Laporan Studi Kasus GSB Academy
"Kasus Employee Churn"

Cornelius Justin Satryo Hadi, Muhammad Hanif Pramudya Zamzami, Tulus Setiawan
(Tim "The Jammaths")

---

1. Abstrak

Employee churn adalah kondisi dimana sebuah organisasi/perusahaan (dalam kasus ini perusahaan) mengalami kekurangan karyawan yang diakibatkan oleh keputusan karyawan untuk pindah atau keluar dari perusahaan.

Penulis mendapatkan beberapa fakta pada kasus ini, yaitu para karyawan yang memutuskan untuk pindah didominasi oleh mereka yang bekerja di perusahaan yang berada di wilayah Amerika Serikat bagian kanan, seperti Chicago, Michigan, Toronto, Ohio, West Virginia, Virginia, Washington D.C, dan Maryland.

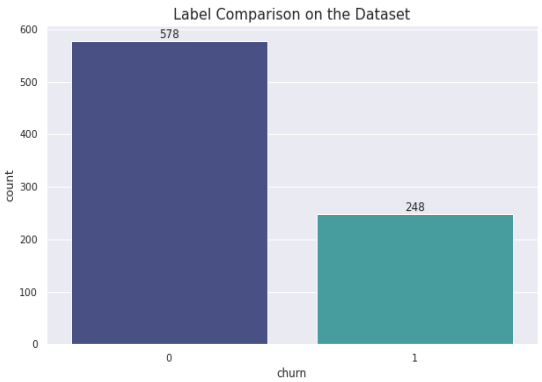
Penulis mencoba untuk mencari solusi untuk mengklasifikasi variabel-variabel independen yang memengaruhi keputusan karyawan untuk pindah atau bertahan dari perusahaan menggunakan *machine learning* dengan algoritma *Support Vector Machine* (SVM). Menggunakan model SVM yang telah penulis buat, diperoleh *recall* sebesar 97%, f1-*score* sebesar 93%, *precision* sebesar 89%, dan akurasi sebesar 96% pada data *test*.

2. Isi

2.1. Exploratory Data Analysis (EDA)

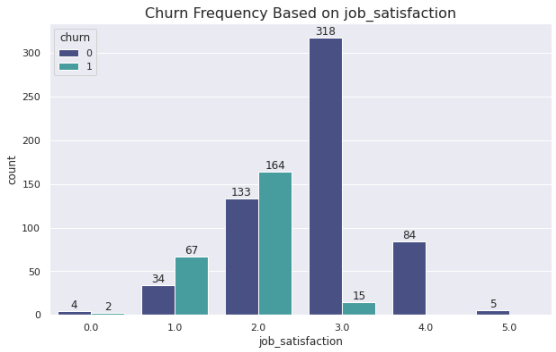
Pada tahap ini, penulis mencoba untuk mengeksplorasi dataset menggunakan beberapa visualisasi supaya bisa mendapatkan *insight-insight* yang sekiranya dapat berguna dalam pemecahan masalah dan mendapatkan solusi yang optimal.

2.1.1. Perbandingan label pada dataset



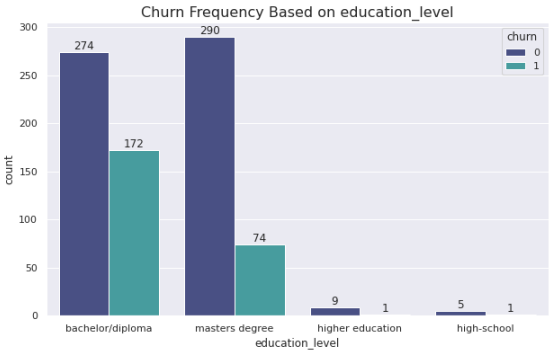
Pada visualisasi di atas, terlihat bahwa tingkat '*Churn*' dari karyawan perusahaan lebih condong ke tidak *Churn* (karyawan tidak pindah) daripada *Churn* (karyawan pindah). Dengan perbandingan rasio Churn dengan tidak Churn yaitu, 309 : 724 (dari 1033 karyawan, 309 memilih untuk pindah), atau sekitar 30% dari total karyawan memilih untuk pindah.

2.1.2. *Insight* dari frekuensi *churn* berdasarkan tingkat kepuasan pekerja



Karyawan yang memberi penilaian tingkat kepuasan kerja dengan skor 0-2 condong untuk Churn (pindah), sedangkan karyawan yang memberi tingkat kepuasan kerja dengan skor 3-5 condong untuk tidak Churn (tidak pindah).

2.1.3. *Insight* dari frekuensi *churn* berdasarkan tingkat pendidikan



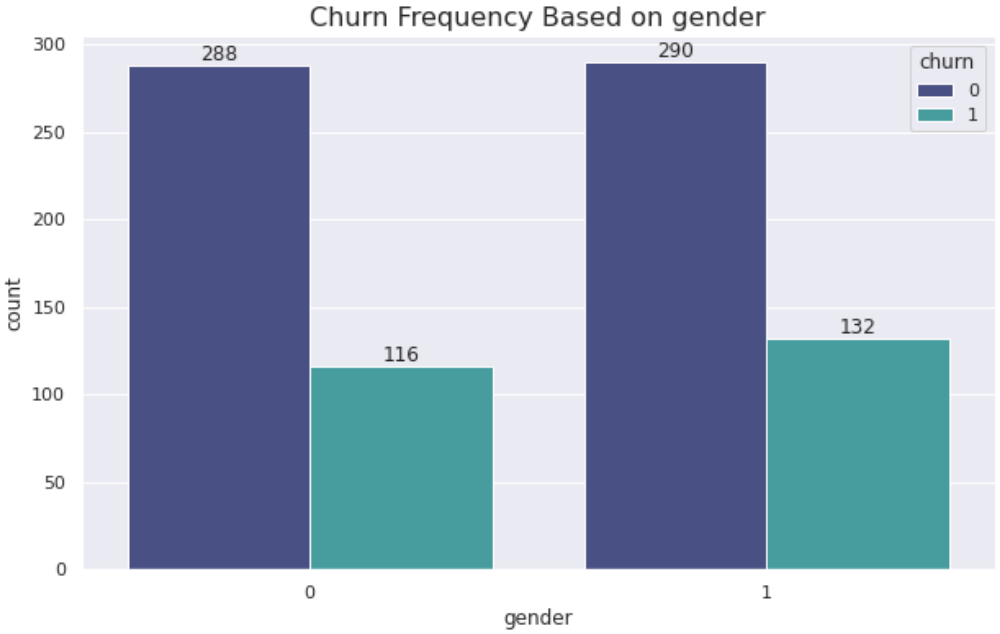
Rasio perbandingan karyawan yang Churn dengan karyawan yang tidak Churn berdasarkan tingkat pendidikannya adalah sebagai berikut ;

- Bachelor/diploma = 86 : 137 (dari 223 karyawan, 86 karyawan memilih untuk pindah), atau sekitar 39% dari total karyawan dengan tingkat pendidikan bachelor/diploma memilih untuk pindah.
- Masters degree = 37 : 145 (dari 182 karyawan, 37 karyawan memilih untuk pindah), atau sekitar 20% dari total karyawan dengan tingkat pendidikan masters degree memilih untuk pindah.
- Higher education = 1 : 9 (dari 10 karyawan, 1 karyawan memilih untuk pindah), atau 10% dari total karyawan dengan tingkat pendidikan higher education memilih untuk pindah.
- High school = 1 : 5 (dari 6 karyawan, 1 karyawan memilih untuk pindah), atau sekitar 17% dari total karyawan dengan tingkat pendidikan high school memilih untuk pindah.

Dari keempat kategori tingkat pendidikan tersebut, dapat disimpulkan karyawan yang memilih untuk pindah didominasi secara berurutan dari tingkat pendidikan bachelor/diploma, masters degree, high school, lalu higher education.

2.1.4. *Insight* dari frekuensi *churn* berdasarkan gender

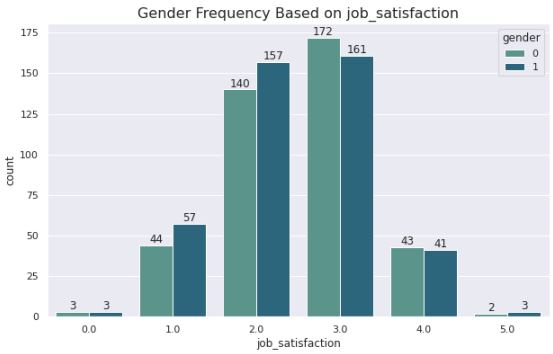
Rasio perbandingan karyawan yang Churn dengan karyawan yang tidak Churn berdasarkan gender-nya adalah sebagai berikut ;



- Perempuan = 29 : 72 (dari 101 karyawan perempuan, 29 karyawan perempuan memilih untuk pindah), atau sekitar 29% karyawan perempuan memilih untuk pindah.
- Laki-laki = 66 : 145 (dari 211 karyawan laki-laki, 66 karyawan laki-laki memilih untuk pindah), atau sekitar 46% karyawan laki-laki memilih untuk pindah.

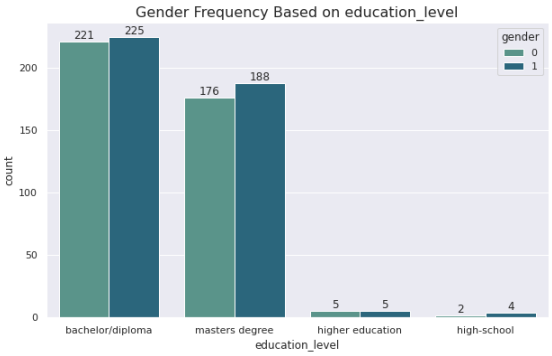
Berdasarkan pengelompokan gender tersebut, dapat disimpulkan karyawan laki-laki cenderung memilih untuk pindah daripada karyawan perempuan.

2.1.5. *Insight* dari frekuensi gender berdasarkan tingkat kepuasan pekerja



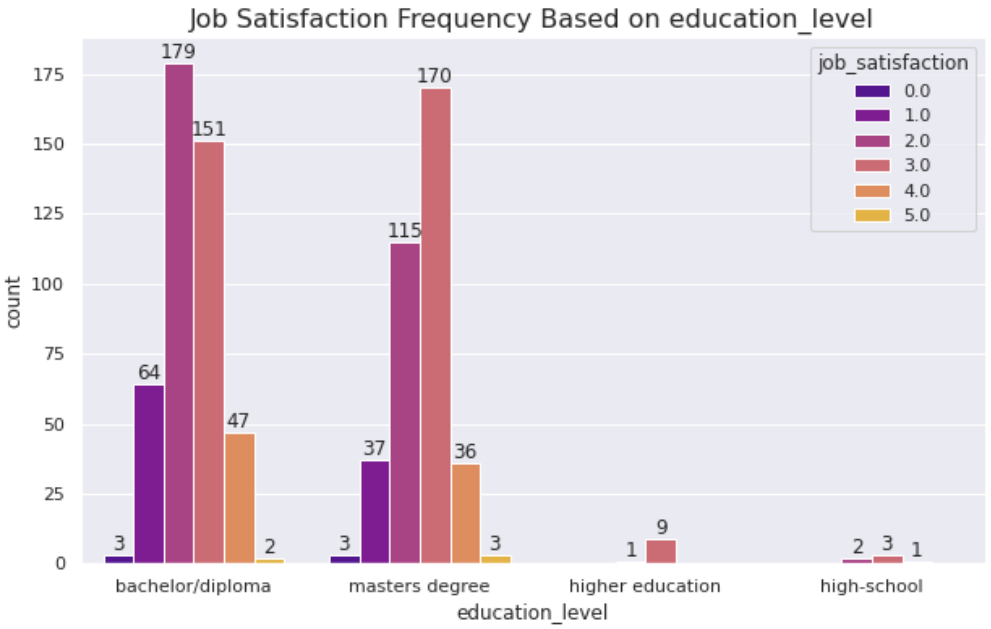
Karyawan laki-laki lebih banyak memberi skor 0-2 daripada karyawan perempuan dalam penilaian tingkat kepuasan kerja, sedangkan karyawan perempuan lebih banyak memberi skor 3-5 daripada laki-laki dalam penilaian tingkat kepuasan kerja.

2.1.6. *Insight* dari frekuensi gender berdasarkan tingkat pendidikan



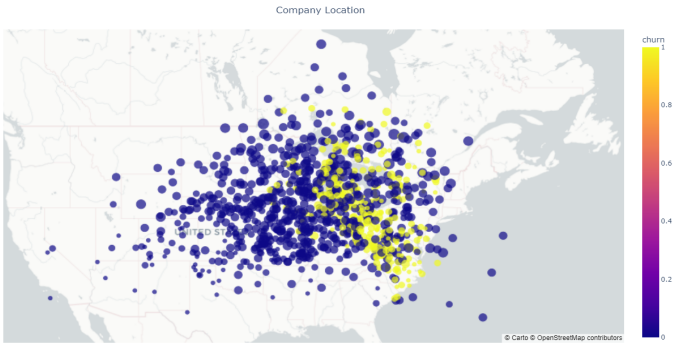
Karyawan dengan tingkat pendidikan bachelor/diploma, masters degree, high school, lebih didominasi oleh laki-laki, sedangkan karyawan dengan tingkat pendidikan higher education setara antara laki-laki dan perempuan.

2.1.7. *Insight* dari frekuensi tingkat kepuasan kerja berdasarkan tingkat pendiidikan



- Karyawan dengan tingkat pendidikan bachelor/diploma paling banyak memberi skor 2 dalam penilaian tingkat kepuasan kerja.
- Karyawan dengan tingkat pendidikan masters degree paling banyak memberi skor 3 dalam penilaian tingkat kepuasan kerja.
- Karyawan dengan tingkat pendidikan higher education paling banyak memberi skor 3 dalam penilaian tingkat kepuasan kerja.
- Karyawan dengan tingkat pendidikan high-school paling banyak memberi skor 3 dalam penilaian tingkat kepuasan kerja.

2.1.8. *Insight* dari lokasi perusahaan

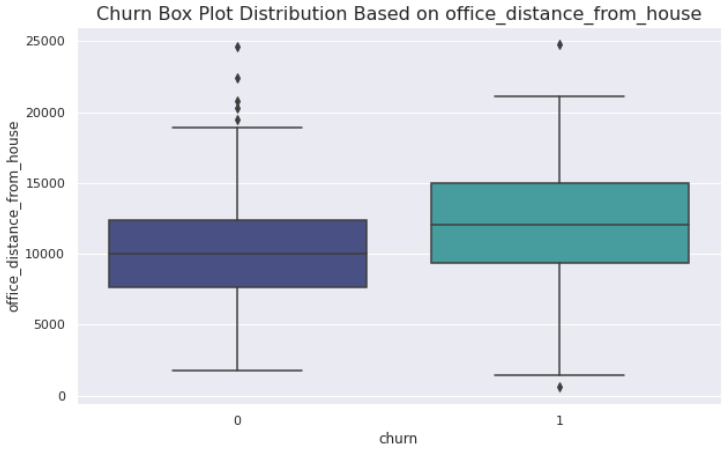


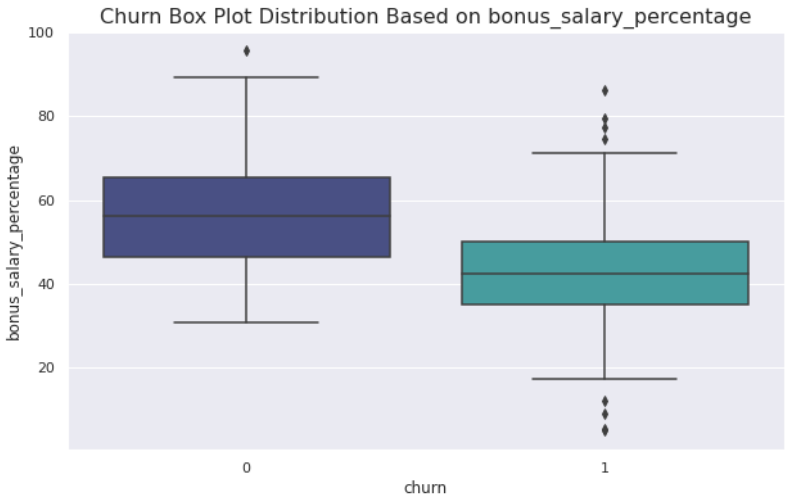
Persebaran lokasi dari perusahaan tempat para karyawan bekerja terletak di benua Amerika tepatnya di negara Amerika Serikat.

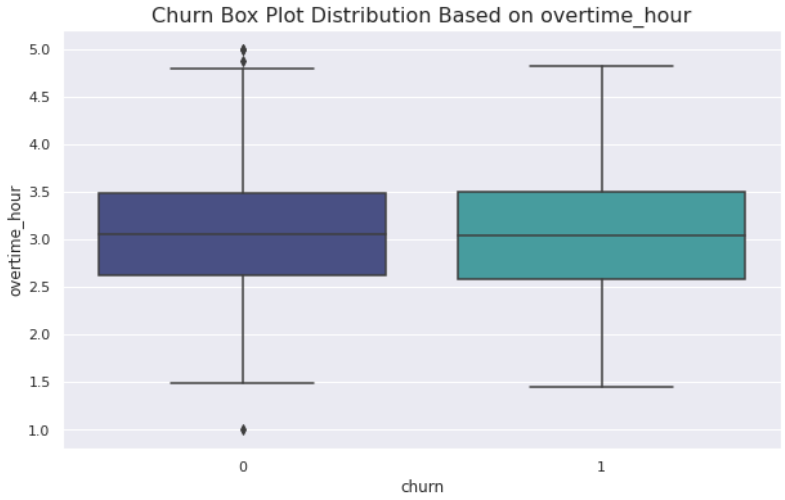
2.1.9. *Insight* dari titik tengah persebaran perusahaan

Jika kita lihat dari titik pusat persebaran, maka banyaknya karyawan yang memutuskan untuk pindah dari perusahaan tempat ia bekerja berasal dari sebelah kanan titik pusat persebaran, seperti di area Chicago, Michigan, Toronto, Ohio, West Virginia, Virginia, Washington D.C, Maryland. Kemudian, untuk wilayah sisanya para karyawan memilih untuk bertahan di perusahaan.

2.1.10. *Insight* dari fitur-fitur data numerik menggunakan *box plot*

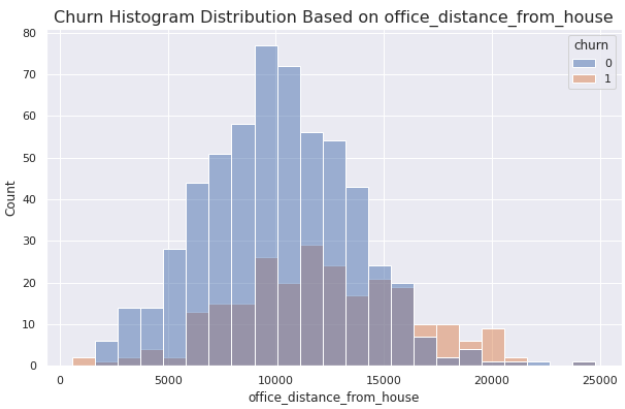






Untuk fitur *office_distance_from_house, bonus_salary_percentage, overtime_hour* yang disandingkan dengan fitur Churn terlihat bahwa terdapat beberapa data outlier pada ketiga fitur tersebut, akan tetapi disini data outlier tersebut tidak akan ditangani/diganti/dihilangkan karena jika kita lihat pada visualisasi histogram di bawah ini, terlihat bahwa data outlier tersebut tidak terlalu jauh perbedaannya dengan data di sekitarnya sehingga penulis mengambil kesimpulan *data outlier* tersebut tidak terlalu sensitif dan tidak perlu untuk ditangani atau dengan kata lain tetap orisinil.

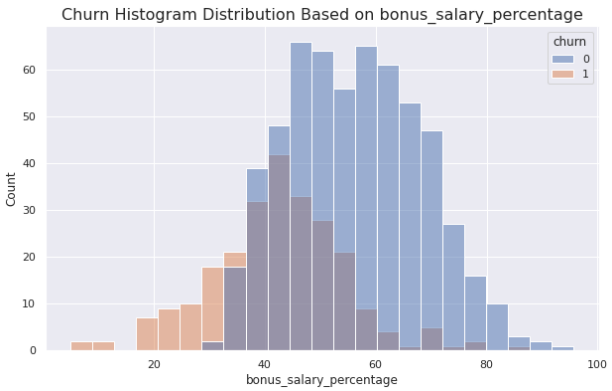
2.1.11. *Insight* dari distribusi histogram fitur *office_distance_from_house* berdasarkan frekuensi churn



Dari distribusi histogram disamping, terlihat bahwa karyawan yang memutuskan untuk pindah didominasi oleh karyawan yang memiliki jarak tempuh dari rumah ke kantor di atas 5000 meter, tetapi

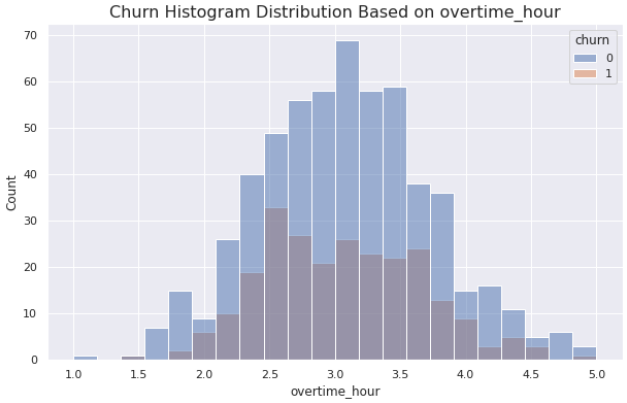
jumlahnya tidak terlalu banyak jika dibandingkan dengan karyawan yang menetap untuk jarak tempuh yang sama.

2.1.12. *Insight* dari distribusi histogram fitur *bonus salary percentage* berdasarkan frekuensi churn



Dari distribusi histogram disamping, terlihat bahwa karyawan yang memilih untuk pindah sebagian besar mendapatkan bonus gaji yang tidak terlalu besar jika dibandingkan dengan karyawan yang menetap.

2.1.13. *Insight* dari distribusi histogram fitur *overtime hour* berdasarkan frekuensi churn



Dari distribusi histogram disamping, terlihat bahwa banyaknya karyawan yang memilih untuk bertahan lebih banyak jumlahnya dibandingkan dengan yang pindah berdasarkan distribusi jam lembur yang sama.

2.2. *Data Preparation*

2.2.1. Standardisasi Data

Ide dari standardisasi data adalah menyeragamkan nilai-nilai data yang pada penginputan formatnya tidak konsisten menggunakan suatu format tertentu, hingga seluruh data menjadi standar. Metode yang penulis gunakan untuk standardisasi adalah *Standard Scaler*, dari *library scikit-learn*. Proses *Standard Scaler* membutuhkan *mean* dan standar deviasi dari nilai-nilai data suatu kolom, setelahnya nilai-nilai data tersebut diubah menjadi data yang berdistribusi normal (memiliki μ = 0 dan σ = 1). Rumus metode *Standard Scaler* sebagai berikut ;

$$x_{new(i)} = \frac{x_{(i)} - \mu}{\sigma}$$

ket:

x_i = nilai data kolom *x* pada index ke - *i*

μ = mean data kolom *x*

σ = standar deviasi kolom *x*

$x_{new(i)}$ = nilai data baru kolom *x* pada index ke - *i*

Penulis melakukan standardisasi data menggunakan metode *Standard Scaler* pada kolom ; *office distance from house, bonus salary percentage, overtime hour, company latitude, company longitude,* karena untuk mempermudah proses *training data* tipe kontinu, agar data yang digunakan tidak memiliki penyimpangan yang besar.

2.2.2. *Encoding* Data

Encoding adalah proses mengubah data bertipe kategorik (format *string*), diubah menjadi bentuk bilangan (format *integer*), hal ini dilakukan karena komputer tidak dapat memproses data bertipe kategorik. Metode yang penulis gunakan adalah *one-hot encoding,* dari *library pandas,* Ide dari metode ini adalah merepresentasikan data bertipe kategori sebagai vektor biner yang bernilai *integer*, 0 dan 1, dimana semua elemen akan bernilai 0 kecuali satu elemen yang bernilai 1, yaitu elemen yang memiliki nilai kategori tersebut. Dalam penerapannya, Setiap kategori pada suatu kolom akan diubah menjadi kolom baru yang bernilai 0 (untuk elemen yang tidak memiliki nilai kategori tersebut) atau 1 (untuk elemen yang memiliki nilai kategori tersebut).

Penulis melakukan *encoding data* menggunakan metode *one hot encoding* pada kolom ; *education level*, karena untuk mempermudah proses training data tipe kategorik, agar data yang digunakan direpresentasikan sebagai vektor biner yang bernilai integer.

2.3. Modelling

Penulis menggunakan *Support Vector Machine* (SVM) sebagai model *machine learning* untuk membuat prediksi pada dataset *employee churn*, karena masalah pada dataset ini adalah untuk memprediksi label berbentuk klasifikasi biner. SVM merupakan salah satu model yang sangat baik untuk membuat prediksi klasifikasi biner.

Model SVM paling optimal yang penulis dapatkan adalah SVM dengan kernel *Radial Basis Function* (RBF) serta hyperparameter regularisasi $C = 8.75$ dan $\gamma = 0.085$. Untuk mengevaluasi model tersebut, penulis menggunakan metrik *recall*, f1-*score*,

precision, dan akurasi. Tingkat metrik yang diutamakan berurutan, dimulai dari *recall* (paling diutamakan) sampai akurasi (paling tidak diutamakan).

Urutan metrik tersebut dipilih karena penulis lebih mementingkan *false negative* dibanding *false positive*, karena ketika *false negative* bernilai tinggi, maka perusahaan akan berekspektasi (berdasarkan prediksi model) bahwa banyak karyawan yang tidak ingin berhenti dari kantornya, tetapi realitanya karyawan akan berhenti dari kantor. Hal tersebut menyebabkan perusahaan akan mengeluarkan biaya yang lebih banyak serta waktu dan tenaga untuk mencari dan merekrut karyawan baru, karena adanya kesalahan model dalam memprediksi karyawan yang akan berhenti dari kantor.

- *False Positive* dari kasus ini adalah ketika model memprediksi bahwa karyawan akan berhenti dari kantor asal, sedangkan realitanya karyawan tidak berhenti dari kantor asal. Hal ini disebabkan karena *class* 1 (positif) pada data menandakan bahwa karyawan berhenti, sedangkan *class* 0 (negatif) pada data menandakan bahwa karyawan tidak berhenti.

- *False Negative* dari kasus ini adalah ketika model memprediksi bahwa karyawan tidak berhenti dari kantor asal, sedangkan realitanya karyawan berhenti dari kantor asal.

Tahapan dalam pembuatan model adalah sebagai berikut.

2.3.1. Membandingkan SVM dengan beberapa algoritma ML lainnya

Model	Nilai *Recall*
SVM	94,33%
Gradient Boosting	93,93%
Random Forest	93,13%
KNN	92,31%
Logistic Regression	87,89%

Pertama, penulis membandingkan SVM dengan beberapa algoritma ML lainnya, antara lain *Logistic Regression*, *Random Forest*, KNN, dan *Gradient Boosting*, menggunakan *cross validation* pada *training set*. Hyperparameter yang digunakan pada setiap model adalah hyperparameter *default* dari scikit-learn. Metrik yang digunakan pada tahap pertama ini adalah *recall*, sehingga penulis pilih SVM sebagai model yang akan ditinjau lebih lanjut, karena mendapatkan nilai *recall* tertinggi, yaitu 94,33%.

2.3.2. Hyperparameter *Tuning* pada SVM

Hyperparameter yang penulis *tuning* pada SVM adalah hyperparameter yang berfungsi untuk regularisasi, yaitu C dan gamma (γ), dengan metode *grid search* dengan *cross validation*. Hyperparameter *tuning* dilakukan sebanyak tiga kali secara terpisah untuk mengurangi lama waktu komputasi. Metrik yang ditinjau untuk menilai hyperparameter adalah *recall*. Jika terdapat lebih dari satu kombinasi hyperparameter yang menghasilkan *recall* yang sama, maka akan peringkat dari hyperparameter akan diurutkan berdasarkan lama waktu *training* dan lama waktu model dalam memprediksi. Berikut adalah nilai - nilai hyperparameter yang di-*tuning*, serta peringkat, *recall*, lama waktu *training*, dan lama waktu prediksi.

Tuning pertama:

C	γ (gamma)	Lama Waktu *Training* (detik)	Lama Waktu Prediksi (detik)	*Recall*	Peringkat
20	0.02	0.013793	0.004142	0.951429	1
2	auto	0.017184	0.006052	0.947347	2
2	scale	0.014418	0.004915	0.943347	3
200	0.02	0.019571	0.005032	0.939347	4
200	0.002	0.015205	0.004546	0.935102	5

Tuning kedua:

C	γ (gamma)	Lama Waktu *Training* (detik)	Lama Waktu Prediksi (detik)	*Recall*	Peringkat
10	0.08	0.028660	0.008530	0.955592	1
16	0.06	0.030894	0.005770	0.951592	2
10	0.06	0.020013	0.007666	0.951510	3
20	0.04	0.022767	0.005704	0.951510	4
16	0.04	0.022829	0.007864	0.951510	5

Tuning ketiga:

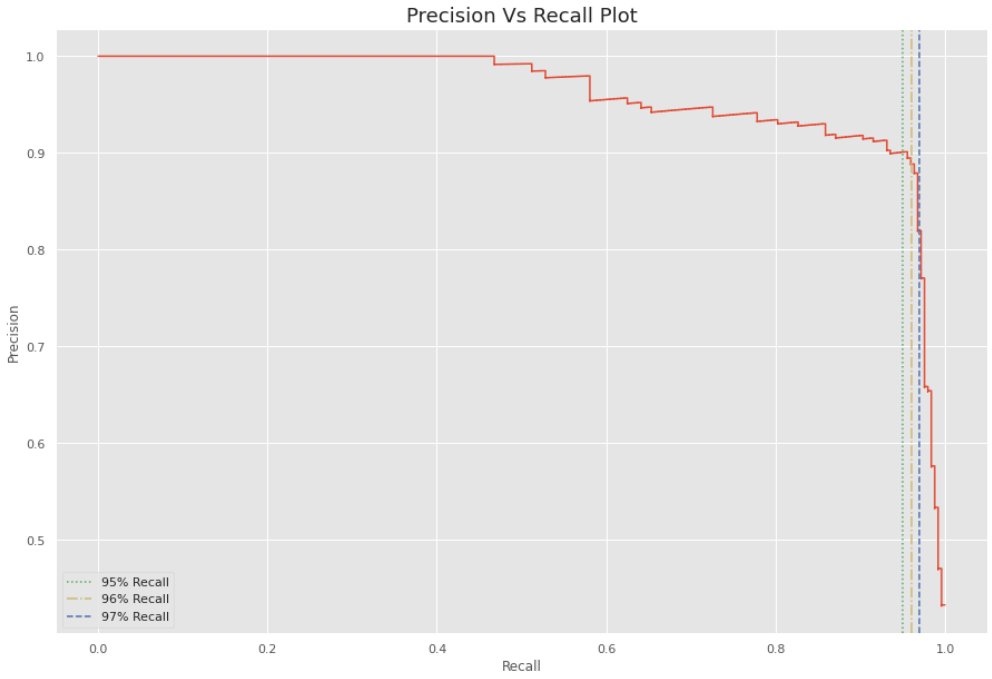
C	γ (gamma)	Lama Waktu *Training* (detik)	Lama Waktu Prediksi (detik)	*Recall*	Peringkat
8.75	0.085	0.015647	0.004952	0.955592	1
10	0.08	0.037646	0.008928	0.955592	2
9	0.085	0.014993	0.004092	0.951592	3
9	0.08	0.015151	0.004703	0.951592	4
8.5	0.085	0.016045	0.005964	0.951592	5

Diperoleh hyperparameter terbaik adalah $C = 8.75$ dan $\gamma = 0.085$.

2.3.3. Mencari nilai *threshold* SVM untuk memaksimalkan *recall*.

Threshold merupakan nilai batas keputusan dari SVM dalam memprediksi label atau *class* dari suatu sampel. *Threshold* akan berpengaruh dalam *precision/recall trade-off*. *Precision/recall trade-off* adalah suatu kondisi dimana kita tidak bisa mendapatkan nilai yang sempurna untuk kedua nilai *precision* dan *recall* secara sekaligus, sehingga kita perlu "mengorbankan" salah satu metrik tersebut (*precision*, dalam kasus penulis) demi mendapatkan nilai yang maksimal pada metrik yang lainnya (*recall*, dalam kasus penulis). Namun, "mengorbankan" bukan berarti kita tidak memperhatikan secara mutlak, tetapi kita juga perlu memperhatikan metrik yang dikorbankan agar kesalahan pada model tidak besar. Disinilah peranan dari *threshold*. Kita dapat mencari nilai *threshold* yang optimal sedemikian sehingga nilai *recall* bisa diperoleh secara maksimal, dengan nilai *precision* yang baik juga.

Berikut adalah plot *precision vs recall* yang didapatkan oleh model SVM dengan $C = 8.75$ dan $\gamma = 0.085$ (tiap titik pada grafik diperoleh menggunakan *threshold* yang berbeda).



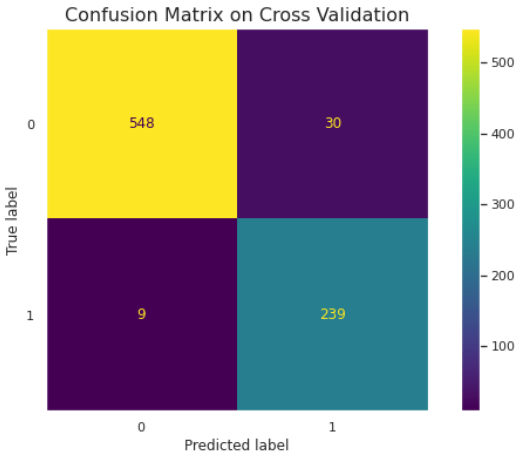
Pada plot diatas, terlihat bahwa hingga *recall* mencapai 95%, *precision* cenderung stabil (diatas 90%). Namun, ketika *recall* lebih dari 96%, *precision* menurun cukup drastis. Pada saat *recall* tepat sebesar 96%, *precision* memiliki nilai yang masih cukup bagus (hampir mendekati 90%). Sehingga, penulis menggunakan *threshold* yang menghasilkan *recall* sebesar 96% tersebut, yaitu (*threshold*) sebesar -0.2269 (cara untuk memperolehnya terdapat pada file .ipynb).

2.3.4. Evaluasi model pada *cross validation* data *training*

Dengan menggunakan *threshold* yang sudah diperoleh sebelumnya, serta dengan menggunakan metode *cross validation* 5-*fold* pada data *training*, diperoleh nilai evaluasi sebagai berikut:

- *Recall*: 96%
- F1-*Score*: 92%
- *Precision*: 89%
- Akurasi: 95%

Dengan *confusion matrix* sebagai berikut:



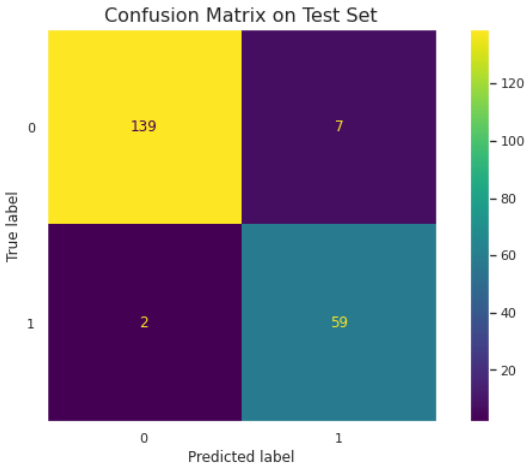
Diperoleh FP (*false positive*) sebanyak 30 dan FN (*false negative*) sebanyak 9. Hasil - hasil tersebut sudah sangat baik karena telah memiliki FN yang cukup kecil dan *recall* yang sangat besar (mencapai lebih dari 95%). Sehingga, model SVM ini adalah model final penulis.

2.3.5. Evaluasi model pada data *test*

Pada *test set* diperoleh nilai evaluasi sebagai berikut:

- *Recall*: 97%
- F1-*score*: 93%
- *Precision*: 89%
- Akurasi: 96%.

Dengan *confusion matrix* sebagai berikut:



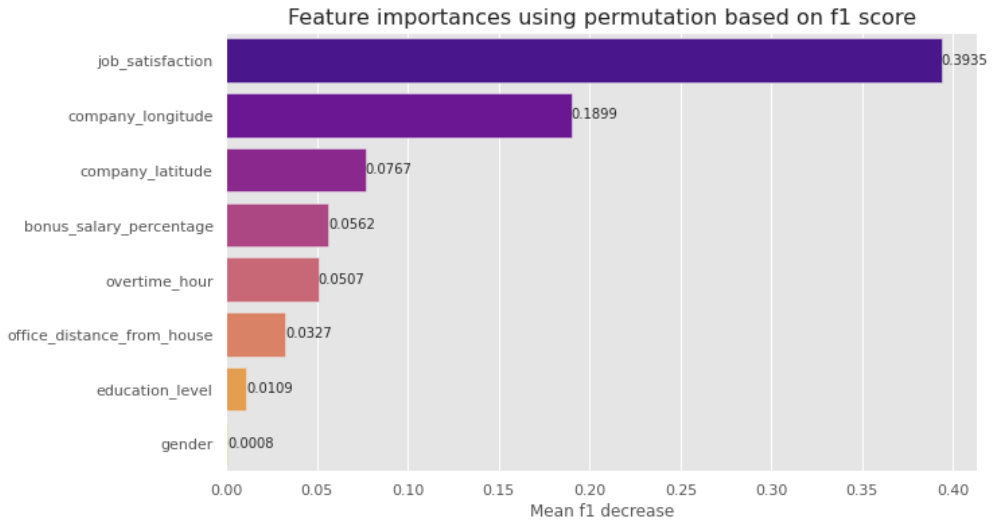
Diperoleh FP sebanyak 7 dan FN sebanyak 2.

2.4. Feature Importances

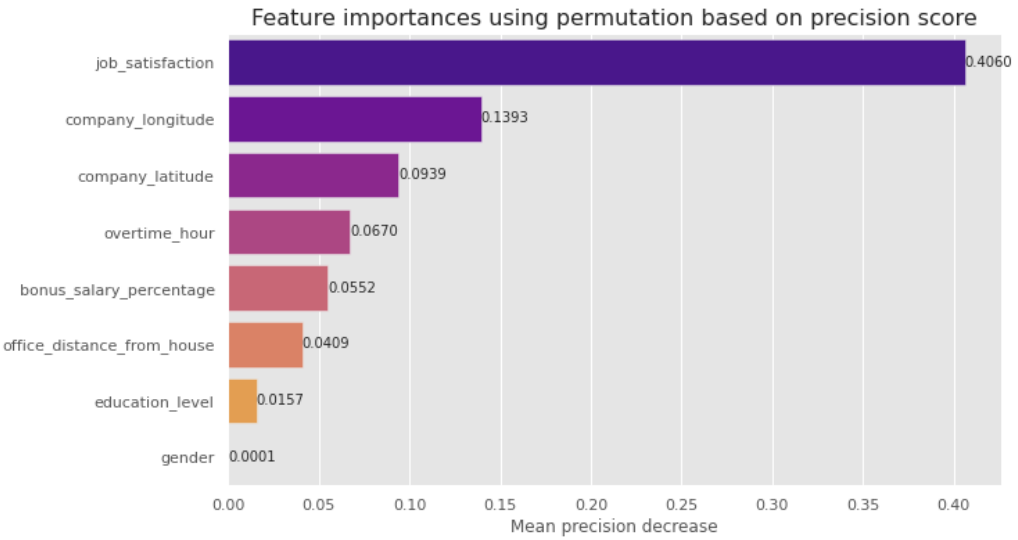
Setelah proses *modelling* selesai, penulis menggunakan metode *permutation importances* untuk mengetahui kepentingan masing - masing fitur untuk model dalam memprediksi seorang karyawan akan pindah dari kantor atau tidak. *Permutation*

importances menghitung penurunan nilai metrik yang diset ketika suatu fitur dari dataset diacak (*shuffle*). Metrik yang penulis gunakan untuk metode ini adalah f1-*score*, *precision*, dan *recall*. Diperoleh, 5 fitur yang paling mempengaruhi f1-*score* dan *recall*, secara berurutan, berdasarkan metode *permutation importances* adalah job_satisfaction, company_longitude, company_latitude, bonus_salary_percentage, dan overtime_hour. Sedangkan, untuk *precision* 5 fitur teratasnya sama dengan f1-*score* dan *recall*, tetapi terdapat perbedaan urutan untuk overtime_hour dan bonus_salary_percentage (overtime_hour lebih mempengaruhi).

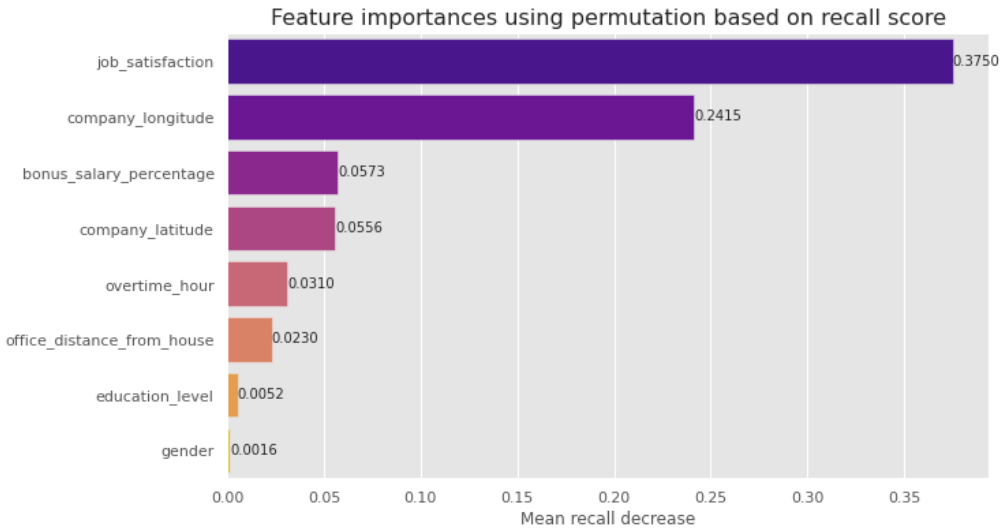
Untuk lebih jelas, kepentingan dari masing - masing fitur dapat dilihat pada plot berikut (semakin besar nilai menandakan fitur semakin penting).



(Plot kepentingan fitur berdasarkan f1-*score*)



(Plot kepentingan fitur berdasarkan *precision*)



(Plot kepentingan fitur berdasarkan *recall*)

3. Kesimpulan

Berdasarkan model yang telah dibuat, dapat disimpulkan bahwa model SVM dengan kernel *Radial Basis Function* (RBF) serta hyperparameter $C = 8.75$ dan $\gamma = 0.085$, telah baik dalam memprediksi kasus *employee churn*, hal ini dibuktikan dari evaluasi model dalam memprediksi data test diperoleh *recall* sebesar 97%, f1-*score* sebesar 93%, dan *precision* sebesar 89%. Dengan begitu, model dapat melakukan antisipasi agar *employee* tersebut tidak jadi *churn.*

Selain itu, dengan menggunakan metode *permutation importances*, diperoleh lima fitur terpenting dalam dataset adalah job_satisfaction, company_longitude, company_latitude, bonus_salary_percentage, dan overtime_hour. Sehingga, dapat direkomendasikan pada perusahaan - perusahaan untuk lebih memperhatikan tingkat kepuasan karyawan terhadap pekerjaan, letak geografis kantor, jam lembur karyawan, dan bonus gaji karyawan untuk mencegah adanya karyawan yang berhenti dari kantor.