

LAPORAN AKHIR
MAGANG & STUDI INDEPENDEN BERSERTIFIKAT
***ACCELERATED MACHINE LEARNING* PROGRAM**
Di PT. Zona Edukasi Nusantara

Diajukan untuk memenuhi persyaratan kelulusan
Program MSIB MBKM

oleh:
Taufik Syah Mauludin / 1902827



PROGRAM STUDI FISIKA
UNIVERSITAS PENDIDIKAN INDONESIA
2022

Lembar Pengesahan
Program Studi Fisika Universitas Pendidikan Indonesia
Accelerated Machine Learning Program
Di PT. Zona Edukasi Nusantara

oleh:
Taufik Syah Mauludin / 1902827

disetujui dan disahkan sebagai
Laporan Studi Independen Bersertifikat Kampus Merdeka

Bandung, 22 Juli 2022
Pembimbing Studi Independen
Program Studi Fisika Universitas Pendidikan Indonesia



Dr. Endi Suhendi, M.Si.

NIP. 197905012003121001

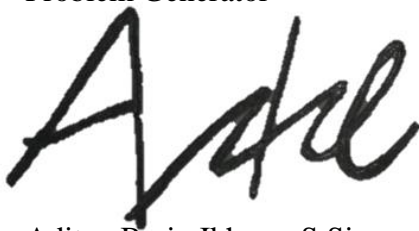
Lembar Pengesahan
Accelerated Machine Learning Program
Di PT. Zona Edukasi Nusantara

oleh:
Taufik Syah Mauludin / 1902827

disetujui dan disahkan sebagai
Laporan Studi Independen Bersertifikat Kampus Merdeka

Jakarta Selatan, 18 Juli 2022

Problem Generator

A handwritten signature in black ink, appearing to read 'Aditya'.

Aditya Bariq Ikhsan, S.Si.

NIP. 202104023

Abstraksi

Banyaknya data belum tentu memberikan nilai tambah apabila tidak mampu menghasilkan informasi yang dibutuhkan. *Machine learning* menjadi solusi strategis atas masalah ini, di mana sebuah program dirancang untuk dapat mempelajari data dalam jumlah yang sangat besar secara untuk menghasilkan informasi yang bernilai. Zenius menghadirkan sesi pendalaman kemampuan *Accelerated Machine Learning* Program membantu peserta untuk dapat menjadi perancang mesin pembelajar ini. Pada *project* ini, sebuah perusahaan mobil Cina Geely Auto bercita-cita untuk memasuki pasar AS dengan mendirikan unit manufaktur mereka di sana dan memproduksi mobil secara lokal untuk memberikan persaingan kepada rekan-rekan mereka di AS dan Eropa. Perusahaan ingin mengetahui variabel mana yang signifikan dalam memprediksi harga mobil dan seberapa baik variabel tersebut menggambarkan harga mobil. *Project* ini diminta untuk memodelkan harga mobil dengan variabel bebas yang tersedia. Model akhir yang digunakan adalah model *random forest regressor*. Manajemen perusahaan mobil Cina Geely Auto sebaiknya memperhatikan spesifikasi *enginesize*, *curbweight*, *hightwaympg*, *horsepower*, dan *carwidth* untuk memenuhi tingkat harga tertentu. Model *random forest regressor* akan menjadi cara yang baik bagi manajemen untuk memahami dinamika harga pasar baru.

Kata Kunci: *Machine Learning, Random Forest Regressor, Car Price Prediction*

Kata Pengantar

Puji syukur kehadiran Tuhan Yang Maha Esa yang telah melimpahkan rahmat dan karunia-Nya sehingga Laporan Akhir Magang & Studi Independen Bersertifikat *Accelerated Machine Learning* Program Di PT. Zona Edukasi Nusantara dapat diselesaikan. Penyusunan makalah ini dimaksudkan sebagai memenuhi persyaratan kelulusan Program MSIB MBKM.

Tersusunnya laporan akhir ini berkat partisipasi seluruh pihak yang terkait. Untuk itu, sepantasnyalah jika kepada penyusun diberikan penghargaan yang setinggi-tingginya. Selanjutnya, penyusun mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang terlibat baik secara langsung maupun tidak langsung sehingga terwujudnya laporan akhir ini kepada:

1. Tuhan Yang Maha Esa karena dengan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan program ini.
2. Orang tua penulis yang telah memberikan dukungan serta mengajarkan arti berusaha dan bekerja keras.
3. Dr. Endi Suhendi, M.Si. selaku Pembimbing Studi Independen Program Studi Fisika Universitas Pendidikan Indonesia.
4. Aditya Bariq Ikhsan, S.Si. selaku Mentor *Accelerated Machine Learning* Program.
5. Grady Matthias Oktavian, S.Mat., M.Kom. dan Stephanie, S.Mat. selaku *Subject Matter Expert Accelerated Machine Learning* Program.
6. Evylia Yanuar Laily selaku teman kelompok final *project*.

Penyusun menyadari dalam laporan akhir ini terdapat banyak kelemahan dan kekurangan. Oleh karena itu, kritik dan saran dari berbagai pihak untuk menyempurnakan makalah ini sangat penyusun nantikan. Semoga laporan akhir ini bermanfaat bagi semua pembaca.

Lembang, Juli 2022

Penulis

Daftar Isi

Lembar Pengesahan Program Studi Fisika Universitas Pendidikan Indonesia.....	ii
Lembar Pengesahan	iii
Abstraksi	iv
Kata Pengantar	v
Daftar Isi.....	vi
Daftar Gambar.....	viii
Daftar Tabel	ix
BAB I Pendahuluan	I-1
I.1 Latar Belakang	I-1
I.2 Lingkup	I-2
I.3 Tujuan.....	I-3
BAB II PT. Zona Edukasi Nusantara	II-1
II.1 Struktur Organisasi.....	II-1
II.2 Lingkup Project	II-2
II.3 Deskripsi Project	II-3
II.4 Jadwal MSIB (<i>Project</i> dan Pembelajaran)	II-4
BAB III Final <i>Project</i>	III-1
III.1 <i>Problem Statement</i>	III-1
III.2 <i>Business Goal</i>	III-1
III.3 <i>Data Cleansing</i>	III-1
III.4 <i>Exploratory Data Analysis</i>	III-2
III.5 <i>Modeling</i>	III-6
III.6 <i>Model Evaluation Metrics</i>	III-6
BAB IV Penutup	IV-1
IV.1 Kesimpulan	IV-1
IV.2 Saran	IV-1
Referensi	x
Lampiran A. TOR	A-1
Lampiran B. Log Activity	B-1

Lampiran C. IPython Notebook.....	C-1
-----------------------------------	-----

Daftar Gambar

Gambar 2.1 Logo PT. Zona Edukasi Nusantara	II-1
Gambar 2.2 Struktur organisasi PT. Zona Edukasi Nusantara.....	II-1
Gambar 3.1 Histogram dan <i>outlier</i> price.....	III-2
Gambar 3.2 Barplot car brand terhadap <i>average</i> price	III-2
Gambar 3.3 Histogram car brand	III-3
Gambar 3.4 Analisis <i>feature</i> symboling	III-3
Gambar 3.5 Analisis <i>feature</i> carbody.....	III-3
Gambar 3.6 Analisis <i>feature</i> drivewheel.....	III-4
Gambar 3.7 Analisis <i>feature</i> enginetype.....	III-4
Gambar 3.8 Analisis <i>feature</i> fuelsystem	III-5
Gambar 3.9 Heatmap <i>correlation</i> data <i>numerical</i>	III-5
Gambar 3.10 Feature importance	III-7

Daftar Tabel

Tabel 2.1 Jadwal pembelajaran <i>Accelerated Machine Learning</i> Program	II-4
Tabel 3.1 Model <i>evaluation metrics</i>	III-6
Tabel 3.2 Model <i>improvement</i>	III-7

BAB I Pendahuluan

I.1 Latar Belakang

Zenius merupakan platform belajar berbasis teknologi yang telah berdiri sejak tahun 2004 yang berfokuskan pada pemahaman konsep dan penalaran ilmiah dengan tujuan dapat memberikan *social impact* yang mampu mentransformasi masa depan bangsa. Zenius percaya bahwa pendidikan yang lebih baik merupakan salah satu hak utama setiap orang, terutama di Indonesia, di manapun mereka berada. Oleh sebab itu, dalam visinya Zenius berupaya untuk menumbuhkan masyarakat Indonesia yang cerdas, cerah dan asyik.

Tidak hanya untuk siswa, sebagai perusahaan yang bergerak di bidang EdTech (*Education Technology*), kami terus melakukan inovasi untuk dapat menciptakan ekosistem belajar yang mudah dan menyenangkan, khususnya bagi para Guru dalam mendukung kegiatan belajar mengajar bersama para siswa. Selaras dengan tujuan tersebut, Zenius telah mengembangkan sistem tata kelola pembelajaran Zenius untuk Guru (“ZenRu”) yang menyediakan layanan pendidikan gratis untuk guru. Selain itu, Zenius juga telah bekerjasama dengan Kementerian Desa, Pembangunan Daerah Tertinggal, dan Transmigrasi untuk memberikan kesetaraan pendidikan dan teknologi bagi masyarakat yang tinggal di daerah yang kurang menguntungkan.

Saat ini, Zenius juga mulai mengembangkan produk yang ditujukan kepada para masyarakat umum dan pekerja, baik dalam sektor formal maupun informal, perusahaan dan perguruan tinggi, yakni ZenPro. Salah satu tujuan dari pengembangan ZenPro adalah untuk menjembatani kesenjangan keterampilan antara kebutuhan industri dan kesediaannya yang dihadirkan dengan mengedepankan pemikiran ilmiah, dan pembelajaran berdasarkan pertanyaan dan kebutuhan praktis. Melalui ZenPro, juga akan diberikan pembelajaran berbasis hasil yang bertujuan untuk menciptakan ekosistem belajar yang menyenangkan. Sejak pertengahan tahun 2020, Zenius melalui ZenPro telah membuat 17 (tujuh belas) pelatihan digital dan telah menjangkau lebih dari 250.000 pengguna melalui program kartu Prakerja.

Dengan semakin berkembangnya Zenius serta aksi nyata Zenius untuk terus menyebarkan kecintaan untuk belajar dan mengembangkan diri, dengan pembelajaran dan pengalaman pelaksanaan program Studi Independen Bersertifikat Zenius pada program Magang Studi Independen Bersertifikat Kampus Merdeka pada *cycle* 1, sekali lagi Zenius berkontribusi pada program Magang Studi Independen Bersertifikat *cycle* 2 melalui program Zenius Studi Independen Bersertifikat yang diadakan untuk memberikan kesempatan bagi tenaga muda atau mahasiswa-mahasiswa di Indonesia untuk dapat mempersiapkan diri sebelum terjun ke dunia profesional terlibat untuk memberikan dampak sosial yang positif terhadap pendidikan di Indonesia dengan cara yang lebih adaptif dan kreatif sesuai dengan ruang lingkup program sertifikasi profesional yang diambil.

I.2 Lingkup

Program Zenius Studi Independen Bersertifikat merupakan program yang diinisiasi oleh Zenius bekerja sama dengan Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi sebagai bagian dari implementasi Magang dan Studi Independen Bersertifikat Kampus Merdeka. Objektif program ini dibuat khusus bagi mahasiswa untuk dapat merasakan pengalaman pembelajaran di luar Kampus berdasarkan tantangan nyata dihadapi oleh mitra/industri untuk mempersiapkan para lulusan sarjana Indonesia untuk mampu berpikir kritis, ilmiah dan terampil sebelum masuk ke dunia kerja setelah lulus dari dunia perkuliahan.

Proses pembelajaran yang dilakukan dalam program Studi Independen ini adalah *experience learning*, di mana peserta harus mengimplementasikan materi yang diperolehnya ke dalam suatu proyek akhir. Pemberian materi akan disediakan dalam bentuk video ajar untuk setiap materi dan kelas *live webinar*. Setiap pemberian materi juga akan dilengkapi dengan kuis/latihan untuk memastikan pemahaman peserta. Selain *hard skill* sesuai kompetensi program sertifikasi yang diambil, *soft skill* juga menjadi target kompetensi peserta Studi Independen yaitu kerja sama tim, berpikir analitis, dan berpikir kritis. Setiap program akan ditutup dengan proyek akhir, di mana peserta akan mengimplementasikan ilmu sertifikasi yang telah didapatkan dalam suatu bentuk

yang konkret untuk kemudian dipresentasikan maupun dipublikasikan kepada publik.

I.3 Tujuan

Tujuan pembelajaran *Accelerated Machine Learning* Program pada studi independen adalah sebagai berikut:

1. Memahami apa itu *machine learning* dan *workflow* dalam proyek penerapan *machine learning*.
2. Mampu melakukan dan memahami proses data *preprocessing*.
3. Mampu mengevaluasi, melakukan, dan memahami cara memproses data dengan berbagai alternatif metode pada *regression*.
4. Mampu mengevaluasi, melakukan, dan memahami cara memproses data dengan berbagai alternatif metode pada *classification*.
5. Mampu mengevaluasi, melakukan, dan memahami cara memproses data dengan berbagai alternatif metode pada *clustering*.
6. Mampu mengevaluasi, melakukan, dan memahami cara memproses data dengan berbagai alternatif metode pada *association*.
7. Mampu mengevaluasi, melakukan, dan memahami cara memproses data dengan berbagai alternatif metode pada *dimetionality reduction*.
8. Mampu mengevaluasi, melakukan, dan memahami cara memproses data dengan berbagai alternatif metode pada *deep learning*.

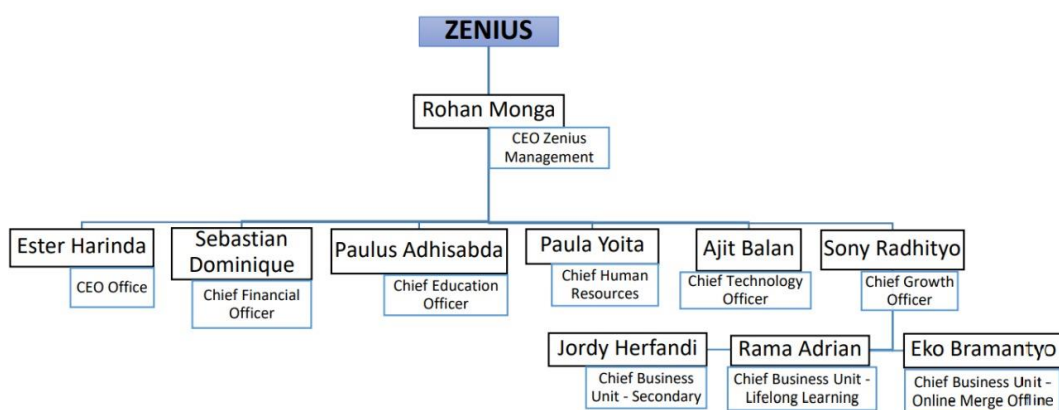
BAB II PT. Zona Edukasi Nusantara

II.1 Struktur Organisasi



Gambar 2.1 Logo PT. Zona Edukasi Nusantara

PT Zona Edukasi Nusantara adalah perusahaan pendidikan berbasis teknologi asal Indonesia yang telah berdiri sejak tahun 2004 yang berfokuskan pada pemahaman konsep dan penalaran ilmiah dengan tujuan memberikan social impact yang mampu mentransformasi masa depan bangsa. Zenius percaya bahwa pendidikan yang lebih baik merupakan salah satu hak utama setiap orang, terutama di Indonesia, dimanapun mereka berada. Zenius menyediakan layanan akses pendidikan dalam format video berbahasa Indonesia yang disajikan secara online melalui website dan aplikasi ponsel.



Gambar 2.2 Struktur organisasi PT. Zona Edukasi Nusantara

II.2 Lingkup Project

Dalam program ini, terdapat 4 (empat) pihak yang terlibat secara intensif untuk mempersiapkan dan menjalankan program.

1. Program Manager:

- Manajemen seluruh proses dalam program, mulai dari seleksi mahasiswa, perekrutan, administrasi dengan pemerintah dan pihak kampus, proses belajar dan asesmen, hingga pemberian nilai dan sertifikat.
- Pusat koordinasi dan komunikasi dari seluruh pihak.
- Mensinkronisasi seluruh pihak, baik dari segi silabus pengajaran maupun *timeline* program.
- Memastikan program berjalan dengan kualitas baik dan sejalan dengan standar yang ditetapkan Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi Republik Indonesia.

2. *Subject Matter Expert*

- Mempersiapkan dan melakukan sinkronisasi silabus yang terkait dengan spesialisasi.
- Menyampaikan bahan ajar, baik dalam bentuk *synchronous* (penyampaian secara langsung/daring) maupun *asynchronous* (memberikan materi atau referensi untuk dipelajari).
- Melakukan asesmen untuk mengukur capaian belajar dan proyek final untuk melatih peserta untuk dapat langsung mempraktikkan ilmu dan keahlian yang dipelajari.
- Berkoordinasi dengan Program Manager dan Mentor.

3. Mentor

- Mendampingi proses belajar peserta dengan melakukan *mentoring* dengan jadwal yang sudah ditetapkan, baik secara berkelompok dan individu sebanyak 1 (satu) kali sebulan.
- Memonitor progres belajar peserta dan menindaklanjuti apabila ada peserta yang tertinggal dalam proses belajar maupun nilai asesmen.

- Membantu Program Manager dan *Subject Matter Expert* untuk melakukan pengecekan tugas/asesmen untuk memastikan tugas telah dikumpulkan dan dikerjakan dengan baik.

4. Peserta

- Mengikuti seluruh rangkaian kegiatan program, pembelajaran oleh *Subject Matter Expert* serta *mentoring* yang diadakan oleh mentor dedikatifnya sesuai dengan jadwal yang ditetapkan.
- Mengikuti keseluruhan program dari awal hingga akhir periode untuk dapat menyelesaikan program sesuai dengan standar dan rubrik penilaian yang ditetapkan Program Manager.
- Mengikuti dan mengerjakan segala bentuk asesmen yang diberikan oleh pengajar ahli dan mentor, baik dalam bentuk, esai, studi kasus, proyek maupun bentuk lainnya.

II.3 Deskripsi Project

Seluruh peserta dalam setiap program akan mendapatkan materi yang sama pada tahapan *Universal Fundamental Skills*. Dalam tahapan ini, peserta akan diberikan pemaparan mengenai kemampuan dasar yang digunakan untuk berpikir dan menghasilkan kualitas berpikir yang dapat dipertanggungjawabkan dari mengolah sebuah informasi, masalah, isu atau situasi. Setelah itu, setiap peserta mempelajari kompetensi terspesialisasi dan pendalaman profesi yang berkaitan dengan kemampuan dasar yang harus dimiliki sesuai dengan *learning path* yang dipilih masing-masing peserta, dengan tutor ahli dan mentor dedikatif pada tahap *Specific Fundamental Skills*. Tahapan pembelajaran berikutnya adalah *Specific Technical Skills* di mana peserta akan belajar untuk membangun dan mempraktikkan kemampuan profesional yang telah dipelajari *learning path* masing-masing peserta melalui berbagai bentuk proyek akhir seperti penelitian, demo produk, pembuatan konten, hingga *study case*.

II.4 Jadwal MSIB (*Project dan Pembelajaran*)

Jadwal topik pembelajaran *Accelerated Machine Learning* Program dapat dilihat pada Tabel 2.1.

Tabel 2.1 Jadwal pembelajaran *Accelerated Machine Learning* Program

Minggu	Topik Pembelajaran
1	Penalaran dan Pemahaman Bacaan
2	
3	
4	Filsafat Dasar
5	
6	
7	Sains Terintegrasi
8	Perilaku Manusia
9	Introduction to Data Science
	Getting Started to Data Science
10	Python I: Installation, Data Structure and Data Types
	Python II: Loops and Functions
11	Python for Data Analysis: DataFrame Basics and Data Cleansing
	Python for Data Analysis: Data Manipulation with Pandas
12	Data Visualization in Python I
	Data Visualization in Python II
13	Statistics for Data Science: Descriptive Statistics
	Statistics for Data Science: Inferential Statistics
14	Hands On: Exploratory Data Analysis with Python I
	Hands On: Exploratory Data Analysis with Python II
15	Machine Learning I (Supervised Learning - Classification)
	Machine Learning II (Supervised Learning - Regression)
16	Machine Learning III (Unsupervised Learning)
	Hyperparameter Tuning
17	Database: SQL Query I

	Database: SQL Query II
18	Making Impact with Data Science
19	Final Project
20	
21	Closing week

BAB III Final Project

III.1 Problem Statement

Sebuah perusahaan mobil Cina Geely Auto bercita-cita untuk memasuki pasar AS dengan mendirikan unit manufaktur mereka di sana dan memproduksi mobil secara lokal untuk memberikan persaingan kepada rekan-rekan mereka di AS dan Eropa. Perusahaan mobil tersebut telah mengontrak sebuah perusahaan konsultan mobil untuk memahami faktor-faktor yang menjadi dasar penentuan harga mobil. Secara khusus, tujuannya adalah untuk memahami faktor-faktor yang mempengaruhi harga mobil di pasar Amerika, karena mungkin sangat berbeda dari pasar Cina. Perusahaan ingin mengetahui variabel mana yang signifikan dalam memprediksi harga mobil dan seberapa baik variabel tersebut menggambarkan harga mobil. Berdasarkan berbagai survei pasar, perusahaan konsultan telah mengumpulkan *dataset* besar dari berbagai jenis mobil di pasar Amerika.

III.2 Business Goal

Final project ini diminta untuk memodelkan harga mobil dengan variabel bebas yang tersedia. Ini akan digunakan oleh manajemen untuk memahami bagaimana sebenarnya harga bervariasi dengan variabel independen. Perusahaan mobil dapat memanipulasi desain mobil, strategi bisnis, dll. untuk memenuhi tingkat harga tertentu. Selanjutnya, model tersebut akan menjadi cara yang baik bagi manajemen untuk memahami dinamika harga pasar baru.

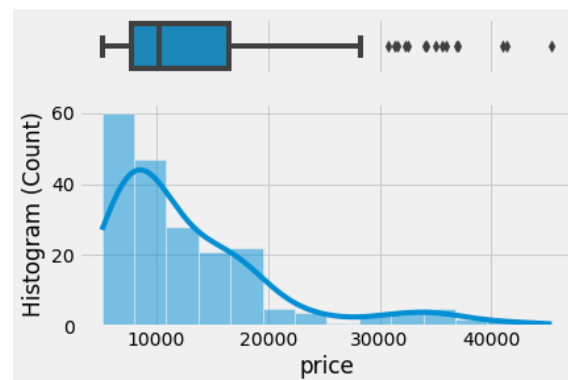
III.3 Data Cleansing

Kumpulan data ini terdiri dari tiga jenis entitas yaitu spesifikasi mobil dalam berbagai karakteristik, peringkat risiko asuransi yang ditetapkan, dan kerugian yang dinormalisasi dalam penggunaan dibandingkan dengan mobil lain. Peringkat kedua sesuai dengan sejauh mana mobil lebih berisiko daripada yang ditunjukkan harganya. Dataset terdiri atas 205 data dan 26 kolom. Dataset tidak memiliki *missing value* maupun *duplicated* data. Kata pertama pada data kolom CarName merupakan *brand* mobil, sehingga diperlukan kolom baru untuk menganalisis *brand* terhadap harga mobil.

Mobil pada awalnya diberi simbol faktor risiko yang terkait dengan harganya. Kemudian, jika lebih berisiko (atau kurang), simbol ini disesuaikan dengan memindahkannya ke atas (atau ke bawah) skala. Para aktuaris menyebut proses ini sebagai *symboling*. Nilai +3 menunjukkan bahwa mobil itu berisiko, -3 bahwa itu mungkin cukup aman. Data pada kolom *symboling* dirubah menjadi data *categorical* [1].

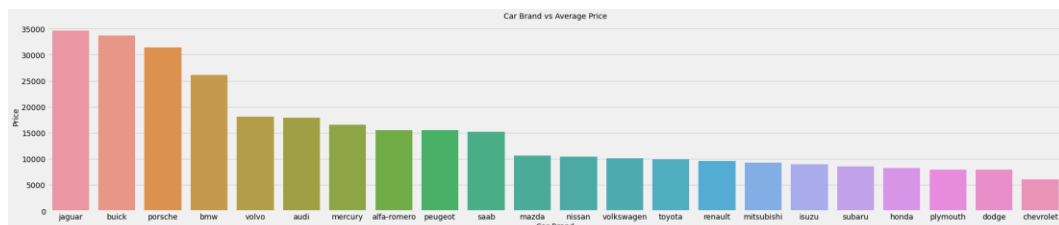
Faktor ketiga adalah pembayaran kerugian rata-rata relatif per tahun kendaraan yang diasuransikan. Nilai ini dinormalisasi untuk semua mobil dalam klasifikasi ukuran tertentu dan mewakili rata-rata kerugian per mobil per tahun. Data pada kolom *fueltype*, *aspiration*, *doornumber*, *engineloaction*, dan *cylindernumber* dirubah menjadi data *numerical*.

III.4 Exploratory Data Analysis



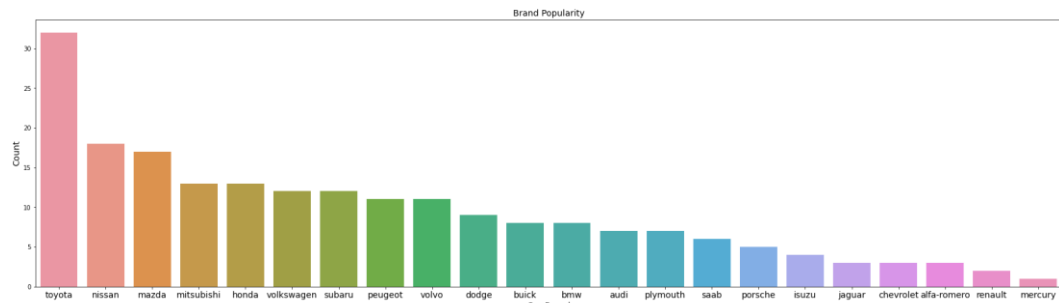
Gambar 3.1 Histogram dan *outlier* price

Distribusi price menunjukkan plot *right skewed*. Median data price berada pada nilai 10.000. Terdapat outlier pada data price yang merupakan bukan data yang janggal. Data yang terdistribusi berada pada kisaran 8.000-17.000 sehingga data price yang lebih tinggi akan dianggap outlier.



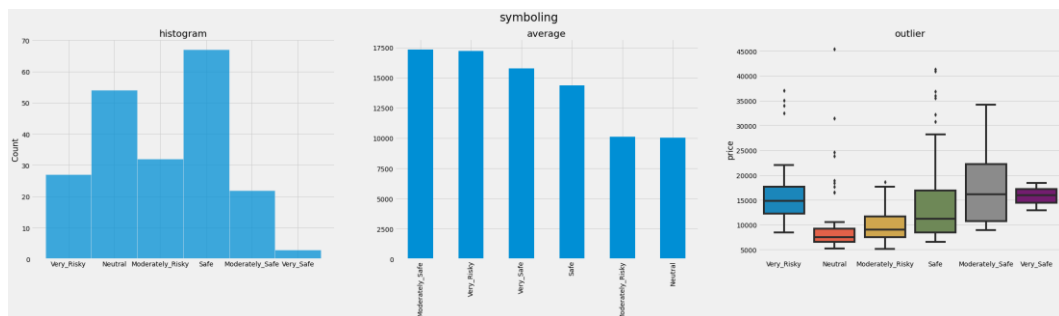
Gambar 3.2 Barplot car brand terhadap *average* price

Brand Jaguar, Buick, dan Porsche memiliki harga rata-rata tertinggi.

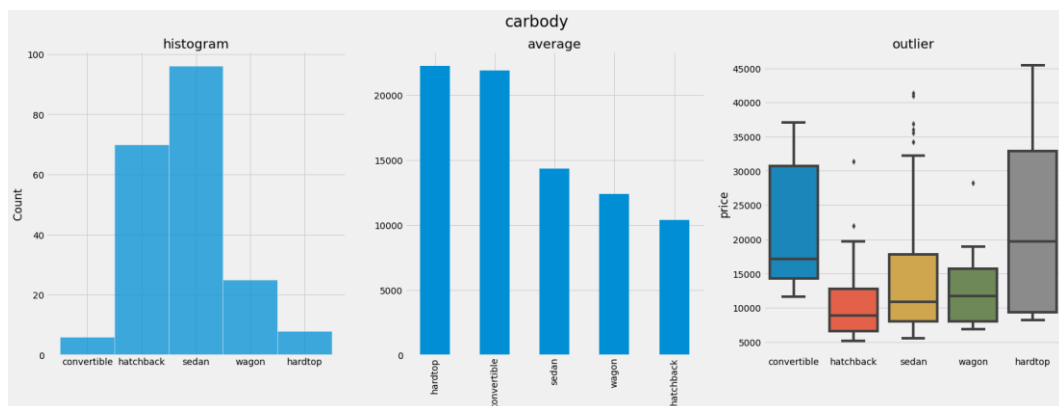


Gambar 3.3 Histogram car brand

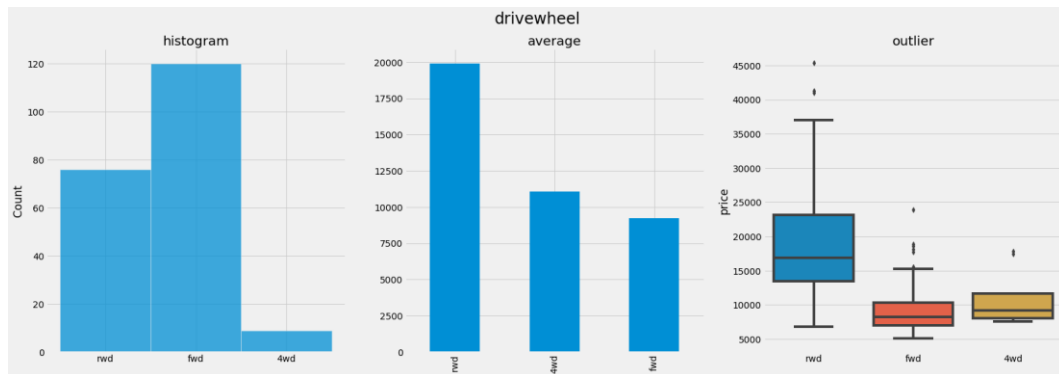
Brand Toyota merupakan mobil yang paling favorit. Sedangkan, brand Mercury merupakan mobil yang paling sedikit disukai.

Gambar 3.4 Analisis *feature* symboling

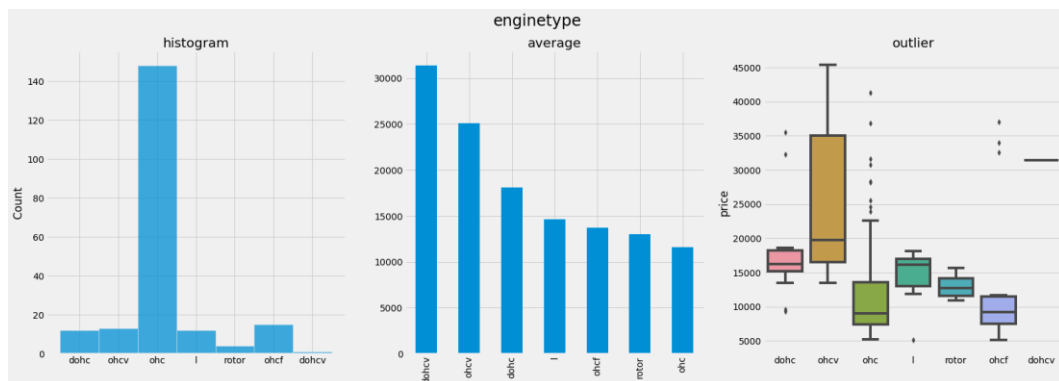
Faktor symboling mobil yang paling banyak digunakan adalah tingkat safe. Tingkat Moderately_Safe dan Very_Risky memiliki harga tertinggi. Harga symboling Moderately_Safe lebih tinggi dibandingkan dengan yang lainnya.

Gambar 3.5 Analisis *feature* carbody

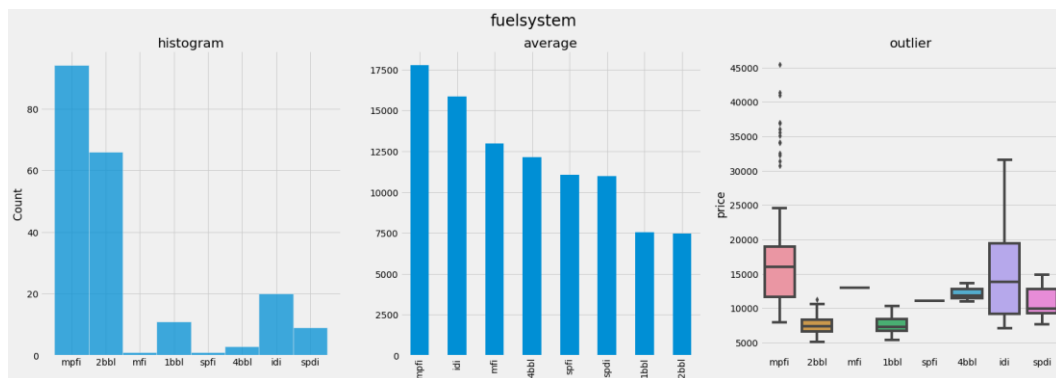
Sebagian besar carbody mobil adalah sedan dan hatchback. Mobil dengan carbody hardtop dan convertible memiliki harga tertinggi. Semua jenis carbody relatif lebih murah dibandingkan dengan carbody hardtop.

Gambar 3.6 Analisis *feature* drivewheel

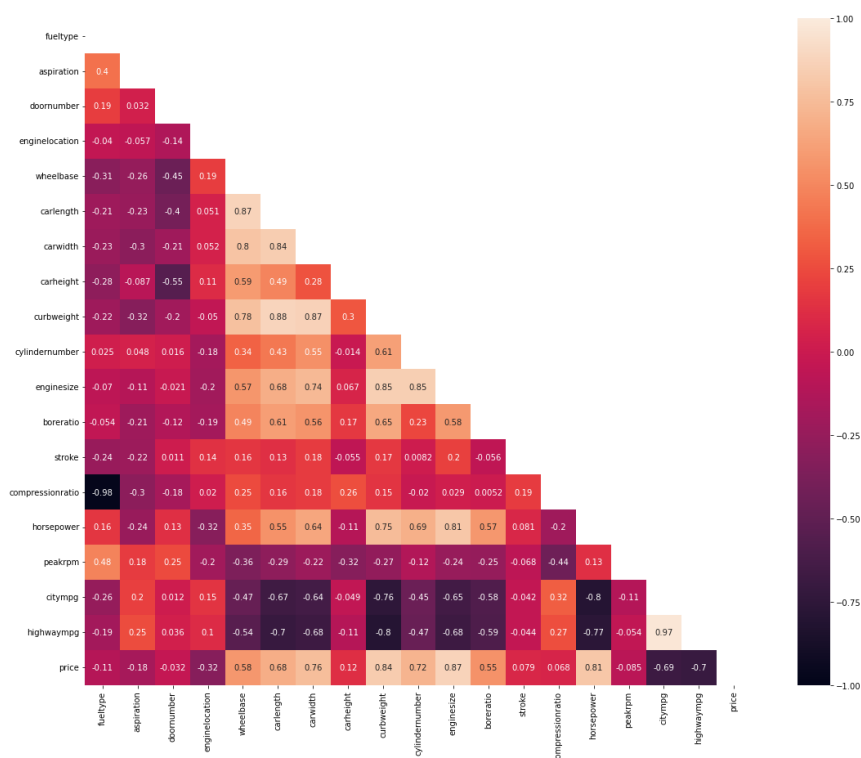
Sebagian besar drivewheel mobil merupakan fwd dan rwd. Mobil dengan drivewheel rwd memiliki harga tertinggi. Semua jenis drivewheel relatif lebih murah dibandingkan dengan drivewheel rwd.

Gambar 3.7 Analisis *feature* enginetype

Sebagian besar enginetype mobil merupakan ohc. Mobil dengan enginetype dohc memiliki harga tertinggi. enginetype ohcv termasuk mobil dengan kisaran harga yang lebih tinggi.

Gambar 3.8 Analisis *feature* fuelsystem

Sebagian besar fuelsystem mobil merupakan mpfi dan 2bbl. Mobil dengan fuelsystem mpfi memiliki harga tertinggi. Kisaran mobil dengan harga tinggi memiliki fuelsystem idi.

Gambar 3.9 Heatmap *correlation* data numerical.

Kolom wheelbase, carlength, dan boreratio berkorelasi kuat secara positif dengan price. Selain itu, kolom carwidth, curbweight, cylindernumber, enginesize, dan horsepower berkorelasi sangat kuat secara positif dengan price. Sedangkan, kolom citympg dan highwaympg berkorelasi sangat kuat secara negative dengan price.

III.5 Modeling

Metode train test split yang digunakan adalah 60% train dan 40% test. Model yang digunakan untuk *final project* adalah *linear regression*, *ridge regression*, *elasticnet regression*, *lasso regression*, dan *random forest regressor*. Evaluasi metrik menggunakan *mean absolute error* (MAE), *root mean square error* (RMSE), dan R^2 . Kolom yang akan dijadikan prediktor adalah kolom *wheelbase*, *carlength*, *carwidth*, *curbweight*, *cylindernumber*, *enginesize*, *boreratio*, *horsepower*, *citympg*, *highwaympg*, *symboling*, *carbody*, *drivewheel*, *enginetype*, *fuelsystem*, dan *brand*. Kolom yang diprediksi adalah kolom *price*.

III.6 Model Evaluation Metrics

Tabel 3.1 Model *evaluation metrics*

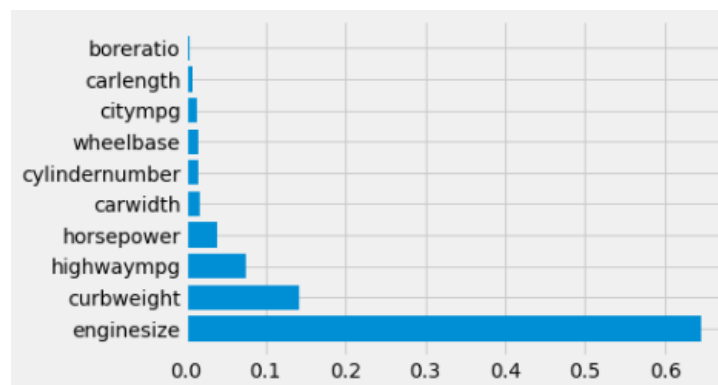
Model	Linear Regression	Ridge Regression	ElasticNet Regression	Lasso Regression	Random Forest Regressor
MAE	1.761	1.699	2.160	1.720	1.400
RMSE	2.491	2.539	3.148	2.485	2.093
R^2	90.30%	89.92%	84.51%	90.34%	93.59%

Nilai error *random forest regressor* yang ditunjukkan oleh MAE dan RMSE pada Tabel 3.1 lebih kecil dibandingkan dengan model lainnya. *Random forest regressor* memiliki korelasi linier antara prediktor dan target yang lebih besar dibandingkan dengan model lainnya. Sehingga, model yang akan dipilih adalah model *random forest regressor*.

Tabel 3.2 Model *improvement*

Model	Random Forest Regressor	
	tanpa Hyperparameter Tuning	dengan Hyperparameter Tuning
MAE	1.400	1.960
RMSE	2.093	2.687
R ²	93.59%	88.71%

Model akhir yang digunakan adalah model *random forest regressor* tanpa *hyperparameter tuning* karena setelah melakukan *improvement*, model tidak menunjukkan hasil yang lebih baik.



Gambar 3.10 Feature importance

Gambar 3.10 menunjukkan 10 *feature* penting dalam model yang dapat dipertimbangkan untuk memenuhi tingkat harga tertentu sebagai dasar penentuan harga mobil. *Feature* tersebut dapat digunakan perusahaan mobil Cina Geely Auto untuk mendirikan unit manufaktur di pasar AS dan memproduksi mobil secara lokal.

BAB IV Penutup

IV.1 Kesimpulan

Kesimpulan yang dapat diperoleh dari hasil analisis final *project* adalah sebagai berikut:

1. Toyota merupakan *brand* mobil yang paling banyak disukai. Namun, brand Jaguar yang memiliki harga yang paling mahal.
2. Manajemen perusahaan mobil Cina Geely Auto sebaiknya memperhatikan spesifikasi *enginesize*, *curbweight*, *hightwaympg*, *horsepower*, dan *carwidth* untuk memenuhi tingkat harga tertentu.
3. Model *random forest regressor* akan menjadi cara yang baik bagi manajemen untuk memahami dinamika harga pasar baru.

IV.2 Saran

Berdasarkan hasil analisis final *project*, saran untuk menghasilkan analisis yang baik adalah sebagai berikut:

1. Menerapkan *treatment* atau *feature engineering* untuk menghasilkan model yang lebih baik.
2. Melakukan *exploratory data analysis* hubungan antar variabel untuk melihat *insight* bisnis lebih lanjut.

Referensi

1. Schlimmer JC (1987) Automobile Data Set. In: Univ. California, Sch. Inf. Comput. Sci. <https://archive.ics.uci.edu/ml/datasets/automobile>.

Lampiran A. TOR



Terms of Reference

Program Zenius Studi Independen Bersertifikat bersama Kampus Merdeka

Program Zenius Studi Independen Bersertifikat merupakan program yang diinisiasi oleh Zenius bekerja sama dengan Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi sebagai bagian dari implementasi Magang dan Studi Independen Bersertifikat Kampus Merdeka. Objektif program ini dibuat khusus bagi mahasiswa untuk dapat merasakan pengalaman pembelajaran di luar Kampus berdasarkan tantangan nyata dihadapi oleh mitra/industri untuk mempersiapkan para lulusan sarjana Indonesia untuk mampu berpikir kritis, ilmiah dan terampil sebelum masuk ke dunia kerja setelah lulus dari dunia perkuliahan.

Proses pembelajaran yang dilakukan dalam program ini adalah *experience learning*, di mana peserta harus mengimplementasikan materi yang diperolehnya ke dalam suatu proyek akhir. Pemberian materi akan disediakan dalam bentuk video ajar untuk setiap materi dan kelas *live webinar*. Setiap pemberian materi juga akan dilengkapi dengan kuis/latihan untuk memastikan pemahaman peserta. Selain *hard skill* sesuai kompetensi program sertifikasi yang diambil, *soft skill* juga menjadi target kompetensi peserta Studi Independen yaitu kerja sama tim, berpikir analitis, dan berpikir kritis. Setiap program akan ditutup dengan proyek akhir, di mana peserta akan mengimplementasikan ilmu sertifikasi yang telah didapatkan dalam suatu bentuk yang konkret untuk kemudian dipresentasikan maupun dipublikasikan kepada publik.

Kerangka Pembelajaran



Seluruh peserta dalam setiap program akan mendapatkan materi yang sama pada tahapan **Universal Fundamental Skills**. Dalam tahapan ini, peserta akan diberikan pemaparan mengenai kemampuan dasar yang digunakan untuk berpikir dan

zenius

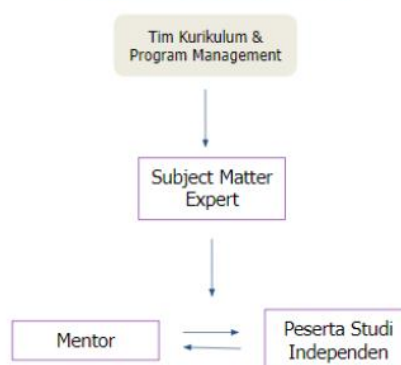
zen PRO

Kampus
Merdeka
INDONESIA JAYA

menghasilkan kualitas berpikir yang dapat dipertanggungjawabkan dari mengolah sebuah informasi, masalah, isu atau situasi. Setelah itu, setiap peserta mempelajari kompetensi terspesialisasi dan pendalaman profesi yang berkaitan dengan kemampuan dasar yang harus dimiliki sesuai dengan *learning path* yang dipilih masing-masing peserta, dengan tutor ahli dan mentor dedikatif pada tahap **Specific Fundamental Skills**. Tahapan pembelajaran berikutnya adalah **Specific Technical Skills** di mana peserta akan belajar untuk membangun dan mempraktikkan kemampuan profesional yang telah dipelajari *learning path* masing-masing peserta melalui berbagai bentuk proyek akhir seperti penelitian, demo produk, pembuatan konten, hingga *study case*.

Kerangka Pendampingan dan Peran Para Pihak

Pendampingan Studi Independen



Dalam program ini, terdapat 4 (empat) pihak yang terlibat secara intensif untuk mempersiapkan dan menjalankan program.

1. Program Manager:

- Manajemen seluruh proses dalam program, mulai dari seleksi mahasiswa, perekrutan, administrasi dengan pemerintah dan pihak



kampus, proses belajar dan asesmen, hingga pemberian nilai dan sertifikat

- Pusat koordinasi dan komunikasi dari seluruh pihak
- Mensinkronisasi seluruh pihak, baik dari segi silabus pengajaran maupun timeline program
- Memastikan program berjalan dengan kualitas baik dan sejalan dengan standar yang ditetapkan Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi Republik Indonesia

2. Subject Matter Expert

- Mempersiapkan dan melakukan sinkronisasi silabus yang terkait dengan spesialisasi
- Menyampaikan bahan ajar, baik dalam bentuk synchronous (penyampaian secara langsung/daring) maupun asynchronous (memberikan materi atau referensi untuk dipelajari)
- Melakukan asesmen untuk mengukur capaian belajar dan proyek final untuk melatih peserta untuk dapat langsung mempraktikkan ilmu dan keahlian yang dipelajari
- Berkoordinasi dengan Program Manager dan Mentor

3. Mentor

- Mendampingi proses belajar peserta dengan melakukan mentoring dengan jadwal yang sudah ditetapkan, baik secara berkelompok dan individu sebanyak 1 (satu) kali sebulan
- Memonitor progres belajar peserta dan menindaklanjuti apabila ada peserta yang tertinggal dalam proses belajar maupun nilai asesmen
- Membantu Program Manager dan Subject Matter Expert untuk melakukan pengecekan tugas/asesmen untuk memastikan tugas telah dikumpulkan dan dikerjakan dengan baik

The logo for zenius, featuring the word "zenius" in a purple, lowercase, sans-serif font with a yellow swoosh underline.The logo for zen PRO, featuring the word "zen" in purple and "PRO" in white inside a yellow rounded rectangle.The logo for Kampus Merdeka, featuring the words "Kampus" and "Merdeka" in blue, with "INDONESIA JAYA" in yellow below them.

4. Peserta

- Mengikuti seluruh rangkaian kegiatan program, pembelajaran oleh Subject Matter Expert serta mentoring yang diadakan oleh mentor dedikatifnya sesuai dengan jadwal yang ditetapkan
- Mengikuti keseluruhan program dari awal hingga akhir periode untuk dapat menyelesaikan program sesuai dengan standar dan rubrik penilaian yang ditetapkan Program Manager
- Mengikuti dan mengerjakan segala bentuk asesmen yang diberikan oleh pengajar ahli dan mentor, baik dalam bentuk, esai, studi kasus, proyek maupun bentuk lainnya

Lampiran B. Log Activity

Minggu/Tgl	Kegiatan	Hasil
1/21-02-22	Pengenalan Zenius & ZenPro, pengenalan Kampus Merdeka & PZSIB, penjelasan rangkaian PZSIB, sambutan dari manajemen Zenius.	Baik
1/22-02-22 s.d. 24-02-22	Belajar asynchronous foundations of learning & thinking, re-learning math, logic, dan verbal skill.	Baik
1/25-02-22	Belajar mengenai keterampilan berfikir fundamental.	Baik
2/01-03-22	Belajar mathematical modelling, building & apply definition, sentence comprehension, logic, dan paragraph comprehension dan asesmen kemampuan berpikir matematis.	Baik
2/02-03-22	Belajar asynchronous logic, konsep = dan aksioma, quantitative data comprehension, dan analogi.	Baik
2/04-03-22	Belajar mengenai pemahaman data kuantitatif dan analogi.	Baik
3/07-03-22	Belajar asynchronous identifikasi kalimat utama dalam paragraf ekspositori, pemahaman bacaan ekspositori dan deskriptif, framework 4 domain, identifikasi ranah fakta dan opini.	Baik
3/08-03-22	Belajar mengenai pemahaman bacaan dan pemetaan domain informasi.	Baik
3/09-03-22	Belajar asynchronous membangun	Baik

s.d. 10-03-22	argumentasi yang koheren dan logis, pemahaman bacaan argumentatif, dan logical fallacy.	
3/11-03-22	Belajar overview fundamental thinking skills serta memahami, menilai, dan membangun argumentasi.	Baik
4/14-03-22	Belajar apa itu filsafat?	Baik
4/15-03-22 s.d. 18-03-22	Belajar asynchronous sejarah filsafat.	Baik
5/21-03-22	Belajar metaconcept.	Baik
5/22-03-22	Belajar asynchronous origin of universe.	Baik
5/23-03-22	Belajar sejarah filsafat.	Baik
5/24-03-22 s.d. 25-03-22	Belajar origin of universe.	Baik
6/28-03-22	Belajar asynchronous asal-usul kehidupan, perkembangan teori evolusi, konsep evolusi, dan keanekaragaman makhluk hidup.	Baik
6/29-03-22	Belajar origin of life.	Baik
6/30-03-22	Belajar human evolution & agricultural revolution.	Baik
6/31-03-22 s.d. 01-04-22	Belajar asynchronous revolusi neolitik dan kota, pendeta dan ksatria.	Baik
7/04-04-22	Belajar scientific revolution.	Baik
7/05-04-22	Belajar asynchronous revolusi sains dan abad pencerahan.	Baik
7/06-04-22	Belajar asynchronous origin of politic and government.	Baik
7/07-04-22	Belajar government, politic, economy.	Baik

s.d. 08-04-22		
8/11-04-22	Belajar asynchronous human evolution, shape by the earth, behavioral modernity, agricultural evolution, dan the problem of modern life.	Baik
8/12-04-22 s.d. 13-04-22	Belajar evolusi perilaku manusia.	Baik
8/14-04-22	Belajar asynchronous human evolution, shape by the earth, behavioral modernity, agricultural evolution, dan the problem of modern life.	Baik
9/18-04-22 s.d. 20-04-22	Belajar introduction to data science.	Baik
9/21-04-22 s.d. 22-04-22	Belajar getting started to data science.	Baik
10/25-04-22 s.d. 26-04-22	Belajar python I: installation, data structure and data types.	Baik
10/27-04-22 s.d. 28-04-22	Belajar python II: loops and functions.	Baik
11/09-05-22	Belajar python for data analysis: DataFrame basics and data cleansing.	Baik
11/10-05-22 s.d. 13-05-22	Belajar python for data analysis: data manipulation with pandas.	Baik
12/17-05-22	Belajar data visualization in python I.	Baik
12/18-05-22 s.d. 19-05-22	Belajar data visualization in python II.	Baik
12/20-05-22	Belajar statistics for data science: descriptive statistics.	Baik

13/23-05-22	Belajar statistics for data science: inferential statistics.	Baik
13/24-05-22 s.d. 27-05-22	Belajar exploratory data analysis with python I.	Baik
14/30-05-22 s.d. 02-06-22	Belajar exploratory data analysis with python II.	Baik
14/03-06-22	Belajar machine learning I (supervised learning - regression).	Baik
15/06-06-22	Belajar machine learning II (supervised learning - classification).	Baik
15/07-06-22	Belajar machine learning III (unsupervised learning).	Baik
15/08-06-22 s.d. 10-06-22	Belajar hyperparameter tuning.	Baik
16/12-06-22	Belajar database: SQL query I.	Baik
16/13-06-22 s.d. 16-06-22	Belajar database: SQL query II.	Baik
16/17-06-22	Belajar making impact with data science.	Baik
17/20-06-22 s.d. 24-06-22	Final project: kick off.	Baik
18/27-06-22 s.d. 01-07-22	Final project.	Baik
19/04-07-22 s.d. 08-07-22	Final project.	Baik
20/11-07-22 s.d. 15-07-22	Final project: showcase presentation.	Baik
21/18-07-22 s.d. 21-07-22	Career coaching	Baik

Lampiran C. IPython Notebook

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import datasets, linear_model, metrics

### Data Understanding

# membaca data
data = pd.read_csv('CarPrice_Assignment.csv')

data.info()

# mengecek missing value
data.isna().sum()

# mengecek data duplikat
data.duplicated().sum()

data.head()

### Exploratory Data Analysis

# mengecek data kategori
categorical_column = data.dtypes==object
categorical_data = data.columns[categorical_column].tolist()

for i in range(len(categorical_data)):
    print("data unik dari {} adalah
    :\n{}\n".format(categorical_data[i],
    data[categorical_data[i]].unique()))

# membuat kolom baru untuk brand
data['Brand'] = data['CarName'].apply(lambda x : x.split()[0])

data['Brand'].unique()

# rename brand
data['Brand'] = data['Brand'].replace({'maxda': 'mazda',
                                     'Nissan': 'nissan',
                                     'porcshce': 'porsche',
                                     'toyouta': 'toyota',
                                     'vokswagen': 'volkswagen',
```

```

        'vw': 'volkswagen'})

data['Brand'].unique()

# convert data yang memiliki 2 value menjadi 1 dan 0
data['fueltype'] = data['fueltype'].map({'gas': 1, 'diesel': 0})
data['aspiration'] = data['aspiration'].map({'std': 1, 'turbo': 0})
data['doornumber'] = data['doornumber'].map({'two': 2, 'four': 4})
data['enginelocation'] = data['enginelocation'].map({'front': 1,
'rear': 0})

# convert `symboling` menjadi categorical value, karena numeric
value tidak memiliki bobot (arti)
def replace_values(a,b):
    data.symboling.replace(a,b,inplace=True)

replace_values(3,'Very_Risky')
replace_values(2,'Moderately_Risky')
replace_values(1,'Neutral')
replace_values(0,'Safe')
replace_values(-1,'Moderately_Safe')
replace_values(-2,'Very_Safe')

# convert `cylindernumber` ke numerical
def replace_values(a,b):
    data.cylindernumber.replace(a,b,inplace=True)

replace_values('four', 4)
replace_values('six', 6)
replace_values('five', 5)
replace_values('three', 3)
replace_values('twelve', 12)
replace_values('two', 2)
replace_values('eight', 8)

# histogram & outlier `price`
sns.color_palette("pastel")
fig, (ax_box, ax_hist) = plt.subplots(2, sharex = True, gridspec_kw
= {'height_ratios':(.2, .8)})

sns.boxplot(data = data, x = 'price', ax = ax_box)
sns.histplot(data = data, x = 'price', ax = ax_hist, kde=True)

ax_box.set(xlabel = '')
ax_hist.set(ylabel = 'Histogram (Count)')

```

```

plt.show()

# drop `car_ID`
data = data.drop(['car_ID'],axis=1)

# grafik `Brand` vs mean `price`
data.groupby(['Brand'])['price'].mean().sort_values(ascending =
False).plot(kind = 'bar')
plt.show()

# histogram `Brand`
data['Brand'].value_counts().plot(kind = 'bar')
plt.show()

### categorical - symboling
fig, (ax1, ax2, ax3) = plt.subplots(1,3, figsize = (35,8), sharey =
False, sharex = False)

# plots
sns.histplot(data = data, x = 'symboling', ax = ax1)
data.groupby(['symboling'])['price'].mean().sort_values(ascending =
False).plot(kind = 'bar', ax = ax2)
sns.boxplot(data = data, x = 'symboling', y = 'price', ax = ax3)

ax1.set_xlabel('')
ax2.set_xlabel('')
ax3.set_xlabel('')
ax1.set_title('histogram')
ax2.set_title('average')
ax3.set_title('outlier')

fig.suptitle('symboling', fontsize = 'xx-large')

plt.show()

### categorical - carbody
fig, (ax1, ax2, ax3) = plt.subplots(1,3, figsize = (25,8), sharey =
False, sharex = False)

# plots
sns.histplot(data = data, x = 'carbody', ax = ax1)
data.groupby(['carbody'])['price'].mean().sort_values(ascending =
False).plot(kind = 'bar', ax = ax2)
sns.boxplot(data = data, x = 'carbody', y = 'price', ax = ax3)
ax1.set_xlabel('')

```

```

ax2.set_xlabel('')
ax3.set_xlabel('')
ax1.set_title('histogram')
ax2.set_title('average')
ax3.set_title('outlier')

fig.suptitle('carbody', fontsize = 'xx-large')

plt.show()

### categorical - drivewheel
fig, (ax1, ax2, ax3) = plt.subplots(1,3, figsize = (25,8), sharey =
False, sharex = False)

# plots
sns.histplot(data = data, x = 'drivewheel', ax = ax1)
data.groupby(['drivewheel'])['price'].mean().sort_values(ascending =
False).plot(kind = 'bar', ax = ax2)
sns.boxplot(data = data, x = 'drivewheel', y = 'price', ax = ax3)

ax1.set_xlabel('')
ax2.set_xlabel('')
ax3.set_xlabel('')
ax1.set_title('histogram')
ax2.set_title('average')
ax3.set_title('outlier')

fig.suptitle('drivewheel', fontsize = 'xx-large')

plt.show()

### categorical - enginetype
fig, (ax1, ax2, ax3) = plt.subplots(1,3, figsize = (25,8), sharey =
False, sharex = False)

# plots
sns.histplot(data = data, x = 'enginetype', ax = ax1)
data.groupby(['enginetype'])['price'].mean().sort_values(ascending =
False).plot(kind = 'bar', ax = ax2)
sns.boxplot(data = data, x = 'enginetype', y = 'price', ax = ax3)

ax1.set_xlabel('')
ax2.set_xlabel('')
ax3.set_xlabel('')
ax1.set_title('histogram')

```

```

ax2.set_title('average')
ax3.set_title('outlier')

fig.suptitle('enginetype', fontsize = 'xx-large')

plt.show()

### categorical - fuelsystem
fig, (ax1, ax2, ax3) = plt.subplots(1,3, figsize = (25,8), sharey =
False, sharex = False)

# plots
sns.histplot(data = data, x = 'fuelsystem', ax = ax1)
data.groupby(['fuelsystem'])['price'].mean().sort_values(ascending =
False).plot(kind = 'bar', ax = ax2)
sns.boxplot(data = data, x = 'fuelsystem', y = 'price', ax = ax3)

ax1.set_xlabel('')
ax2.set_xlabel('')
ax3.set_xlabel('')
ax1.set_title('histogram')
ax2.set_title('average')
ax3.set_title('outlier')

fig.suptitle('fuelsystem', fontsize = 'xx-large')

plt.show()

# data numerik
numerical_column = data.select_dtypes(exclude=['object']).columns
numerical_data = data[numerical_column]

# heatmap data numerical
plt.figure(figsize=(20, 15))
mask = np.zeros_like(numerical_data.corr())
mask[np.triu_indices_from(mask)] = True
sns.heatmap(numerical_data.corr(), mask=mask, vmin=-1, vmax=1, annot
= True, square=True)

# data untuk model
data_model = data.copy()

data_model.drop(['fueltype', 'aspiration', 'enginelocation',
'carheight', 'stroke', 'compressionratio', 'peakrpm', 'doornumber',
'CarName'], axis=1, inplace=True)

```

```

data_encoding = pd.get_dummies(data_model)
data_encoding.head()

X = data_encoding.loc[:, data_encoding.columns != 'price']
y = data_encoding['price']

# kolom prediktor
X.columns

# splitting X and y into training and testing sets
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.4, random_state=1)

# create linear regression object
reg = linear_model.LinearRegression()

# train the model using the training sets
reg.fit(X_train, y_train)

# regression coefficients
print('Coefficients: ', reg.coef_)

# regression intercept
print('Intercept: ', reg.intercept_)

# variance score: 1 means perfect prediction
print('Variance score: {}'.format(reg.score(X_test, y_test)))

# plot for residual error

## setting plot style
plt.style.use('fivethirtyeight')

## plotting residual errors in training data
plt.scatter(reg.predict(X_train), reg.predict(X_train) - y_train,
            color = "green", s = 10, label = 'Train data')

## plotting residual errors in test data
plt.scatter(reg.predict(X_test), reg.predict(X_test) - y_test,
            color = "blue", s = 10, label = 'Test data')

## plotting line for zero residual error
plt.hlines(y = 0, xmin = 0, xmax = 40000, linewidth = 2)

```



```
## plotting legend
plt.legend(loc = 'best')

## plot title
plt.title("Residual errors")

## method call for showing the plot
plt.show()

y_pred = reg.predict(X_test)

# MAE
from sklearn.metrics import mean_absolute_error
mean_absolute_error(y_test, y_pred)

# RMSE
from sklearn.metrics import mean_squared_error
mean_squared_error(y_test, y_pred, squared=False)

# R2
from sklearn.metrics import r2_score
r2_score(y_test, y_pred)

# ridge regression
from sklearn.linear_model import Ridge
model = Ridge()
model.fit(X_train, y_train)

# regression coefficients
print('Coefficients: ', model.coef_)

# regression intercept
print('Intercept: ', model.intercept_)

# variance score: 1 means perfect prediction
print('Variance score: {}'.format(model.score(X_test, y_test)))

# plot for residual error

## setting plot style
plt.style.use('fivethirtyeight')
## plotting residual errors in training data
plt.scatter(model.predict(X_train), model.predict(X_train) -
y_train, color = "green", s = 10, label = 'Train data')
```

```

## plotting residual errors in test data
plt.scatter(model.predict(X_test), model.predict(X_test) - y_test,
            color = "blue", s = 10, label = 'Test data')

## plotting line for zero residual error
plt.hlines(y = 0, xmin = 0, xmax = 40000, linewidth = 2)

## plotting legend
plt.legend(loc = 'best')

## plot title
plt.title("Residual errors")

## method call for showing the plot
plt.show()

y_pred = model.predict(X_test)

# MAE
from sklearn.metrics import mean_absolute_error
mean_absolute_error(y_test, y_pred)

# RMSE
from sklearn.metrics import mean_squared_error
mean_squared_error(y_test, y_pred, squared=False)

# R2
from sklearn.metrics import r2_score
r2_score(y_test, y_pred)

# lasso regression
from sklearn.linear_model import Lasso
model = Lasso()
model.fit(X_train, y_train)

# regression coefficients
print('Coefficients: ', model.coef_)

# regression intercept
print('Intercept: ', model.intercept_)
# variance score: 1 means perfect prediction
print('Variance score: {}'.format(model.score(X_test, y_test)))

```

```

# plot for residual error

## setting plot style
plt.style.use('fivethirtyeight')

## plotting residual errors in training data
plt.scatter(model.predict(X_train), model.predict(X_train) -
y_train, color = "green", s = 10, label = 'Train data')

## plotting residual errors in test data
plt.scatter(model.predict(X_test), model.predict(X_test) - y_test,
            color = "blue", s = 10, label = 'Test data')

## plotting line for zero residual error
plt.hlines(y = 0, xmin = 0, xmax = 40000, linewidth = 2)

## plotting legend
plt.legend(loc = 'best')

## plot title
plt.title("Residual errors")

## method call for showing the plot
plt.show()

y_pred = model.predict(X_test)

# MAE
from sklearn.metrics import mean_absolute_error
mean_absolute_error(y_test, y_pred)

# RMSE
from sklearn.metrics import mean_squared_error
mean_squared_error(y_test, y_pred, squared=False)

# R2
from sklearn.metrics import r2_score
r2_score(y_test, y_pred)

# elasticnet regression
from sklearn.linear_model import ElasticNet
model = ElasticNet()
model.fit(X_train, y_train)

```

```

# regression coefficients
print('Coefficients: ', model.coef_)

# regression intercept
print('Intercept: ', model.intercept_)

# variance score: 1 means perfect prediction
print('Variance score: {}'.format(model.score(X_test, y_test)))

# plot for residual error

## setting plot style
plt.style.use('fivethirtyeight')

## plotting residual errors in training data
plt.scatter(model.predict(X_train), model.predict(X_train) -
y_train, color = "green", s = 10, label = 'Train data')

## plotting residual errors in test data
plt.scatter(model.predict(X_test), model.predict(X_test) - y_test,
            color = "blue", s = 10, label = 'Test data')

## plotting line for zero residual error
plt.hlines(y = 0, xmin = 0, xmax = 40000, linewidth = 2)

## plotting legend
plt.legend(loc = 'best')

## plot title
plt.title("Residual errors")

## method call for showing the plot
plt.show()

y_pred = model.predict(X_test)

# MAE
from sklearn.metrics import mean_absolute_error
mean_absolute_error(y_test, y_pred)

# RMSE
from sklearn.metrics import mean_squared_error
mean_squared_error(y_test, y_pred, squared=False)

```

```
# R2
from sklearn.metrics import r2_score
r2_score(y_test, y_pred)

# random forest regressor
from sklearn.ensemble import RandomForestRegressor

rf = RandomForestRegressor()
rf.fit(X_train, y_train)

# variance score: 1 means perfect prediction
print('Variance score: {}'.format(rf.score(X_test, y_test)))

# plot for residual error

## setting plot style
plt.style.use('fivethirtyeight')

## plotting residual errors in training data
plt.scatter(rf.predict(X_train), rf.predict(X_train) - y_train,
            color = "green", s = 10, label = 'Train data')

## plotting residual errors in test data
plt.scatter(rf.predict(X_test), rf.predict(X_test) - y_test,
            color = "blue", s = 10, label = 'Test data')

## plotting line for zero residual error
plt.hlines(y = 0, xmin = 0, xmax = 40000, linewidth = 2)

## plotting legend
plt.legend(loc = 'best')

## plot title
plt.title("Residual errors")

## method call for showing the plot
plt.show()

y_pred = rf.predict(X_test)

# MAE
from sklearn.metrics import mean_absolute_error
mean_absolute_error(y_test, y_pred)
```

```

# RMSE
from sklearn.metrics import mean_squared_error
mean_squared_error(y_test, y_pred, squared=False)

# R2
from sklearn.metrics import r2_score
r2_score(y_test, y_pred)

# feature importance
rf.feature_importances_

feature = pd.Series(rf.feature_importances_, index=X.columns)
feature.sort_values(ascending=False, inplace=True)
print(feature.head(10))

# hyperparameter tuning
rf = RandomForestRegressor(random_state=42, n_jobs=-1)

params = {
    'max_depth': [2,3,5,10,20],
    'min_samples_leaf': [5,10,25,50,100,200],
    'n_estimators': [5,10,25,50,100,200]
}

from sklearn.model_selection import GridSearchCV

# Instantiate the grid search model
grid_search = GridSearchCV(estimator=rf,
                           param_grid=params,
                           cv = 5,
                           n_jobs=-1, verbose=1, scoring="accuracy")

grid_search.fit(X_train, y_train)

grid_search.best_score_

rf_best = grid_search.best_estimator_
rf_best.fit(X_train, y_train)

# variance score: 1 means perfect prediction
print('Variance score: {}'.format(rf_best.score(X_test, y_test)))
# plot for residual error

## setting plot style
plt.style.use('fivethirtyeight')

```

```
## plotting residual errors in training data
plt.scatter(rf_best.predict(X_train), rf_best.predict(X_train) -
y_train,
            color = "green", s = 10, label = 'Train data')

## plotting residual errors in test data
plt.scatter(rf_best.predict(X_test), rf_best.predict(X_test) -
y_test,
            color = "blue", s = 10, label = 'Test data')

## plotting line for zero residual error
plt.hlines(y = 0, xmin = 0, xmax = 40000, linewidth = 2)

## plotting legend
plt.legend(loc = 'best')

## plot title
plt.title("Residual errors")

## method call for showing the plot
plt.show()

y_pred = rf_best.predict(X_test)

# MAE
from sklearn.metrics import mean_absolute_error
mean_absolute_error(y_test, y_pred)

# RMSE
from sklearn.metrics import mean_squared_error
mean_squared_error(y_test, y_pred, squared=False)

# R2
from sklearn.metrics import r2_score
r2_score(y_test, y_pred)
```