Nama :
- Ghina Khoerunnisa (1301181066)
- Delvanita Sri Wahyuni (1301184014)

Kelass : IF 42 04

Laporan Tugas Besar Pembelajaran Mesin Tahap Kedua (Classification)

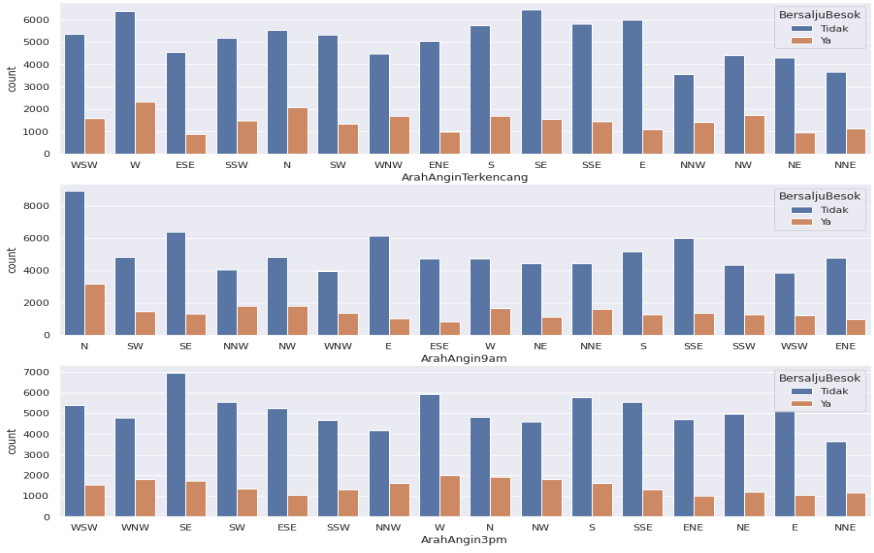
1. Formulasi Masalah

   Pada tugas ini terdapat dataset “salju_test.csv” dan “salju_train.csv” dengan jumlah 23 kolom dan 109095 baris untuk salju_train dan 18182 baris untuk salju_test. Dataset tersebut akan dilakukan classification untuk mencari nilai optimum sebagai acuan memprediksi apakah besok salju turun atau tidak.

2. Eksplorasi dan Persiapan Data

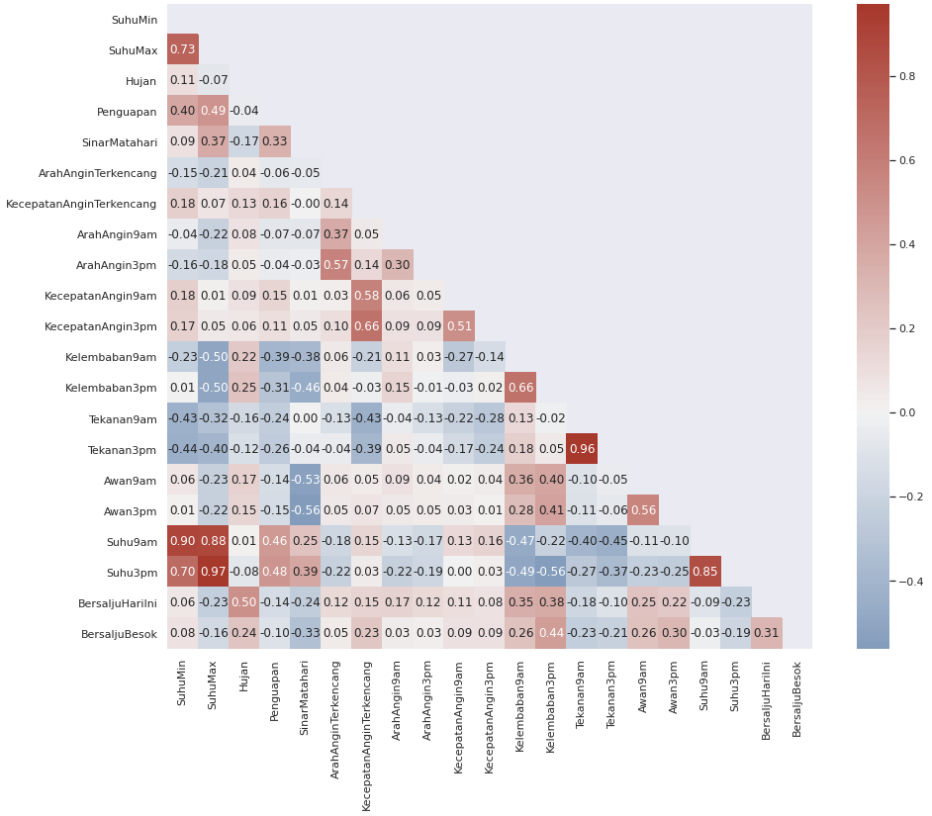
   Untuk melakukan *classification* terhadap dataset salju ini dilakukan eksplorasi dan preparasi data terlebih dahulu dengan melakukan Check Missing Values terhadap data yang akan digunakan agar kita tahu apakah adanya kehilangan data atau tidak. Karena pada kasus ini terdapat missing values maka dilakukan handle missing values dengan mendrop beberapa baris dalam atribut subset yang memiliki nilai “NaN” yang tidak banyak dan mengganti nilai atribut yang bernilai “NaN” lainnya menjadi nilai mean untuk atribut numerik dan mengganti nilai kategorikal menjadi nilai modus berdasarkan pengelompokannya. Banyaknya data sehingga dibutuhkan juga untuk mengecek duplikat data agar tidak ada data yang sama. Pada dataset ini kolom “BersaljuBesok” merupakan data target atau data label yang merupakan kelasnya.

   Melakukan eksplorasi data dengan hanya memilih beberapa atribut untuk memahami data di setiap bin kategori variabel dan data yang dipilih untuk dilihat adalah:



Pada gambar ini terlihat bahwa perbandingan antara hubungan arah angin dan label sangat jauh berbeda. Dalam eksplorasi data ini menunjukkan bahwa hasil untuk berpotensi “BersaljuBesok” itu lebih tinggi pada bar class “Tidak”.

Dataset ini terdiri dari 23 atribut dan terdapat 5 atribut yang bertipe object kategori yaitu: Arah Angin Terkencang, Arah Angin 9am, Arah Angin 3pm, Bersalju Hari Ini, dan Bersalju Besok. Kelima atribut tersebut diubah terlebih dahulu menjadi numerik agar dapat diobservasi korelasinya dengan atribut numerik lainnya. Untuk melihat korelasi antara data fitur dan label dapat menggunakan *plot correlation matrix*, hasilnya adalah :



Pada gambar diatas terdapat *plot correlation matrix* yang terdiri dari berbagai atribut, untuk warna merah kuat itu menandakan bahwa korelasi positif yang kuat, sedangkan warna biru menandakan sebaliknya yaitu korelasi negatif. Sedangkan untuk warna yang terang menuju putih menandakan korelasi yang rendah atau tidak adanya korelasi antar atribut. Berdasarkan plot dapat dilihat bahwa ada beberapa atribut fitur yang memiliki korelasi tinggi dengan fitur lainnya. Fitur dengan korelasi tinggi lebih bergantung secara linier dan karenanya memiliki efek yang hampir sama pada variabel dependen. Jadi, jika dua fitur memiliki korelasi tinggi, kita dapat melepaskan salah satu dari dua fitur tersebut. Sedangkan jika fitur memiliki korelasi yang tinggi dengan target/label maka fitur dapat dipertahankan.

Feature Engineering adalah memilih fitur atau membuat fitur baru agar model machine learning dapat bekerja lebih akurat dalam memecahkan masalah serta memilih atribut yang akan digunakan dan men-drop atribut yang tidak

diperlukan dicontoh ini atribut yang di drop adalah: Tanggal, KodeLokasi, Suhu3pm, Suhu9am, SuhuMax, KecepatanAngin3pm, KecepatanAngin9am, Kelembaban9am, Tekanan9am, ArahAnginTerkencang, ArahAngin3pm, ArahAngin9am, dan Awan9am. Justifikasi dari pemilihan fitur ini adalah untuk Tanggal dan KodeLokasi, kolom tersebut tidak memiliki hubungan dengan data target/label. Suhu3pm, Suhu9am, SuhuMax, dan SuhuMin memiliki korelasi yang tinggi antar fiturnya sehingga dapat didrop dan karena SuhuMin memiliki korelasi positif tertinggi dengan data label maka ketiga kolom lainnya dihapus. Begitu juga dengan kolom kecepatan angin, kelembaban, tekanan, arah angin, dan awan.

Setelah memilih data fitur, dapat dilihat bahwa jangkauan datanya berbeda-beda. Sehingga perlu dilakukan scaling data. Pada tugas ini digunakan teknik scaling data dengan standarisasi. Standarisasi berfungsi untuk mengubah data sedemikian rupa sehingga distribusi yang dihasilkan memiliki rata-rata 0 dan simpangan baku 1. Setelah melakukan scaling, data harus dipisahkan features dengan labelnya supaya data dapat diproses oleh library model yang dipilih.

3. Pemodelan

   Klasifikasi merupakan salah satu tipe dari *Supervised Learning*. *Supervised Learning* itu sendiri merupakan metode dalam pembelajaran mesin yang mana algoritma tersebut seolah-olah dilatih terlebih dahulu agar dapat melakukan klasifikasi. Pada tugas ini akan digunakan tiga algoritma yaitu K-Nearest Neighbors (KNN), Decision Tree, dan Random Forest.

   a. K-Nearest Neighbors (KNN)

      K-Nearest Neighbors adalah algoritma yang melakukan prediksi label dengan menghitung jumlah tetangga terdekatnya. Algoritma KNN telah tersedia library yang siap dipakai yaitu dari “sklearn.neighbors” dengan mengimport “KNeighborsClassifier” yang disediakan oleh scikit-learn.

   b. Decision Tree

      Decision Tree adalah algoritma dengan membagi data menjadi himpunan bagian berdasarkan inputan variabelnya. Algoritma ini berbentuk seperti diagram alir yang membantu dalam proses pengambilan keputusan. Algoritma Decision Tree telah tersedia library yang siap dipakai yaitu dari “sklearn.tree” dengan import “DecisionTreeClassifier” yang disediakan scikit-learn

   c. Random Forest

      Random Forest adalah algoritma metode berbasis klasifikasi dan regresi sehingga prosesnya mirip dengan decision tree. Random Forest akan melakukan voting untuk menentukan hasil mayoritas dari decision tree. Maka setelah itu random forest akan melakukan voting untuk menentukan hasil mayoritas dari decision tree. Algoritma Random Forest telah tersedia

library yang siap dipakai yaitu dari "sklearn.ensemble" dengan import "RandomForestClassifier" yang disediakan scikit-learn.

4. Evaluasi

Pada *machine learning* untuk mengetahui seberapa bagus model yang dibangun diperlukan evaluasi model. Teknik untuk mengevaluasi model terbagi menjadi "*Holdout*" dan *"Cross Validation"*. Kedua teknik ini menggunakan data test (data yang belum pernah dilihat oleh model) untuk mengevaluasi model yang dibangun agar menghindari terjadinya *overfitting*. Pada tugas ini model akan dievaluasi dengan *Cross Validation*. *Cross Validation* merupakan metode statistik yang mengevaluasi dan membandingkan algoritma pembelajaran dengan membagi data menjadi dua yaitu data training dan data validation. Teknik *cross validation* yang paling umum adalah *K-Fold Cross Validation* yang juga akan digunakan pada tugas ini. Pada *K-Fold*, dataset dipartisi menjadi K bagian yang sama. K adalah angka yang ditentukan oleh user, biasanya dipilih dari 5 hingga 10 dan pada tugas ini K yang dipilih adalah 7. Satu dari K bagian akan dijadikan data validation dan K-1 bagian akan dijadikan data train. Hal ini diulang sebanyak K kali dengan subsets yang berbeda-beda. Hasil performansi akan dirata-ratakan pada semua K percobaan untuk mendapatkan total keefektifan model.

Metrik evaluasi model diperlukan untuk mengukur kinerja model. Pilihan metrik evaluasi bergantung pada task formulasi masalah yang ingin diselesaikan. Pada kasus ini adalah klasifikasi. Terdapat banyak sekali metrik yang dapat digunakan untuk klasifikasi dan pada tugas ini metrik evaluasi yang digunakan adalah akurasi klasifikasi, confusion matrix, dan f1-score.

Akurasi adalah metrik evaluasi yang umum digunakan untuk masalah klasifikasi. Nilainya didapat dari pembagian antara jumlah prediksi benar terhadap jumlah total prediksi. Confusion matrix merepresentasikan prediksi dan kondisi sebenarnya dari data yang dihasilkan oleh algoritma Machine Learning yang menampilkan rincian mengenai klasifikasi yang benar dan salah untuk setiap kelas. Berdasarkan confusion matrix kita dapat mengetahui nilai precision dan recall. Kemudian F1-Score merupakan metrik yang mempertimbangkan precision dan recall. Precision adalah jumlah hasil positif yang benar dibagi dengan total observasi positif yang diprediksi. Sedangkan recall adalah jumlah hasil positif benar dibagi dengan jumlah semua sampel yang relevan (total positif aktual). Setiap metrik tersebut dapat digunakan dengan memanggil library sklearn.metrics yang disediakan oleh scikit-learn.

5. Eksperimen

Pada tugas ini dilakukan eksperimen sebanyak tiga kali dari dataset hasil preparasi yang sama dengan eksperimen pertama menggunakan algoritma K-Nearest Neighbors, kedua dengan Decision Tree, dan terakhir dengan Random Forest. Pada setiap eksperimen dilakukan tuning hyperparameter untuk mendapatkan hasil evaluasi yang maksimal. Untuk melakukan tuning

hyperparameter akan digunakan Randomized Search CV yang disediakan oleh scikit-learn. Sebelum melatih model, dilakukan pendefinisian kriteria-kriteria yang akan di tuning terlebih dahulu. Kemudian melakukan tuning hyperparameter dari kriteria-kriteria yang telah didefinisikan dan melatih algoritmanya dengan data train. Setelah mendapatkan parameter terbaiknya, melakukan prediksi terhadap data testnya dan dievaluasi dengan melihat confusion matrixnya serta nilai akurasi dan f1-score (average: weighted).

a. Eksperimen Pertama

Eksperimen pertama menggunakan algoritma K-Nearest Neighbors dengan kriteria-kriteria untuk tuning hyperparameter sebagai berikut:

Parameter	**Value**
n_neighbors	[3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25]
weights	['uniform', 'distance']
metric	['euclidean', 'minkowski']

Dengan menggunakan Randomized Search CV, model yang dibentuk mendapatkan skor evaluasi sebesar 0.8449 atau 84,49% dengan performansi metriknya akurasi dan parameter terbaiknya adalah n_neighbors yang bernilai 25, weights bernilai distance, dan metrik perhitungan jaraknya adalah minkowski.

Kemudian model tersebut akan melakukan prediksi terhadap data test dan dievaluasi dengan melihat confusion matrix dan skor performansi akurasi dan f1-scorenya. Berdasarkan hasil confusion matrix True Negative (TN) bernilai 12.983, True Positive (TP) bernilai 1.888, False Negative (FN) bernilai 1.966, dan False Positive (FP) bernilai 692. Untuk performansi metrik akurasi didapatkan skor 0.8483 atau 84,83% dan performansi metrik f1-score dengan average sama dengan weighted mendapatkan skor 0.8367 atau 83,67%.

Skor Akurasi Data Train	**Skor Akurasi Data Test**
0.8449 atau 84,49%	0.8483 atau 84,83%

b. Eksperimen Kedua

Eksperimen kedua menggunakan algoritma Decision Tree dengan kriteria-kriteria untuk tuning hyperparameter sebagai berikut:

Parameter	Value
criterion	['gini', 'entropy']
splitter	['best', 'random']
max_depth	[1, 2, 3, 4, 5, 6, 7, 8, 9]
min_samples_split	[2, 5, 10, 15, 100]

Dengan tuning hyperparameter menggunakan Randomized Search CV, model yang dibentuk dengan classifier Decision Tree mendapatkan skor evaluasi dengan metrik akurasi sebesar 0.8419 atau 84,19% dengan parameter terbaiknya adalah criterion gini, splitter bernilai best, max_depth bernilai 8, dan min_samples_split bernilai 5.

Kemudian model tersebut akan melakukan prediksi terhadap data test dan dievaluasi dengan melihat confusion matrix, skor performansi akurasi, dan f1-scorenya. Berdasarkan hasil confusion matrix True Negative (TN) bernilai 12.939, True Positive (TP) bernilai 1.782, False Negative (FN) bernilai 2.072, dan False Positive (FP) bernilai 736. Untuk performansi metrik akurasi didapatkan skor 0.8398 atau 83,98% dan performansi metrik f1-score dengan average sama dengan weighted mendapatkan skor 0.8267 atau 82,67%

Skor Akurasi Data Train	Skor Akurasi Data Test
0.8419 atau 84,19%	0.8398 atau 83,98%

c. Eksperimen Ketiga

Eksperimen terakhir menggunakan algoritma Random Forest dengan kriteria-kriteria untuk tuning hyperparameter sebagai berikut:

Parameter	Value
n_estimators	[4, 6, 9]
max_features	['log2', 'sqrt', 'auto']
criterion	['entropy', 'gini']
max_depth	[2, 3, 5, 10]

min_samples_split	[2, 3, 5]
min_samples_leaf	[1, 5, 8]

Dengan tuning hyperparameter menggunakan Randomized Search CV, model yang dibentuk dengan classifier Random Forest mendapatkan skor evaluasi dengan metrik akurasi sebesar 0.8468 atau 84,68% dengan parameter terbaiknya adalah n_estimators benilai 9, min_samples_split bernilai 3, min_samples_leaf bernilai 5, max_features bernilai log2, max_depth bernilai 10, dan criterionnya gini.

Kemudian model tersebut akan melakukan prediksi terhadap data test dan dievaluasi dengan melihat confusion matrix, skor performansi akurasi, dan f1-scorenya. Berdasarkan hasil confusion matrix True Negative (TN) bernilai 13.071, True Positive (TP) bernilai 1.825, False Negative (FN) bernilai 2.029, dan False Positive (FP) bernilai 604. Untuk performansi metrik akurasi didapatkan skor 0.8497 atau 84,97% dan performansi metrik f1-score dengan average sama dengan weighted mendapatkan skor 0.8364 atau 83,64%

Skor Akurasi Data Train	**Skor Akurasi Data Test**
0.8468 atau 84,68%	0.8497 atau 84,97%

6. Kesimpulan

Berdasarkan hasil eksperimen yang dilakukan dapat ditarik kesimpulan bahwa untuk dataset salju dengan fitur yang dipilih rata-rata memiliki skor evaluasi sekitar 80an persen. Skor evaluasi tertinggi berasal dari classifier Random Forest dengan nilai skor model dengan data train sebesar 84,68% dan skor prediksi dengan data test sebesar 84,97%. Ini berarti model dengan classifier Random Forest tersebut merupakan model yang baik dalam mengklasifikasikan data yang diberikan.