AND CRISP-DM : A PARALLEL OVERVIEW," *Conferention IADIS'08,* 2009, pp. 182–185.

[12] N. Winitthumkul, P. Phondeenana, dan N. Noomwongs, "Driving Risk Rating for Driver Monitoring Based on Satellite Data," *TSAE.*, Mar. 2016.

[13] A. Pratama, R. C. Wihandika, dan D. E. Ratnawati, "Implementasi Algoritme Support Vector Machine (SVM) untuk Prediksi Ketepatan Waktu Kelulusan Mahasiswa,"*J-PTIIK.*, vol. 2, no.4, pp. 1704–1708, Mar. 2018.

[14] Z. C. Qin, "ROC analysis for predictions made by probabilistic classifiers," *Prosiding ICMLC'05, 2005*, pp. 3119–3124.