UNIVERSITAS PEMBANGUNAN NASIONAL VETERAN JAKARTA

LAPORAN TUGAS BESAR

Teori dan Praktikum Big Data

Disusun Oleh:

Salsabila Oktafani	2010511001
Jihan Kamilah	2010511013
Nurhikmah Mawaddah Solin	2010511026
Mochammad Adhi Buchori	2010511028
Yaasintha La Jopin Arisca Corpputy	2010511091
Muhammad Ghozi Attamimi	2010511092

Dosen Pengampu:

Muhammad Adrezo, MSc.

PROGRAM STUDI S-1 INFORMATIKA

FAKULTAS ILMU KOMPUTER

2022

KATA PENGANTAR

Puji dan syukur kehadirat Tuhan Yang Maha Esa atas kehendak dan izin-Nya Laporan Tugas Besar Ujian Akhir Semester pada mata kuliah Teori dan Praktikum Big Data dapat selesai dengan baik sesuai dengan waktu yang telah ditargetkan.

Dalam hal ini, Laporan Tugas Besar disusun dengan tujuan untuk memenuhi penilaian Ujian Akhir Semester pada mata kuliah Teori dan Praktikum Big Data. Selain itu, laporan ini juga disusun dengan tujuan untuk menambah wawasan khususnya pada bidang Big Data.

Dalam menyusun laporan, tim penulis mendapat banyak dukungan dari berbagai pihak baik moril maupun materi. Oleh karena itu, tim penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya terutama kepada:

1. Muhammad Adrezo, MSc. selaku dosen pada mata kuliah Teori dan Praktikum Big Data yang telah membantu tim penulis baik secara masukan, doa, saran, maupun perhatian.
2. Rekan kerja kelompok yang telah mendukung sehingga dapat menyelesaikan laporan dengan waktu yang telah ditargetkan.

Tim penulis menyadari, bahwa laporan yang dibuat ini masih jauh dari kata sempurna baik segi penyusunan, bahasa, maupun penulisan. Oleh karena itu, kami sangat mengharapkan kritik dan saran yang membangun dari pihak pembaca guna menjadi acuan agar kami dapat menjadi lebih baik lagi di masa mendatang.

Harapan tim penulis, semoga laporan yang kami buat dapat bermanfaat, dipahami, serta dapat dijadikan acuan sebagai media pembelajaran yang baik bagi para pembaca untuk sekarang maupun di masa yang akan datang.

BAGIAN I
INTRODUCTION

1.1. Domain Proyek

Domain proyek yang diambil untuk proyek Big Data ini, yaitu **Kesehatan** dengan judul ***Predictive Analytics* : Prediksi Kebutuhan Perawatan Kesehatan Mental terhadap Seorang Karyawan**.

1.2. Case Overview *(Bagaimana teknologi big data mengambil peran pada kasus yang kalian pilih)*

Menurut seorang ahli kesehatan, Merriam Webster, mengartikan kesehatan mental menjadi suatu keadaan emosional dan psikologis yang baik. Dimana individu bisa menggunakan kemampuan kognisi, emosi, yang bermanfaat dalam suatu komunitas, serta bisa memenuhi kebutuhan hidupnya [1]. Kesehatan mental memiliki pengaruh dalam kesejahteraan emosional, psikologis, serta sosial seseorang. Tentunya hal itu mempengaruhi cara seseorang dalam berpikir, merasakan, serta bertindak pada suatu hal. Kesehatan mental juga memiliki pengaruh dalam memberi bantuan untuk seseorang menentukan cara menangani stress, korelasi dengan orang lain, serta dalam menentukan pilihan.

Gubernur Jawa Tengah, Ganjar Pranowo, meminta big data dan artificial intelligence harus dilakukan guna memudahkan pelayanan kesehatan bagi masyarakat. Big data memiliki banyak peran dalam dunia kesehatan. Big data sendiri bisa digunakan dalam mengidentifikasi suatu penyakit, mengatasi berbagai masalah dengan merekomendasikan sistem, sensor-based health condition, studi asosiasi genome, dan sebagainya. Dengan melihat banyaknya data saat ini, nilai prediktif menjadi poin informasi dalam suatu kasus pada kesehatan mental. Dukungan dengan memanfaatkan teknologi, big data dalam pelayanan kesehatan mental menciptakan adanya kebutuhan data dan informasi yang akan terintegrasi pada pelayanan pasien, rekam medis, ataupun keuangan pasien. Big data menjadi bentuk dukungan dalam menjaga keamanan penyimpanan data, manajemen data pasien, menyimpan serta menemukan data rekam medis pasien. Sehingga penanganan pasien bisa dilakukan dengan baik.

Kesehatan mental menjadi salah satu faktor penting. Dimana kesehatan mental secara langsung maupun tidak langsung bisa mempengaruhi kinerja, produktivitas dari karyawan. Dalam hasil penelitian oleh Koopman, dkk (2002) memperoleh hasil bahwa produktivitas kerja dipengaruhi oleh kesehatan mental para pekerja [2]. Ditemukan juga dalam penelitian, Koopman, dkk (2002) adanya hubungan sangat erat antara kedua variabel tersebut. Dengan begitu, dalam menciptakan ruang lingkup kerja yang efektif dan efisien, kesehatan mental memiliki peran yang sangat penting. Dengan membangun model *machine learning* yang dapat digunakan dalam melakukan identifikasi seorang karyawan yang sedang memerlukan perawatan kesehatan mental, bisa terciptanya ruang lingkup kerja yang efektif dan efisien.

BAGIAN II
BUSINESS UNDERSTANDING

2.1. Problem Statements

Dalam pelaksanaannya, ditetapkannya rumusan masalah yang perlu diselesaikan pada proyek ini yang di antaranya adalah sebagai berikut:

1. Dari serangkaian fitur yang ada, fitur apa yang paling berpengaruh terhadap kebutuhan seorang karyawan yang membutuhkan perawatan kesehatan mental?
2. Apa model *machine learning* yang paling baik untuk memprediksi kebutuhan seorang karyawan yang membutuhkan perawatan kesehatan mental?

2.2. Goals

Dalam pelaksanaan suatu proyek, tentu memiliki tujuan yang akan dicapai. Adapun tujuan dari proyek ini yang di antaranya adalah sebagai berikut:

1. Membangun model *machine learning* yang dapat digunakan untuk memprediksi karyawan yang membutuhkan perawatan kesehatan mental.
2. Membandingkan beberapa algoritma guna memperoleh akurasi terbaik dalam melakukan prediksi terhadap karyawan yang membutuhkan perawatan kesehatan mental.
3. Mencari fitur yang paling berpengaruh terhadap kebutuhan seorang karyawan yang membutuhkan perawatan kesehatan mental.

2.3. Innovation *(Inovasi apa saja yang akan kalian kembangkan berdasarkan kasus yang kalian pilih)*

Pada proyek ini penulis mengembangkan model *machine learning* yang mampu melakukan prediksi dengan menggunakan 7 algoritma berbeda. Seluruh model tersebut nantinya akan dibandingkan guna mendapatkan model terbaik. Berikut merupakan algoritma yang penulis gunakan dalam membangun model prediksi:

1. Logistic Regression

   Logistic Regression merupakan suatu cara pemodelan masalah keterhubungan antara suatu variabel independen terhadap variabel dependen. Contohnya adalah

menentukan apakah suatu nilai ukuran tumor tertentu termasuk ke dalam tumor ganas atau tidak [6].

2. K-Nearest Neighbors

   K-Nearest Neighbors merupakan algoritma supervised learning dimana hasil dari instance yang baru diklasifikasikan berdasarkan mayoritas dari kategori k-tetangga terdekat [7].

3. Decision Tree

   Decision Tree merupakan model prediksi menggunakan struktur pohon atau struktur berhirarki [8].

4. Random Forest

   Random Forest merupakan algoritma klasifikasi yang merupakan pengembangan Decision Tree karena RF dapat dikatakan sebagai gabungan dari beberapa Decision Tree. Algoritma ini menggunakan bagging dan feature randomness ketika membangun setiap pohon individu untuk mencoba membuat hutan pohon yang tidak berkorelasi yang prediksinya oleh komite lebih akurat daripada pohon individu mana pun [9].

5. Gradient Boosting

   Gradient Boosting merupakan algoritma supervised learning dimana prediktor yang kuat dibangun dengan cara aditif atau berurutan menggunakan prediktor lemah, seperti Decision Tree. Algoritma ini biasa digunakan untuk tugas klasifikasi dan regresi [10].

6. AdaBoost

   AdaBoost merupakan meta-algoritma klasifikasi statistik yang dirumuskan oleh Yoav Freund dan Robert Schapire pada tahun 1995, yang memenangkan Penghargaan Gödel 2003 untuk karya mereka. Algoritma ini dapat digunakan bersama dengan banyak jenis algoritma pembelajaran lainnya untuk meningkatkan kinerja. Output dari algoritma pembelajaran lainnya ('weak learners') digabungkan menjadi jumlah tertimbang yang mewakili output akhir dari pengklasifikasi yang ditingkatkan [1X].

7. XGBoost

   XGBoost yang merupakan singkatan dari Extreme Gradient Boosting merupakan library machine learning Gradient-Boosted Decision Tree (GDBT) terdistribusi yang

dapat diskalakan dan didorong oleh gradien. Algoritma ini memberikan peningkatan pohon paralel dan merupakan perpustakaan pembelajaran mesin terkemuka untuk masalah regresi, klasifikasi, dan peringkat [11].

2.4. Challenges *(Apa tantangan yang akan muncul ketika mengembangkan teknologi ini.)*

Dalam proses pembuatannya, kendala pertama yang ditemui adalah penggunaan pyspark. Tipe data pyspark dataframe yang berbeda dengan dataframe pandas mengharuskan kami untuk mencari metode baru, yang memiliki tujuan yang sama dengan pandas, khusus untuk pyspark. Contoh yang dapat ditemukan di awal adalah metode info() pada pandas. Karena penggunaan metode describe() pada pandas dan pyspark memiliki hasil yang hampir sama, diasumsikan untuk metode info() pada pandas juga dimiliki oleh pyspark. Akan tetapi sayangnya tidak, karena pyspark menggunakan metode printSchema() yang walaupun memiliki tampilan yang berbeda tapi memberikan informasi yang sama.

Ketidakdekatan kami dengan pyspark merupakan tantangan yang terus dialami selama proses perancangan model berlangsung. Untuk mencari nilai unik dari setiap fitur yang disediakan dataset cukup membingungkan dibandingkan dengan versi pandasnya. Selain itu, pencarian *missing value* juga menjadi tantangan yang cukup membingungkan. Missing data dalam dataframe pandas akan dibaca sebagai NaN atau “?”. Karena lebih familiar dengan hal ini, pencarian *missing* data dilakukan dengan menjadi np.nan atau “?” yang ada di dalam setiap fitur. Hasilnya tidak ada, sedangkan saat dibandingkan menggunakan pandas banyak missing data yang terdeteksi. Masalah ini terselesaikan setelah mengetahui bahwa pyspark membaca missing data sebagai “NA”. Setelah mengetahui hal ini lah missing data baru bisa dideteksi.

Perbedaan cara membaca pyspark dengan pandas berlanjut ketika menyeleksi fitur “Age” yang seharusnya dibaca sebagai integer, tapi dibaca sebagai String oleh pyspark. Pyspark juga memiliki fitur unik atau berbeda, salah satunya adalah saat mengambil salah satu kolom dari dataframe data yang diberikan bertipe dataframe pyspark yang memiliki tampilan, seperti berikut:

Gambar 1. Perbedaan Cara Membaca Pyspark dengan Pandas pada Fitur “Age”.

Data tersebut menjadi tidak bisa diolah secara langsung, sebagaimana pandas yang memberikan datanya dengan format *list*. Perbedaan ini terjadi berulang kali, terutama pada saat melakukan *plotting* terhadap suatu data. Untuk memudahkan *plotting* tanpa mengubah isi dari dataset, khusus untuk beberapa *plotting*, dataframe diubah menjadi pandas terlebih dahulu untuk mengurangi proses perubahan format.

Tantangan lainnya hadir pada saat melakukan perubahan data kategorikal yang berupa teks menjadi angka. Metode yang kami gunakan mengharuskan penambahan kolom baru untuk hasil dari bentuk data ini. Oleh karena itu, untuk memudahkan pembacaan, kolom lama harus selalu dihapus dari dataset setelah diubah. Hal yang sama juga terjadi pada saat normalisasi, dimana hasilnya harus berada di kolom baru dengan format yang tidak bisa langsung diolah.

2.5. Literature Review

2.5.1. PySpark

Dengan adanya spark, memungkinkan data dan komputasi bisa disebarkan dengan banyak node. PySpark, python API untuk aplikasi spark, menjadi salah satu modul pada python untuk digunakan pada Apache Spark. PySpark merupakan bahasa pemrograman python yang telah disediakan oleh Apache Spark. Dimana Apache Spark sendiri merupakan *framework* yang berguna dalam melakukan proses dan melakukan analisis Big Data. Apache Spark juga digunakan dalam membangun jalur dalam pemrosesan suatu data dalam skala besar. Dalam memproses suatu data dilakukan secara in-memory, dengan dilengkapi API pengembangan yang ekspresif guna membantu memudahkan para orang yang bekerja dalam bidang data saat melakukan eksekusi pekerjaannya yang butuh perulangan akses yang cepat saat proses berlangsung.

2.5.2. Supervised Learning

Supervised learning merupakan algoritma dari *machine learning* dengan menggunakan data latih. Dimana artinya data latih tersebut mempunyai data yang

sudah berlabel. Dengan tujuan agar suatu mesin bisa melakukan identifikasi label *input* dengan menggunakan fitur yang nanti akan digunakan untuk membuat atau melakukan klasifikasi dan prediksi. Supervised learning mengenali suatu data dari label khusus dimana telah ada sebelumnya. Dengan memanfaatkan ribuan data, dapat melatih model dari supervised learning dalam melakukan prediksi terhadap suatu hal.

2.5.3. Klasifikasi

Klasifikasi merupakan salah satu metode dari supervised learning yang populer. Klasifikasi merupakan suatu pengelompokan data yang memiliki label. Klasifikasi, merupakan suatu proses guna menemukan suatu model yang bisa membedakan antara kelas data atau konsep. Adapun metode klasifikasi pada *machine learning*, seperti Support Vector Machines (SVM), Naive Bayes, dan algoritma sejenis lainnya.

2.5.4. Regresi

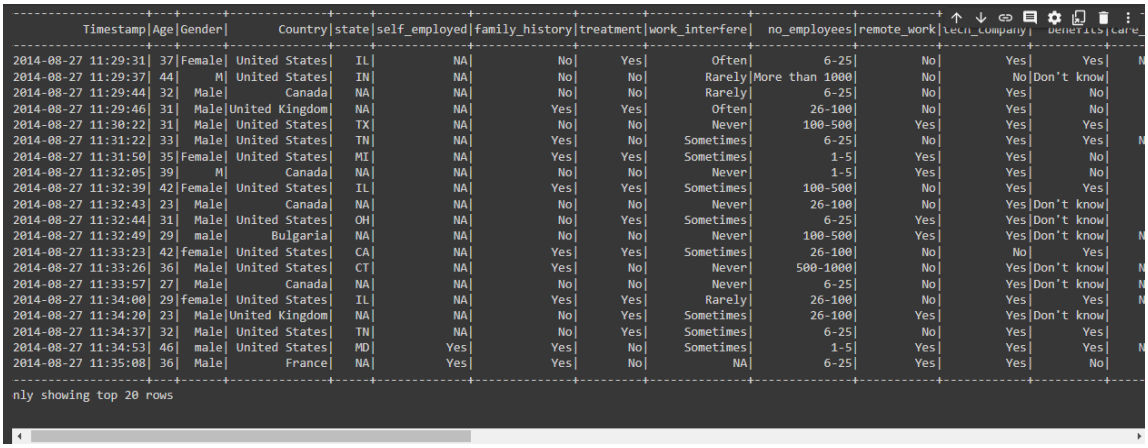
Regresi merupakan salah satu metode lain dari supervised learning yang bekerja dengan mengembalikan target numerik pada setiap sampelnya. Data *input* dalam regresi juga sudah memiliki label sebelumnya. Dalam metode regresi, bisa dimanfaatkan dalam menguji hipotesis, melihat hubungan sebab akibat antar variabel dalam penelitian, mengetahui apakah variabel yang berpengaruh terhadap variabel dependen, serta membuat kisaran rata-rata dan value dari variabel tersebut.

BAGIAN III
DATA LOADING DAN DATA UNDERSTANDING

3.1. Data Loading

Data loading berarti memuat komponen dari proses ‘ETL’. Dimana ETL merupakan singkatan dari Extraction yang berarti proses melakukan pengambilan dan menggabungkan data, Transformation yang menjadi proses ekstraksi, pembersihan, serta pemformatan data, dan Load, yaitu proses yang berhubungan dengan data yang telah dilakukan pemuatan dalam suatu sistem penyimpanan. Data Loading adalah tahap memuat data yang didapatkan dari sumber ke dalam sistem yang akan dibangun. Data loading merupakan proses dimana penyalinan dan memuat data dilakukan dari suatu *file* sumber ataupun folder ke suatu database. Data loading juga digunakan dalam melakukan ekstraksi basis data. Dalam memuat suatu data, biasanya memiliki perbedaan format antara format aslinya ke tujuan aplikasi, seperti pengubahan pada format .doc atau .txt menjadi .CSV.

Pada tahapan ini penulis mulai masuk ke pengerjaan kode program, dimana dilakukannya *import* library yang diperlukan, dan pembuatan spark session yang diberi nama ‘cluster’. Setelahnya, dapat dilakukan load dataset dengan cara membaca file .csv yang telah disimpan di drive. Karena menggunakan PySpark, proses pembacaan file .csv nantinya akan dimasukkan ke dalam bentuk dataframe spark. Dalam hal ini, penulis menggunakan *method* .read.option().csv() yang disediakan oleh pyspark, penggunaan fungsi *option* dimaksudkan agar dapat melakukan kustomisasi penggunaan header yang telah tersedia pada file .csv, dan untuk fungsi .csv() berguna untuk menampung *file* dataset yang akan dibaca. Hasil dari dataframe tersebut akan disimpan pada variabel mental_health, dan guna memastikannya penulis melakukan pencetakan isi dari variabel tersebut. Berikut isi dari dataframe mental_health:



Timestamp	Age	Gender	Country	state	self_employed	family_history	treatment	work_interfere	no_employees	remote_work	tech_company	benefits	care_
2014-08-27 11:29:31	37	Female	United States	IL	NA	No	Yes	Often	6-25	No	Yes	Yes	N
2014-08-27 11:29:37	44	M	United States	IN	NA	No	No	Rarely	More than 1000	No	No	Don't know	
2014-08-27 11:29:44	32	Male	Canada	NA	NA	No	No	Rarely	6-25	No	Yes	No	
2014-08-27 11:29:46	31	Male	United Kingdom	NA	NA	Yes	Yes	Often	26-100	No	Yes	No	
2014-08-27 11:30:22	31	Male	United States	TX	NA	No	No	Never	100-500	Yes	Yes	Yes	
2014-08-27 11:31:22	33	Male	United States	TN	NA	Yes	No	Sometimes	6-25	No	Yes	Yes	N
2014-08-27 11:31:50	35	Female	United States	MI	NA	Yes	Yes	Sometimes	1-5	Yes	Yes	No	
2014-08-27 11:32:05	39	M	Canada	NA	NA	No	No	Never	1-5	Yes	Yes	No	
2014-08-27 11:32:39	42	Female	United States	IL	NA	Yes	Yes	Sometimes	100-500	No	Yes	Yes	
2014-08-27 11:32:43	23	Male	Canada	NA	NA	No	No	Never	26-100	No	Yes	Don't know	
2014-08-27 11:32:44	31	Male	United States	OH	NA	No	Yes	Sometimes	6-25	Yes	Yes	Don't know	
2014-08-27 11:32:49	29	male	Bulgaria	NA	NA	No	No	Never	100-500	Yes	Yes	Don't know	N
2014-08-27 11:33:23	42	female	United States	CA	NA	Yes	Yes	Sometimes	26-100	No	No	Yes	
2014-08-27 11:33:26	36	Male	United States	CT	NA	Yes	No	Never	500-1000	No	Yes	Don't know	N
2014-08-27 11:33:57	27	Male	Canada	NA	NA	No	No	Never	6-25	No	Yes	Don't know	N
2014-08-27 11:34:00	29	female	United States	IL	NA	Yes	Yes	Rarely	26-100	No	Yes	Yes	N
2014-08-27 11:34:20	23	Male	United Kingdom	NA	NA	No	Yes	Sometimes	26-100	Yes	Yes	Don't know	
2014-08-27 11:34:37	32	Male	United States	TN	NA	No	Yes	Sometimes	6-25	No	Yes	Yes	
2014-08-27 11:34:53	46	male	United States	MD	Yes	Yes	No	Sometimes	1-5	Yes	Yes	Yes	N
2014-08-27 11:35:08	36	Male	France	NA	Yes	Yes	No	NA	6-25	Yes	Yes	No	

nly showing top 20 rows

Gambar 2. Isi dari Variabel mental_health.

3.2. Data Understanding

Data understanding diartikan oleh suatu organisasi sebagai glosarium bisnis, kamus data, serta model lain dari metadata yang menyimpan informasi data. Data understanding mengandung informasi pengetahuan mengenai data. Pada tahapan data understanding, penulis melakukan pemahaman dalam mengamati aset data, serta pengelolaannya. Kemampuan dalam memahami data spesifik diperlukan, dapat dilakukan dengan mengumpulkan, mengelola, serta menggunakan metadata. Data Understanding merupakan tahapan dimana penulis mulai memahami setiap informasi yang ada pada suatu dataset yang digunakan, seperti halnya darimana sumber dataset didapatkan, nama dataset, jumlah baris dan kolom, serta penjelasan terkait data pada tiap kolomnya.

3.2.1. Dataset Information

Dataset yang penulis gunakan dalam proyek ini, yaitu Dataset dengan judul Mental Health in Tech Survey yang diambil pada laman Kaggle [1x]. Dataset tersebut berisikan 1259 data dengan 27 kolom. Berikut merupakan informasi lebih detail dari masing-masing kolom pada dataset:

No	Nama Kolom	Deskripsi
1.	timestamp	Catatan digital tentang waktu terjadinya peristiwa tertentu.
2.	age	Umur.
3.	gender	Jenis Kelamin.

4.	country	Negara.
5.	state	Jika Anda tinggal di Amerika Serikat, di negara bagian atau wilayah mana Anda tinggal?
6.	self_employed	Apakah Anda wiraswasta?
7.	family_history	Apakah Anda memiliki riwayat keluarga penyakit mental?
8.	treatment	Sudahkah Anda mencari perawatan untuk kondisi kesehatan mental?
9.	work_interfere	Jika Anda memiliki kondisi kesehatan mental, apakah Anda merasa itu mengganggu pekerjaan Anda?
10.	no_employees	Berapa banyak karyawan yang dimiliki perusahaan atau organisasi Anda?
11.	remote_work	Apakah Anda bekerja dari jarak jauh (di luar kantor) setidaknya 50% dari waktu?
12.	tech_company	Apakah atasan Anda merupakan seseorang dari perusahaan / organisasi teknologi?
13.	benefits	Apakah atasan Anda memberikan manfaat kesehatan mental?
14.	care_options	Apakah Anda tahu pilihan untuk perawatan kesehatan mental yang diberikan atasan Anda?
15.	wellness_program	Apakah atasan Anda pernah membahas kesehatan mental sebagai bagian dari program kesehatan karyawan?
16.	seek_help	Apakah atasan Anda menyediakan sumber daya untuk mempelajari lebih lanjut tentang masalah kesehatan mental dan cara mencari bantuan?

17.	anonymity	Apakah anonimitas Anda dilindungi jika Anda memilih untuk memanfaatkan sumber daya perawatan kesehatan mental atau penyalahgunaan zat?
18.	leave	Seberapa mudah bagi Anda untuk mengambil cuti medis untuk kondisi kesehatan mental?
19.	mentalhealthconsequence	Apakah menurut Anda mendiskusikan masalah kesehatan mental dengan atasan Anda akan memiliki konsekuensi negatif?
20.	physhealthconsequence	Apakah menurut Anda mendiskusikan masalah kesehatan fisik dengan atasan Anda akan memiliki konsekuensi negatif?
21.	coworkers	Apakah Anda bersedia mendiskusikan masalah kesehatan mental dengan rekan kerja Anda?
22.	supervisor	Apakah Anda bersedia mendiskusikan masalah kesehatan mental dengan atasan langsung Anda?
23.	Mentalhealthinterview	Apakah Anda akan mengemukakan masalah kesehatan mental dengan calon atasan dalam sebuah wawancara?
24.	Physhealthinterview	Apakah Anda akan mengemukakan masalah kesehatan fisik dengan calon atasan dalam sebuah wawancara?
25.	mentalvsphysical	Apakah Anda merasa bahwa atasan Anda menganggap kesehatan mental seserius kesehatan fisik?
26.	obs_consequence	Pernahkah Anda mendengar atau mengamati konsekuensi negatif bagi rekan kerja dengan kondisi kesehatan mental di tempat kerja Anda?
27.	comments	Catatan atau komentar tambahan apa pun.

Tabel 1. Informasi pada Dataset Mental Health in Tech Survey.

BAGIAN IV
EXPLORATORY DATA ANALYSIS

4.1. Definition of Exploratory Data Analysis

Exploratory data analysis merupakan tahapan yang mengacu pada suatu proses kritis ketika melakukan investigasi awal pada suatu data dengan mencari tahu pola, error atau missing value data, serta melakukan pengujian dan pengecekan hipotesis. Dalam exploratory data analysis, peneliti melakukan analisis dan melakukan investigasi pada kumpulan data, dan mengambil ringkasan karakteristik dengan memanfaatkan metode dari visualisasi data. Sehingga pola ataupun pengujian hipotesis yang dilakukan oleh peneliti dipermudah. Dengan diadakannya tahapan Exploratory Data Analysis, bisa membantu dalam melakukan identifikasi, mendeteksi adanya kesalahan dalam memahami data.

4.1.1. Variable Description

Pada tahapan ini, penulis melakukan beberapa pengecekan dari dataframe yang digunakan, seperti halnya menampilkan schema dataframe dalam bentuk tree dengan nama kolom dan tipe data dengan menggunakan method .printSchema(). Selain itu, dilakukannya pengecekan deskripsi dari dataframe menggunakan method .describe().show() guna mengetahui nilai jumlah, rata-rata, standar deviasi, minimal, dan maksimal. Serta, diceknya beberapa isi dari kolom data dengan menggunakan bantuan method .groupBy().count().show(.select().distinct().count(), False), groupby dimaksudkan agar dapat ditentukannya suatu kolom saja, dan .distinct() berguna untuk hanya menampilkan data yang berbeda saja atau jika terdapat data yang sama akan hanya diwakilkan. Kemudian, penulis melakukan pengecekan apakah adanya missing data. Setelah memahami dataframe yang digunakan, didapati informasi bahwa dataset terdiri dari 26 kolom yang terdiri dari 25 kolom dengan tipe data object dan 1 kolom numerik dengan tipe data integer, mayoritas data yang terdapat pada kolom comments merupakan null value karena pengisian kolom yang bersifat opsional, sekitar 60% responden berasal dari Amerika Serikat, banyak negara lainnya yang hanya mempunyai 1 responden. Oleh karena itu,

penulis akan menghapus kolom timestamp, country, state, dan comments karena tidak relevan terhadap pembangunan model prediksi, disini pengguna menggunakan bantuan method .drop().

4.1.2. Feature Engineering

Pada tahapan ini, penulis melakukan pengecekan isi dari masing-masing kolom yang bersifat unique dengan menggunakan method .printSchema(). Setelahnya, penulis memutuskan untuk menerapkan feature engineering pada kolom Age dan Gender. Feature Engineering itu sendiri ialah tahapan dalam meningkatkan kinerja dari algoritma yang digunakan penulis. Dalam tahapan ini, bisa melakukan perubahan variabel yang paling relevan dari suatu data saat pemodelan dilakukan. Adapun fitur dari feature engineering terdiri dari pembuatan fitur, transformasi, ekstraksi, serta pemilihan fitur yang digunakan dalam pengakuratan suatu algoritma. Dalam feature engineering memerlukan beberapa langkah, yaitu melakukan persiapan data, analisis eksplorasi., serta mengidentifikasi dalam penetapan akurasi pada variabel. Yang penulis lakukan yaitu dilakukan pengecekan isi dari masing-masing kolom yang bersifat unique dengan melakukan perintah printScheme() dalam menampilkan skema dataframe. Akan menampilkan nama kolom dan tipe data. Pada tahap ini guna menghapus nilai pada fitur Age yang bernilai kurang dari nol dan lebih dari seratus. Lalu, dataframe yang fitur age nya telah dilakukan pengurangan akan disimpan pada variabel baru untuk menampung dataframe.

4.1.3. Handling Missing Value

Tahap ini berguna untuk menangani missing value atau suatu kolom yang tidak memiliki data. Permasalahan terhadap missing value menjadi hal yang umum dalam pengumpulan data. Missing value sendiri diartikan sebagai nilai yang hilang atau tidak ada dalam suatu kumpulan data. Nilai dari missing value bisa mengakibatkan kurangnya keakuratan pada model nantinya. Maka dibutuhkannya penanganan serta pemahaman terhadap missing value. Banyak algoritma yang bisa digunakan dalam penanganan missing value. Dengan demikian, hal yang pertama penulis lakukan adalah melakukan pengecekan missing value di tiap kolomnya, dan didapati terdapat missing value pada kolom

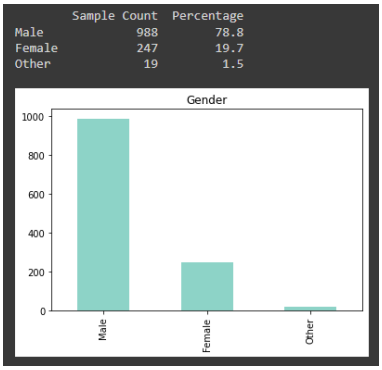
self_employed dan work_interfere. Disini, penulis menetapkan dalam mengganti NA yang merupakan representasi missing value pada kolom work_interfere menjadi “Don’t Know”, setelah itu penulis melakukan pengecekan tiap value berbeda pada kolom tersebut, dan didapati terdapat lima value berbeda yaitu Sometimes, Don't know, Rarely, Often, dan Never. Sedangkan, untuk kolom self_employed missing value akan diubah menjadi “No”, sehingga terdapat dua value pada kolom tersebut yaitu Yes, dan No. Lalu, untuk memastikan apakah proses penanganan missing value berhasil, maka dicek kembali, dan diketahui sudah tidak terdapat missing value lagi pada dataframe.

4.1.4. Data Analysis

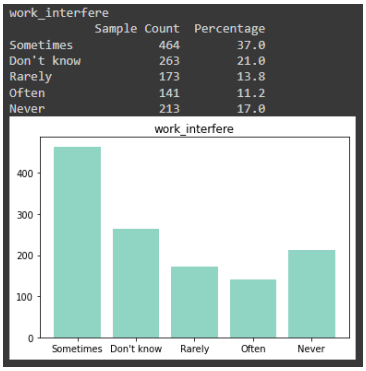
Pada tahap ini, penulis melakukan analisis jenis tiap fitur pada dataframe. Disini penulis menggunakan bantuan atribut dtypes guna mengecek tipe data tiap kolomnya, dan ditambahnya perkondisian jika suatu kolom bertipe integer maka akan dikatakan numerical_features, dan jika bukan maka akan dikatakan sebagai categorical_features. Hasil yang didapati ialah kolom dengan tipe data numerical_features yaitu Age, dan sisanya merupakan categorical_features. Pada tahapan ini, penulis memecah lagi menjadi dua kategori. Berikut uraian lebih jelasnya.

1. Univariate Analysis

   Univariate Analysis merupakan analisis data yang dinilai paling sederhana. Di lihat dari katanya ‘uni’ berarti ‘satu’, berarti pada univariate analysis hanya memiliki satu variabel. Adapun tujuan dari dilakukannya univariate analysis yaitu memahami distribusi nilai dari satu variabel. Ada beberapa jenis dari univariate analysis, seperti bivariate analysis dan multivariate analysis. Ada beberapa cara dalam menunjukkan univariate analysis, yaitu dengan ringkasan statistik, distribusi frekuensi, serta bagan atau charts. Tahapan ini, penulis menggunakan method .toPandas() guna mengubah dataframe spark ke format dataframe pandas agar mempermudah proses analisis. Untuk categorical features penulis menampilkan persentase tiap value dari masing-masing data pada kolom. Misalkan untuk kolom gender dan work_interfere jika dilakukan analisis, maka akan tampil analisa berikut.

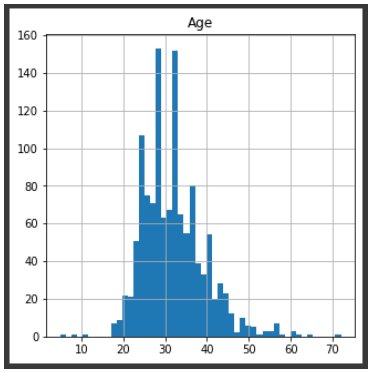


Gambar 3. Tampilan Chart Univariate Analysis Fitur Gender.



Gambar 4. Tampilan Chart Univariate Analysis Fitur work_interfere.

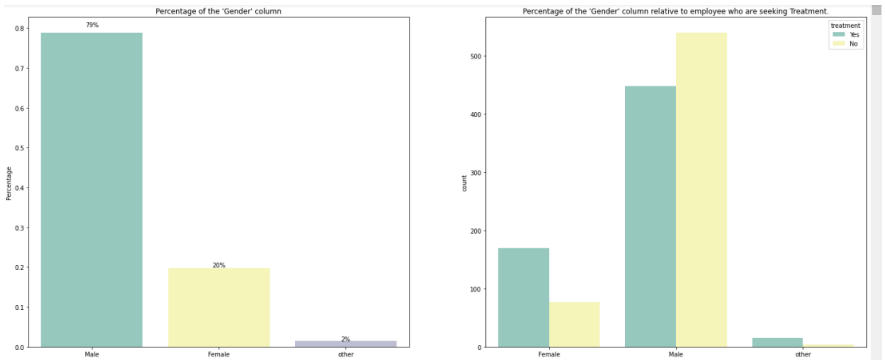
Lalu, untuk numerical features yaitu kolom age akan menampilkan plot histogram berikut.



Gambar 5. Tampilan Chart Univariate Analysis FiturAge.

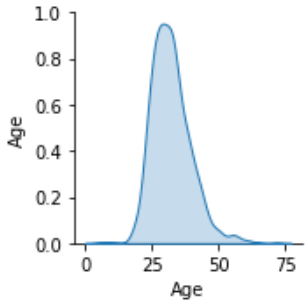
2. Multivariate Analysis

Multivariate analysis merupakan analisis yang melibatkan dua variabel atau lebih, bisa juga memecahkan masalah dimana lebih dari satu variabel dilakukan analisis bersamaan dengan variabel lainnya. Multivariate analysis biasanya banyak digunakan dalam bidang industri, seperti layanan kesehatan. Dengan melakukan multivariate analysis dapat melakukan pertimbangan lebih dari satu faktor variabel yang mempengaruhi variabilitas variabel, sehingga bisa lebih akurat dan realistis. Sebagai contoh disini penulis akan melakukan analisis antara dua fitur berjenis categorical features pada dataframe, yakni age dan treatment. Berikut chart dari multivariate analisis antara age dengan treatment:



Gambar 6. Tampilan Chart Multivariate Analysis Fitur Gender dengan Treatment.

Berdasarkan analisis tersebut, didapatkan nilai persentase relasi antara gender dengan karyawan yang melakukan treatment untuk male sebesar 0.787879, female sebesar 0.196970, dan other sebesar 0.015152. Sedangkan, untuk fitur yang berkategori numerikal, penulis melakukan pengamatan hubungan antar fitur numerik yaitu age dengan fungsi pairplot(). Berikut scatter plot visualisasi keterhubungan fitur age.



Gambar 7. Tampilan Chart Multivariate Analysis Fitur Age dengan Age.

BAGIAN V
DATA PREPARATION

5.1. Technique Used

Teknik yang penulis gunakan pada tahap Data Preparation adalah sebagai berikut:

5.1.1. Encoding Feature Categorical

Pada tahap ini, penulis melakukan proses encoding terhadap fitur kategori dengan menggunakan teknik StringIndexer yang merupakan teknik untuk memetakan kolom string ke kolom indeks. Indeks dimulai dengan 0 dan diurutkan berdasarkan frekuensi label. Jika itu adalah kolom numerik, kolom tersebut pertama-tama akan dimasukkan ke kolom string dan kemudian diindeks oleh StringIndexer.

5.1.2. Standarisasi

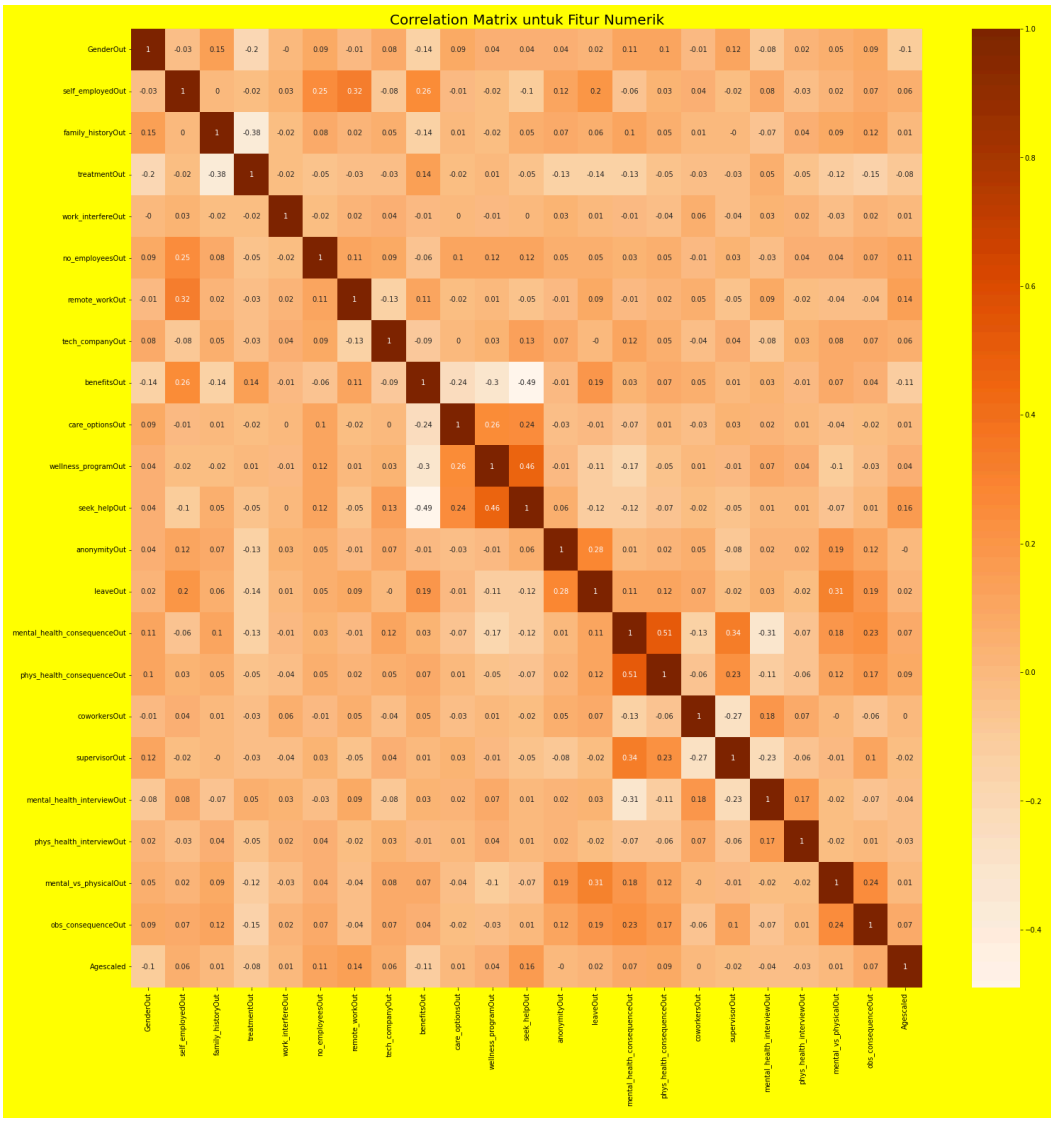
Pada tahap ini, penulis menerapkan proses standarisasi guna menyamakan data pada skala yang relatif sama. Proses tersebut penulis lakukan agar algoritma machine learning menghasilkan performa yang lebih baik dan konvergen lebih cepat. Dalam hal ini, penulis menerapkan proses standarisasi pada fitur numerik dengan menggunakan teknik StandarScaler dari library Scikitlearn. Pada tahapan ini penulis, akan melakukan transformasi nilai fitur age ke bentuk vector menggunakan class VectorAssembler, dan akan disimpan ke kolom baru yang diberi nama “AgeFeature”. Setelahnya, AgeFeature akan distandarisasi dengan menskalakan nilai dari fitur tersebut, dan hasilnya akan ditampung pada kolom baru yaitu “Agescaled”. Setelah, nilai pada kolom Agescaled didapatkan, maka akan dilakukannya penghapusan kolom Age dan AgeFeature dengan menggunakan bantuan method .drop(). Nilai pada fitur Agescaled masih berbentuk vector, maka perlu kita ubah menjadi bertipe data float, disini penulis menggunakan user defined function (UDF) atau fungsi yang didefinisikan sendiri oleh pengguna baik dari isi parameternya, dan tipe data apa yang akan direturn, salah satu parameter yang penulis tetapkan adalah penggunaan class FloatType() guna mengubah nilai vector menjadi float.

5.1.4. Train-Test-Split

Pada tahap ini, penulis melakukan pembagian dataset menjadi data latih dan data uji menggunakan train_test_split dari library Scikitlearn. Pembagian dataset ini bertujuan agar nantinya dapat digunakan untuk melatih dan mengevaluasi kinerja model. Pada proyek ini, 90% dataset digunakan untuk melatih model, dan 10% sisanya digunakan untuk mengevaluasi model. Penulis juga menetapkan nilai random_state sebesar 123. Setelah proses split, penulis mengecek *sample* untuk tiap proses baik data training maupun data testing. Didapatkan *sample* keseluruhan sebesar 1254, lalu dipecah lagi dengan sample training sebanyak 1128, dan sample testing sebanyak 126.

5.1.5. Correlation Matrix

Pada tahap Correlation matrix, akan menunjukkan tabel koefisien korelasi antar variabel. Digunakan juga untuk melakukan peringkasan data sebagai masukan dalam analisis lanjutan. Dalam menampilkan correlation matrix, bisa mempertimbangkan menampilkan seluruh matriks atau tidak, bisa di beri kode warna sesuai pada korelasinya. Untuk projek ini, penulis mendapatkan correlation matrix sebagai berikut:



Gambar 8. Tampilan Correlation Matrix.

BAB VI
MODELING

6.1. Build Models

Pada tahap ini, penulis membangun model prediksi dengan menggunakan 7 algoritma berbeda, yaitu Logistic Regression, K-Nearest Neighbor, Decision Tree, Random Forest, Gradient Boosting, AdaBoost, dan XGBoost. Dikarenakan penggunaan 7 algoritma berbeda, penulis menggunakan dictionary guna menampung setiap algoritma beserta parameter pendukungnya. Dimana, untuk key pada dictionary akan digunakan untuk menyimpan library dari seluruh algoritma yang digunakan, dan untuk valuenya akan digunakan untuk menyimpan pemanggilan class algoritma serta penempatan parameter pendukung. Berikut tabel yang akan menampilkan algoritma dengan parameter pendukungnya:

No.	Algoritma	Parameter
1.	LogisticRegression	-
2.	KNeighborsClassifier	n_neighbors=7
3.	DecisionTreeClassifier	random_state=42
4.	RandomForestClassifier	n_estimators=60 random_state=42
5.	GradientBoostingClassifier	random_state=42
6.	AdaBoostClassifier	-
7.	xgb.XGBClassifier	random_state=0 booster="gbtree"

Tabel 2. Tampilan algoritma beserta parameter pendukungnya.

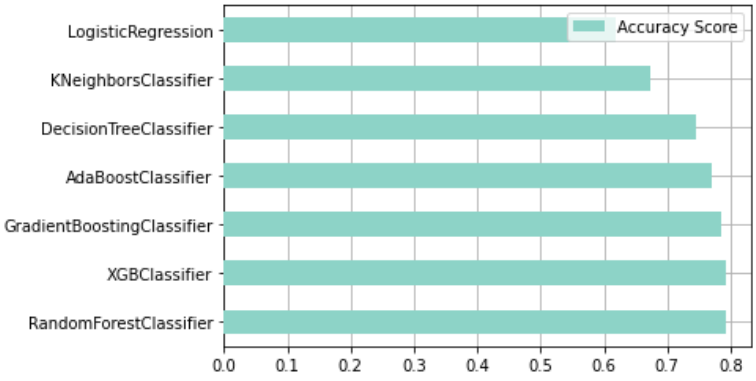
Selanjutnya, penulis melakukan pengecekkan nilai akurasi dari setiap algoritma, yang mana nantinya algoritma dengan hasil nilai akurasi tertinggilah yang akan dilakukan

evaluasi. Pertama kali, penulis melakukan proses fitting dan variabel yang di-fit adalah x_train dan y_train. Kemudian, ditetapkannya nilai pada x_test sebagai variabel prediksi. Karena, sudah terpenuhinya seluruh kebutuhan variabel, maka dilakukannya proses cek akurasi dengan menggunakan variabel y_test dan x_test sebagai prediksi. Kemudian, didapatkannya nilai akurasi setiap algoritma dan untuk algoritma dengan akurasi tertinggi yaitu Random Forest Classifier dan XGBClassifier dengan nilai akurasi sebesar 0.793651. Berikut hasil seluruh akurasi yang didapatkan:

NO.	Algoritma	Nilai akurasi
1.	LogisticRegression	0.619048
2.	KNeighborsClassifier	0.674603
3.	DecisionTreeClassifier	0.746032
4.	RandomForestClassifier	0.793651
5.	GradientBoostingClassifier	0.785714
6.	AdaBoostClassifier	0.769841
7.	xgb.XGBClassifier	0.793651

Tabel 3. Tampilan hasil akurasi dari setiap algoritma.

Selain itu, penulis juga merepresentasikan nilai akurasi tiap algoritmanya dalam bentuk diagram. Berikut diagram nilai akurasi tiap algoritma:

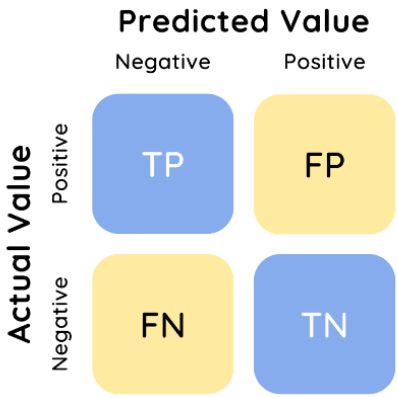


Gambar 9. Tampilan Diagram Perbandingan Akurasi Tiap Algoritma.

BAB VII
EVALUATION

7.1. Evaluation

Dikarenakan model merupakan model klasifikasi, akan dilakukan evaluasi dari model yang telah dibangun dengan menggunakan metode Confusion Matrix. Evaluasi yang dilakukan dengan menggunakan confusion matrix ini akan menghasilkan nilai dari akurasi, presisi, recall, dan kappa guna melakukan pengukuran kesuksesan dari model prediksi dengan algoritma Random Forest yang merupakan model dengan tingkat akurasi tertinggi. Confusion matrix merupakan metode validasi data teks dari jumlah opini negatif dan positif yang tidak seimbang. Berikut merupakan ilustrasi dan penjelasan lebih detail mengenai confusion matrix:



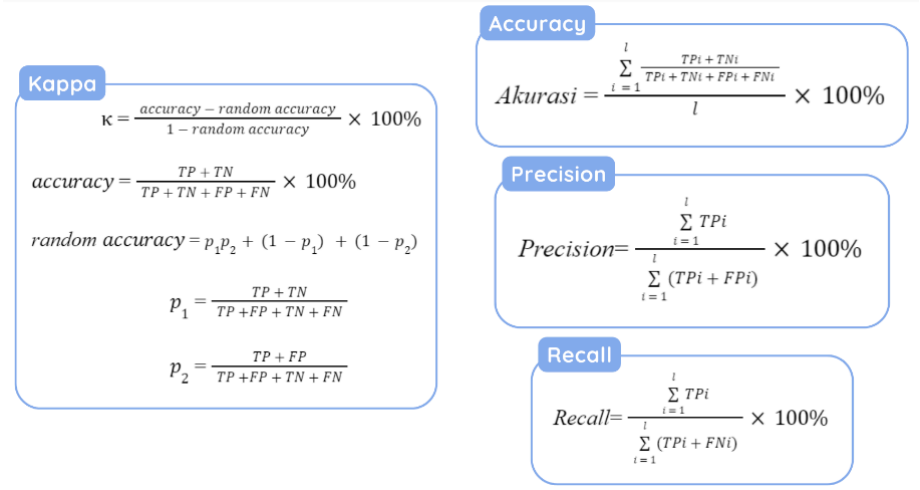
Gambar 10. Ilustrasi Tabel Confusion Matrix.

Berikut merupakan penjelasan dari masing-masing nilai yang terdapat pada confusion matrix:

1. Nilai Prediksi adalah keluaran dari program dimana nilainya Positif dan Negatif.
2. Nilai Aktual adalah nilai sebenarnya dimana nilainya True dan False.
3. True Positive (TP) adalah keluaran program dimana nilainya Positif dan bersifat True.
4. True Negative (TN) adalah keluaran program dimana nilainya Negatif dan bersifat True.

5. False Positive (FP) adalah keluaran program dimana nilainya positif dan bersifat False.
6. False Negative (FN) adalah keluaran program dimana nilainya Negatif dan bersifat False.

Di samping itu, berikut merupakan ilustrasi dan penjelasan lebih detail dari nilai accuracy, recall, precision, dan kappa:



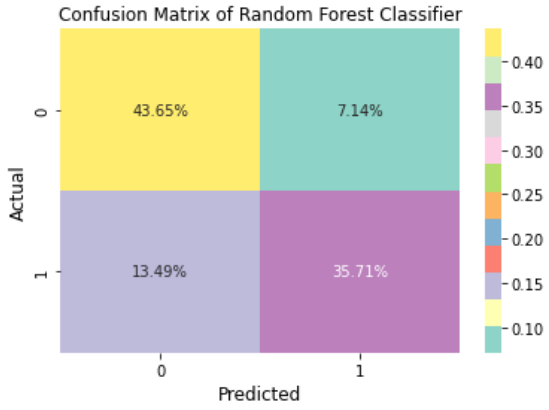
Gambar 10. Rumus Perhitungan Evaluasi.

Berikut merupakan penjelasan dari masing-masing nilai:

1. Accuracy merupakan rasio antara prediksi yang benar dengan seluruh sampel yang diprediksi.
2. Recall merupakan rasio antara prediksi positif yang benar dengan seluruh sampel positif (Actual).
3. Precision merupakan rasio antara prediksi positif yang benar dengan seluruh prediksi positif (Prediction).
4. Kappa merupakan metrik yang membandingkan akurasi yang teramati dengan akurasi yang diharapkan.

Pada projek ini penulis menggunakan tujuh algoritma berbeda, namun pada proses evaluasi ini penulis menetapkan bahwa algoritma dengan nilai akurasi tertinggi saja yang akan dilakukan evaluasi, maka akan dilakukannya evaluasi dengan algoritma Random

Forest Classifier. Hal yang pertama dilakukan adalah membangun confusion matrix, dan didapatkan confusion matrix berikut:



Gambar 11. Tampilan Confusion Matrix Penelitian.

Pada evaluasi projek ini, penulis menggunakan nilai akurasi, presisi, recall, dan kappa. Berikut tabel hasil evaluasi dari algoritma Random Forest Classifier:

No.	Evaluasi	Nilai Evaluasi
1.	Akurasi	0.793651
2.	Precision	0.833333
3.	Recall	0.725806
4.	Kappa	0.586364

Tabel 4. Tampilan hasil evaluasi dari algoritma Random Forest Classifier.

BAB VIII
CLOSING

8.1. Conclusion

Domain proyek yang diambil pada proyek machine learning ini, yaitu Kesehatan dengan judul Predictive Analytics : Prediksi Kebutuhan Perawatan Kesehatan Mental terhadap Seorang Karyawan. Proyek ini menggunakan 7 algoritma yang penulis bandingkan guna mendapatkan model terbaik. Model prediksi terbaik diperoleh dengan menggunakan algoritma Random Forest dengan tingkat akurasi sebesar 0.79 atau 79%. Terakhir, penulis melakukan evaluasi terhadap model tersebut dengan menggunakan confusion matrix. Evaluasi tersebut menghasilkan nilai accuracy sebesar 0.78 atau 78%, nilai precision sebesar 0,73 atay 73%, nilai recall sebesar 0.89 atau 89%, dan nilai kappa sebesar 0.55 atau 55%.

REFERENCES

[1] K. S. Dewi, Buku Ajar Kesehatan Mental, 2012, ISBN : 978-979-097-043-4, http://eprints.undip.ac.id/38840/1/KESEHATAN_MENTAL.pdf#:~:text=Kesehatan%20mental%20menurut%20seorang%20ahli%20kesehatan%20Merriam%20Webster%2C,dalam%20komunitasnya%2C%20da%20n%20memenuhi%20kebutuhan%20hidupnya%20sehari-hari

[2] C. Koopman, K. R. Pelletier, J. F. Murray, C. E. Sharda, M. L. Berger, R. S. Turpin, P. Hackleman, P. Gibson, D. M. Holmes, and T. Bendel, "Stanford Presenteeism Scale: Health Status and Employee Productivity. Journal of Occupational and Environmental Medicine," Journal of Occupational and Environmental Medicine, vol. 44, no. 1, 14-20, 2002, http://www.jstor.org/stable/44995848.

[3] O. Sourcing Mental Illness, LTD., "Mental Health in Tech Survey," Kaggle, 2014. https://www.kaggle.com/datasets/osmi/mental-health-in-tech-survey.

[4] Z. Luna., "One-Hot Encoding Categorical Variables — What is it? Why is it? How is it?," Medium, 2021, https://medium.com/analytics-vidhya/one-hot-encoding-categorical-variables-what-is-it-why-is-it-how-is-it-6fd9ed3a161.

[5] Machine Learning Terapan, Dicoding.

[6] V. Michael., "Machine Learning: Mengenal Logistic Regression," Medium, 2019, https://vincentmichael089.medium.com/machine-learning-2-logistic-regression-96b3d4e7b603.

[7] A. M. Ismail., "Cara Kerja Algoritma k-Nearest Neighbor (k-NN)," Medium, 2018, https://medium.com/bee-solution-partners/cara-kerja-algoritma-k-nearest-neighbor-k-nn-389297de543e.

[8] IYKRA., "Mengenal Decision Tree dan Manfaatnya," Medium, 2018, https://medium.com/iykra/mengenal-decision-tree-dan-manfaatnya-b98cf3cf6a8d.

[9] T. Yiu., "Understanding Random Forest," Medium, 2019, https://towardsdatascience.com/understanding-random-forest-58381e0602d2#:~:text=The%20random%20forest%20is%20a,that%20of%20any%20individual%20tree.

[10] MLMath.io., "Gradient Boosting Algorithm," Medium, 2019, https://ankitnitjsr13.medium.com/gradient-boosting-algorithm-800e5b2bb3e4.

[11] "AdaBoost," Wikipedia, https://en.m.wikipedia.org/wiki/AdaBoost.

[12] "XGBOOST," NVIDIA, https://www.nvidia.com/en-us/glossary/data-science/xgboost/#:~:text=XGBoost%2C%20which%20stands%20for%20Extreme,%2C%20classification%2C%20and%20ranking%20problems.