

“Implementasi PyCaret untuk Meningkatkan Akurasi Prediksi Penjualan di PT. Sinar Megartha Perkasa”

“Implementing PyCaret to Enhance Sales Prediction Accuracy at PT. Sinar Megartha Perkasa”

Darmawan Susilo¹⁾, Terttiaavini²⁾, Tri Kumalinda³⁾, Lestari Sagita⁴⁾

¹Magister Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Indo Global Mandiri
Jl. Jendral Sudirman No.629 Km.4, Palembang, Sumatera Selatan

Telp : (0711) 3227-05(-06)

E-mail : darma.awan.susilo@gmail.com¹⁾, avini.saputra@uigm.ac.id²⁾, tri.kumalinda28@gmail.com³⁾
lestarisagita@gmail.com⁴⁾

Abstract

PT. Sinar Megartha Perkasa faces challenges in enhancing the accuracy of sales predictions, which is crucial for supporting strategic planning and effective decision-making. This study aims to implement PyCaret, an open-source machine learning library based on Python that utilizes a low-code approach, to develop a more accurate sales prediction model. The methodology includes the collection of historical sales data, data preprocessing, and the application of regression algorithms through PyCaret. To evaluate the model's performance, metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are employed. The results indicate that the use of PyCaret significantly improves sales prediction accuracy compared to traditional methods, with a substantial reduction in both MAE and RMSE values. This research provides a practical solution for developing an efficient sales prediction model that can be integrated into the management system of PT. Sinar Megartha Perkasa. With more accurate predictions, the company can make data-driven strategic decisions to maintain its competitive advantage in the market.

Keywords: PyCaret, machine learning, accuracy, Python, sales prediction

Abstrak

PT. Sinar Megartha Perkasa menghadapi tantangan dalam meningkatkan akurasi prediksi penjualan, yang sangat penting untuk mendukung perencanaan strategis dan pengambilan keputusan yang efektif. Penelitian ini bertujuan untuk menerapkan *PyCaret*, sebuah pustaka *machine learning open-source* berbasis *Python* yang dirancang dengan pendekatan *low-code*, guna mengembangkan model prediksi penjualan yang lebih tepat. Metodologi yang digunakan mencakup pengumpulan data penjualan historis, pra-pemrosesan data, dan penerapan algoritma regresi melalui *PyCaret*. Untuk mengevaluasi kinerja model, digunakan metrik seperti *Mean Absolute Error* (MAE) dan *Root Mean Squared Error* (RMSE). Hasil penelitian menunjukkan bahwa penggunaan *PyCaret* secara signifikan meningkatkan akurasi prediksi penjualan jika dibandingkan dengan metode tradisional, dengan penurunan yang signifikan pada nilai MAE dan RMSE. Penelitian ini memberikan solusi praktis untuk pengembangan model prediksi penjualan yang efisien, yang dapat diintegrasikan ke dalam sistem manajemen PT. Sinar Megartha Perkasa. Dengan adanya prediksi yang lebih akurat, perusahaan dapat membuat keputusan strategis berbasis data untuk mempertahankan keunggulan kompetitif di pasar.

Kata kunci: PyCaret, machine learning, akurasi, Python, Prediksi penjualan

1. Pendahuluan

Di era digital saat ini, data penjualan merupakan aset berharga bagi perusahaan, mencerminkan kinerja bisnis serta potensi untuk mengungkap wawasan strategis. Namun, PT. Sinar Megartha Perkasa menghadapi kesulitan dalam mengolah data penjualan yang besar dan kompleks dengan metode konvensional. Analisis manual yang memakan waktu tidak hanya rentan terhadap kesalahan, tetapi juga kurang optimal dalam menemukan pola mendalam. Teknologi *machine learning* menawarkan solusi atas tantangan ini [1].

PyCaret, sebuah pustaka *machine learning* otomatis berbasis *Python*, memungkinkan perusahaan mengotomatiskan proses *machine learning*, mulai dari

eksplorasi data hingga pembuatan model prediktif, secara efisien dan mudah digunakan. Implementasi *PyCaret* dapat merevolusi cara perusahaan memanfaatkan data mereka, terutama dalam meningkatkan akurasi prediksi penjualan.

Penelitian ini bertujuan untuk mengimplementasikan *PyCaret* dalam menganalisis data penjualan di PT. Sinar Megartha Perkasa. Dengan memanfaatkan fitur seperti jumlah penjualan (*salesqty*), harga, nilai diskon (*valuedisc*), serta mengkategorikan jumlah penjualan bersih (*netamount*) menjadi Rendah, Sedang, Tinggi, dan Sangat Tinggi menggunakan metode diskretisasi berbasis kuantil, model prediksi yang dihasilkan diharapkan memberikan akurasi lebih tinggi. Model ini dirancang untuk memprediksi jumlah penjualan, rentang

harga, dan nilai diskon yang dapat mempertahankan profitabilitas di masa depan.

Metode ini selaras dengan penelitian sebelumnya yang membuktikan efektivitas PyCaret dalam peramalan pendapatan penjualan, seperti yang dilakukan di KC Sumedang oleh Febriansyah, F. (2024) [2]. Dengan hasil yang terbukti akurat dan efisien, penelitian ini bertujuan memberikan kontribusi signifikan dalam mentransformasi analisis data penjualan menjadi aset strategis bagi PT. Sinar Megartha Perkasa, mendukung pengambilan keputusan yang lebih baik di masa mendatang [3].

Dengan hasil yang terbukti akurat dan efisien, penelitian ini diharapkan memberikan kontribusi nyata dalam transformasi analisis data penjualan menjadi aset strategis bagi PT. Sinar Megartha Perkasa, mendukung pengambilan keputusan yang lebih baik di masa depan.

2. Tinjauan Pustaka

Analisis data penjualan adalah aspek penting dalam pengambilan keputusan strategis bisnis. Dengan meningkatnya *volume* data yang dihasilkan perusahaan, metode tradisional dalam analisis data sering kali kurang memadai. Oleh karena itu, penerapan teknik *machine learning* menjadi semakin relevan. *Machine learning* memungkinkan perusahaan untuk mengekstrak wawasan dari data besar dan kompleks, menghasilkan prediksi yang lebih akurat terkait perilaku konsumen dan tren penjualan [4].

Dalam penelitian Abidin dan Nuryana (2023), dilakukan perbandingan metode klasifikasi *data mining* untuk mengukur tingkat kepuasan mahasiswa terhadap Sistem Informasi Penilaian Non-Akademik UNESA (SIPENA) [5]. Penelitian ini menggunakan *PyCaret* untuk membandingkan performa 16 algoritma klasifikasi.

2.1. *Machine Learning* dalam Analisis Data Penjualan

Machine learning telah menjadi alat yang efektif dalam menganalisis data penjualan. Menurut Rahmawati dan Setiawan (2022), pemodelan menggunakan algoritma *machine learning* dapat memprediksi penjualan di masa depan dengan evaluasi kinerja menggunakan RMSE, MEA, dan R^2 [6].

2.2. *PyCaret* sebagai Alat untuk Analisis Data

PyCaret adalah pustaka *machine learning* open-source berbasis *Python* yang dirancang untuk menyederhanakan alur kerja *machine learning*. Dengan *PyCaret*, proses analisis data menjadi lebih efisien dan mudah diakses, memungkinkan pengguna untuk fokus pada analisis daripada penulisan kode [7]

Dalam penelitian Abidin dan Nuryana (2023), *PyCaret* digunakan untuk membandingkan kinerja beberapa metode klasifikasi data mining. Penelitian tersebut bertujuan mengukur tingkat kepuasan mahasiswa

terhadap SIPENA menggunakan data kuesioner yang divalidasi dengan SPSS. *PyCaret* menyederhanakan proses analisis dengan mendukung otomatisasi dalam membangun dan membandingkan model *machine learning*. [8]

2.3. Penerapan Analisis Prediktif dalam Penjualan

Pengoptimalan proses bisnis dapat dicapai dengan penerapan data *science*. Melalui analisis mendalam terhadap data yang terkait dengan alur kerja, sumber daya, dan waktu operasional, perusahaan dapat meningkatkan efisiensi sekaligus efektivitas operasional mereka. Pendekatan ini menjadi relevan dalam membantu perusahaan teknologi untuk beradaptasi terhadap tantangan bisnis modern [9].

Analisis prediktif menggunakan *machine learning* memungkinkan perusahaan untuk meramalkan tren penjualan dan perilaku konsumen. Elvina (2021) mengembangkan model prediksi penjualan multi-item time series berbasis *machine learning*, yang dapat meningkatkan akurasi prediksi penjualan [10].

2.4. Tantangan dalam Analisis Data Penjualan

Meskipun *machine learning* menawarkan banyak manfaat, terdapat tantangan dalam implementasinya, seperti kualitas data, pemilihan algoritma yang tepat, serta interpretasi hasil. Suryadi dan Kurniawan (2022) membahas penerapan algoritma *random forest* untuk prediksi penjualan dan manajemen inventaris, menyoroti pentingnya pemilihan algoritma yang sesuai dengan karakteristik data [11].

3. Metodologi Penelitian

Metode penelitian terdiri atas beberapa tahap, yaitu:

3.1 Pengumpulan Data

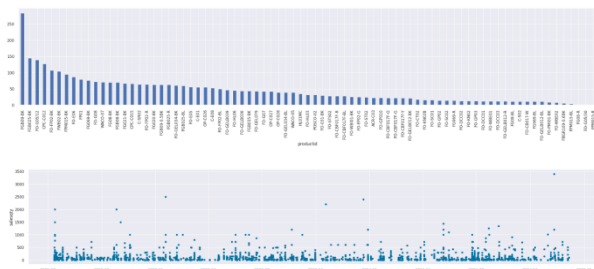
Pengumpulan data merupakan langkah awal yang krusial dalam analisis data. Dalam konteks ini, data penjualan bulanan dari tahun 2022 hingga 2024 dikumpulkan untuk memperoleh gambaran komprehensif mengenai tren penjualan selama periode tersebut. Data ini mencakup berbagai variabel relevan, seperti jumlah penjualan (*salesqty*), harga per unit, jumlah bruto, dan jumlah bersih terkait penjualan, nilai diskon, serta data pendukung tambahan seperti tanggal transaksi (*transdate*), kode unit, dan departemen.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3477 entries, 0 to 3476
Data columns (total 25 columns):
#   Column              Non-Null Count  Dtype
---  ---
0   transid             3477 non-null   int64
1   transdate           3477 non-null   object
2   salesid             3477 non-null   object
3   salesidref          3477 non-null   object
4   productid           3477 non-null   object
5   snproduct           0 non-null      float64
6   expireddate        0 non-null      float64
7   salesqty            3477 non-null   int64
8   returnqty           3477 non-null   int64
9   unit                3477 non-null   object
10  price               3477 non-null   int64
11  grossamount         3477 non-null   int64
12  taxid              3477 non-null   int64
13  salestax           3477 non-null   int64
14  returntax          3477 non-null   int64
15  percentdisc        3475 non-null   object
16  valuedisc           3477 non-null   int64
17  netamount          3477 non-null   int64
18  cogs               3477 non-null   int64
19  memo               3477 non-null   object
20  departement        3477 non-null   int64
21  servicedoerid      3477 non-null   int64
22  usercreate         3477 non-null   object
23  useredit           3477 non-null   object
24  updatetimestamp    3477 non-null   object
dtypes: float64(2), int64(13), object(10)
memory usage: 679.2+ KB
```

Gambar 1. Summary Data

3.2 Eksplorasi Data

Setelah data dikumpulkan, langkah berikutnya adalah eksplorasi data. Visualisasi data digunakan untuk mengidentifikasi pola, tren, dan anomali dalam data. Teknik visualisasi seperti grafik garis, histogram, dan diagram pencar membantu memahami distribusi data dan mengidentifikasi *outlier*. Eksplorasi data juga melibatkan analisis statistik deskriptif untuk mendapatkan wawasan awal tentang karakteristik data.



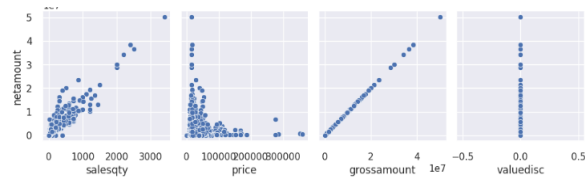
Gambar 2. Grafik jumlah “salesqty” dan Plot berdasarkan waktu untuk melihat *outlier*

Gambar 2. Grafik Plot untuk melihat *outlier*, didapatkan analisis singkat sebagai berikut :

- Sebagian besar jumlah penjualan (*salesqty*) berkisar antara 0 hingga 500 unit, dengan beberapa outlier (pencilan) yang menunjukkan jumlah penjualan yang jauh lebih tinggi, bahkan melebihi 3000 unit.
- Pada rentang waktu tertentu (misalnya akhir 2022, awal 2023, dan akhir 2024), terdapat lonjakan signifikan dalam jumlah penjualan.
- Pola penjualan secara umum cenderung berfluktuasi, dengan periode waktu yang tenang dan beberapa titik lonjakan penjualan.

3.2 Pemilihan Data dan Penentuan Model

Setelah eksplorasi data, data yang dianalisis menentukan fitur ("features") dan target. Dalam hal ini, fitur ditentukan sebagai "salesqty", "price", dan "valuedisc", sedangkan target adalah "netamount".



Gambar 3. Pair plot atau scatterplot matrix untuk seluruh pasangan kolom dalam dataset yang akan digunakan

Gambar 3. Pair plot atau scatterplot matrix untuk seluruh pasangan kolom dalam dataset yang akan digunakan, ini adalah plot scatter matrix yang menunjukkan hubungan antara variabel-variabel berikut:

- salesqty* (Jumlah penjualan) vs *netamount* (Jumlah bersih), terdapat korelasi positif di mana peningkatan jumlah penjualan cenderung meningkatkan jumlah bersih.
- price* (Harga) vs *netamount* (Jumlah bersih), tampak sebagian besar data terkonsentrasi di harga rendah, tetapi ada beberapa outlier pada harga tinggi yang mempengaruhi *netamount*.
- valuedisc* (Diskon nilai) vs *netamount* (Jumlah bersih), variabel *valuedisc* tampaknya terkonsentrasi di sekitar nol, dengan beberapa titik menyebar tetapi tidak menunjukkan pola yang signifikan terhadap *netamount*.

Scatter matrix ini membantu menganalisis pola hubungan antarvariabel, mendeteksi outlier, dan memahami struktur data.

Tabel 1. Tabel Deskripsi Data

Column	Count	Unique	Top	Freq	Mean	Min	25%	50%	75%	Max	Std Dev
salesqty	3477				75.881795	1	6	24	72	3400	174.247283
unit	3477	27	LSN144	746							
price	3477				28696.19758	0	15680	21945	30200	360000	25379.52651
grossamount	3477				1454515.111	0	150000	498960	1558200	50143200	2787079.399
valuedisc	3477				0	0	0	0	0	0	0
netamount	3477				1454515.111	0	150000	498960	1558200	50143200	2787079.399

Berdasarkan hasil analisis data dari fungsi *describe*, ditemukan bahwa variabel *salesqty* menunjukkan adanya variasi yang cukup besar dalam jumlah penjualan. Distribusi data melibatkan nilai ekstrem, dengan nilai maksimum mencapai 3400 yang jauh lebih tinggi dari nilai rata-rata. Hal ini menunjukkan adanya perbedaan signifikan dalam jumlah penjualan antar data. Selanjutnya, variabel *price* juga memiliki rentang nilai yang cukup luas, dengan beberapa nilai outlier pada harga maksimum, yang menunjukkan adanya transaksi dengan harga jauh lebih tinggi dibandingkan nilai rata-rata. Untuk variabel *netamount*, data memperlihatkan total nilai sebelum diskon atau pengurangan, dengan rentang yang sangat besar, mencerminkan variasi signifikan dalam transaksi. Adapun variabel *valuedisc*, nilainya konstan (0) di seluruh data, sehingga tidak memberikan pengaruh terhadap analisis saat ini.

Namun, mempertahankan variabel *valuedisc* dalam model tetap menjadi langkah yang dipertimbangkan, terutama untuk tujuan berikut:

- a. Konsistensi Format Data, dengan menyertakan *valuedisc* memungkinkan format *dataset* tetap seragam, yang penting untuk integrasi data atau penggunaan model secara berkelanjutan. Hal ini juga memastikan bahwa *pipeline* data tidak perlu dimodifikasi ketika nilai diskon mulai bervariasi di masa depan.
- b. Antisipasi Perubahan di Masa Depan, walaupun nilai *valuedisc* saat ini nol ada kemungkinan bahwa diskon akan diterapkan di masa depan. Dengan menyertakan variabel ini sejak awal, model akan siap menerima fitur tersebut tanpa perlu dilakukan pelatihan ulang. Pendekatan ini sangat relevan dalam skenario bisnis dinamis, di mana kebijakan diskon dapat berubah sewaktu-waktu

Namun demikian, menyertakan variabel tidak informatif seperti *valuedisc* juga memiliki potensi risiko, seperti *overfitting* pada Model Kompleks Beberapa algoritma sensitif terhadap dimensi data (seperti *regresi linear* atau *neural networks*), sehingga variabel konstan dapat memengaruhi performa model, meskipun dampaknya mungkin kecil. Sehingga pada kasus ini akan diputuskan menggunakan metode klasifikasi.

Dengan mempertimbangkan pentingnya fleksibilitas model di masa depan, menyertakan *valuedisc* merupakan keputusan strategis. Diskon adalah salah satu elemen penting dalam memenuhi kebutuhan pasar. Dengan memasukkan fitur ini, model dirancang untuk jangka panjang, memungkinkan penerapan diskon di kemudian hari tanpa perlu perubahan besar. Pendekatan ini juga selaras dengan strategi bisnis yang berorientasi pada adaptasi dan keberlanjutan.

3.2 Model dengan Pycaret

Data yang telah dipilih kemudian dinormalisasi atau distandardisasi (jika diperlukan), dan model dibangun menggunakan *PyCaret*. Dalam penelitian ini, *Agglomerative Clustering* dipilih karena kemampuannya dalam menghasilkan kluster yang lebih jelas dan terstruktur, sebagaimana diusulkan oleh Terttiaavini (2024) [12]

Dalam langkah ini, modul yang digunakan adalah klasifikasi dengan metode transformasi *netamount* menjadi kategori menggunakan *pycaret.classification*.

```
from pycaret.classification import *

# Convert 'netamount' ke kategorikal
train_data['netamount_category'] = pd.cut(
    train_data['netamount'], bins=4, labels=['Low', 'Medium', 'High', 'Very High']
) # q merupakan jumlah bilangan (0,1,2,3)
```

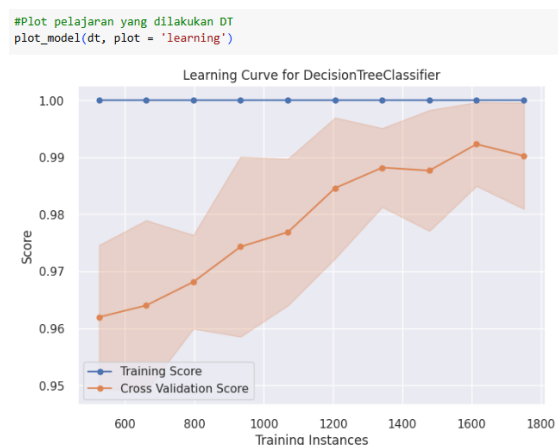
Gambar 4. Merubah “netamount” menjadi kategorikal menggunakan “pycaret.classification”

	Description	Value
0	Session id	123
1	Target	netamount_category
2	Target type	Multiclass
3	Target mapping	High: 0, Low: 1, Medium: 2, Very High: 3
4	Original data shape	(2781, 5)
5	Transformed data shape	(2781, 4)
6	Transformed train set shape	(1946, 4)
7	Transformed test set shape	(835, 4)
8	Ignore features	1
9	Numeric features	3
10	Preprocess	True
11	Imputation type	simple
12	Numeric imputation	mean
13	Categorical imputation	mode
14	Fold Generator	StratifiedKFold
15	Fold Number	10
16	CPU Jobs	-1
17	Use GPU	False
18	Log Experiment	False
19	Experiment Name	clf-default-name
20	USI	5255

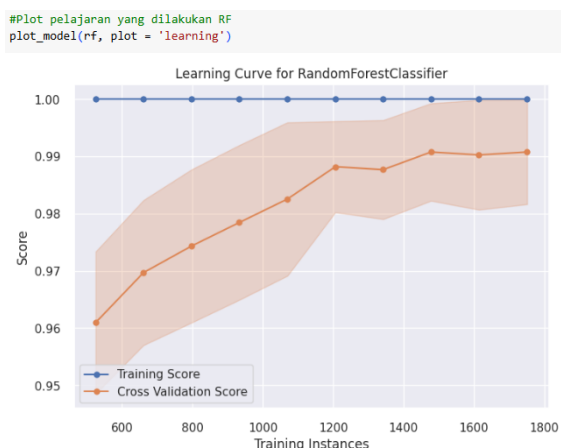
Gambar 5. Mapping Model
Tabel 2. Tabel hasil perbandingan Model

No	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec.)
1	Extra Trees Classifier	0.9918	0.9999	0.9918	0.9921	0.9918	0.989	0.9891	0.338
2	Light Gradient Boosting Machine	0.9913	0.9998	0.9913	0.9914	0.9913	0.9884	0.9884	1.065
3	Random Forest Classifier	0.9908	0.9996	0.9908	0.9911	0.9908	0.9877	0.9877	0.307
4	Decision Tree Classifier	0.9902	0.9935	0.9902	0.9905	0.9902	0.987	0.987	0.075
5	Extreme Gradient Boosting	0.9866	0.9996	0.9866	0.9866	0.9867	0.9823	0.9823	0.153
6	Gradient Boosting Classifier	0.9815	0	0.9815	0.9813	0.9815	0.9755	0.9755	0.987
7	Logistic Regression	0.7852	0	0.7852	0.7848	0.7848	0.5746	0.5746	0.116
8	Naive Bayes	0.7194	0.8998	0.7194	0.7493	0.7341	0.5154	0.5154	0.069
9	K Neighbors Classifier	0.7107	0.9947	0.7107	0.7301	0.7132	0.6141	0.6141	0.161
10	Ada Boost Classifier	0.6367	0	0.6367	0.7779	0.6063	0.5153	0.5153	0.166
11	Linear Discriminant Analysis	0.5092	0	0.5092	0.5863	0.5041	0.3537	0.3537	0.084
12	Ridge Classifier	0.4665	0	0.4665	0.4874	0.3975	0.2886	0.2886	0.333
13	SVM - Linear Kernel	0.3258	0	0.3258	0.3633	0.3422	0.1691	0.1691	1.144
14	Dummy Classifier	0.2518	0	0.2518	0.0643	0.1139	0	0	0.03
15	Quadratic Discriminant Analysis	0.2497	0	0.2497	0.0624	0.0998	0	0	0.044

Hasil dari komparasi model di atas menunjukkan beberapa model dengan nilai performa tinggi (mendekati 1). Namun, untuk mencegah terjadinya *overfitting*, dipilih model dengan performa yang stabil dan tidak terlalu ekstrem. Dalam kasus ini, model *Random Forest* dan *Decision Tree* dipilih sebagai model terbaik untuk klasifikasi. Seperti dikutip dari Terttiaavini (2024), *Random Forest* menunjukkan performa unggul dalam pengolahan data dan mampu mengurangi risiko *overfitting*, menjadikannya pilihan ideal untuk prediksi yang akurat.



Gambar 6. Grafik Plot learning Decision Tree



Gambar 7. Grafik Plot learning Random Forest

3.3 Evaluasi Model

Random Forest dan *Decision Tree* adalah algoritma berbasis pohon yang unggul dalam menangani data dengan fitur kategorikal, kontinual, atau kombinasi keduanya. *Dataset* penjualan biasanya memiliki fitur yang tidak linier, seperti diskon, harga, dan jumlah penjualan. Algoritma berbasis pohon mampu menangkap pola *non-linier* ini secara efektif dan *Random Forest* bekerja dengan membangun banyak pohon keputusan dan melakukan agregasi (*ensemble learning*), sehingga lebih tahan terhadap *overfitting* dibandingkan *Decision Tree* tunggal.

Dataset penelitian ini (penjualan PT. Sinar Megartha Perkasa) memiliki ukuran data sedang, yang cocok untuk *Random Forest* dan *Decision Tree* karena algoritma ini tidak terlalu memerlukan data yang sangat besar seperti model berbasis *deep learning*. Setelah melihat grafik pair plot, keputusan menggunakan kedua model yang tidak terlalu terpengaruh oleh outlier dan pembagian data dalam pohon keputusan lebih fokus pada aturan pembagian yang optimal.

```
## 8. MENGEVALUASI HASIL MODEL
# Fitur dan Target
X = df[['salesqty', 'price', 'valuedisc']]
y = pd.qcut(df['netamount'], q=4, labels=["Low", "Medium", "High", "Very High"])
# Discretize target menjadi kategori

# Split Data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Model 1: Decision Tree Classifier
# Importing the library containing DecisionTreeClassifier
from sklearn.tree import DecisionTreeClassifier
dt_model = DecisionTreeClassifier(random_state=42)
dt_model.fit(X_train, y_train)
dt_y_pred = dt_model.predict(X_test)

# Evaluasi Decision Tree
print("=== Evaluasi Decision Tree Classifier ===")
print(classification_report(y_test, dt_y_pred))
print(f"Akurasi Decision Tree: {accuracy_score(y_test, dt_y_pred):.4f}")
```

Gambar 8. Model Klasifikasi Decision Tree

Tabel 3. Hasil Evaluasi Decision Tree Classifier

Class	Precision	Recall	F1-Score	Support
High	0.98	0.99	0.99	180
Low	1	1	1	174
Medium	0.99	0.99	0.99	162
Very High	1	0.98	0.99	180
Accuracy			0.99	696
Macro Avg	0.99	0.99	0.99	696
Weighted Avg	0.99	0.99	0.99	696

Tabel 3. Hasil Evaluasi *Decision Tree Classifier*, menunjukkan bahwa Model *Decision Tree* yang dievaluasi memiliki performa yang sangat baik dengan akurasi mendekati 99% dan metrik evaluasi lainnya yang konsisten tinggi di semua kelas. Model ini cocok untuk diterapkan pada data serupa dengan distribusi kelas yang terprediksi dengan baik.

```
# Model 2: Random Forest Classifier
# Import RandomForestClassifier
from sklearn.ensemble import RandomForestClassifier

rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
rf_y_pred = rf_model.predict(X_test)

# Evaluasi Random Forest
print("\n=== Evaluasi Random Forest Classifier ===")
print(classification_report(y_test, rf_y_pred))
print(f"Akurasi Random Forest: {accuracy_score(y_test, rf_y_pred):.4f}")
```

Gambar 8. Model Klasifikasi Random Forest

Tabel 3. Hasil Evaluasi Random Forest Classifier

Class	Precision	Recall	F1-Score	Support
High	0.98	0.99	0.99	180
Low	1	1	1	174
Medium	0.99	0.99	0.99	162
Very High	1	0.98	0.99	180
Accuracy			0.99	696
Macro Avg	0.99	0.99	0.99	696
Weighted Avg	0.99	0.99	0.99	696

Dari Tabel 3. Hasil Evaluasi *Random Forest Classifier*, dapat dilihat bahwa Model *Random Forest Classifier* menunjukkan performa yang sangat baik pada *dataset*, dengan akurasi keseluruhan 99.28% dan skor metrik lain yang mendekati sempurna di semua kelas.

Menggunakan *confusion_matrix*, dapat dilihat hasil dari model yang dijalankan juga berguna untuk memahami kinerja klasifikasi pada level yang lebih mendetail, bukan hanya akurasi secara keseluruhan. *Confusion Matrix* adalah tabel yang menggambarkan kinerja model klasifikasi, dengan membandingkan label aktual dan label prediksi. Setiap elemen pada matriks memiliki makna:

- True Positive* (TP): Prediksi benar untuk kelas target tertentu.
- False Positive* (FP): Prediksi salah untuk kelas target tertentu (terklasifikasi salah sebagai kelas ini).

- c. *True Negative* (TN): Prediksi benar bukan kelas target.
- d. *False Negative* (FN): Prediksi salah karena gagal mendeteksi kelas target.

Tabel 4. *Confusion Matrix* (Hasil dari kinerja klasifikasi)

Actual/Predicted	Class 0	Class 1	Class 2	Class 3
Class 0	178	0	2	3
Class 1	0	174	0	0
Class 2	2	0	160	0
Class 3	0	0	0	177

Dari gambar 7. Hasil dari kinerja klasifikasi, analisis untuk mengevaluasi kinerja model dalam mengklasifikasikan data ke dalam empat kategori: *Low*, *Medium*, *High* dan *Very High* dimana simpulan sebagai berikut :

- a. Baris 0: Kelas 0 (aktual)
 - 178 prediksi benar sebagai '0' (*True Positive*).
 - 0 prediksi salah sebagai '1'.
 - 2 prediksi salah sebagai '2'.
 - 0 prediksi salah sebagai '3'.
- b. Baris 1: Kelas 1 (aktual)
 - 174 prediksi benar sebagai '1' (*True Positive*).
 - 0 prediksi salah sebagai '0'.
 - 0 prediksi salah sebagai '2'.
 - 0 prediksi salah sebagai '3'.
- c. Baris 2: Kelas 2 (aktual)
 - 160 prediksi benar sebagai '2' (*True Positive*).
 - 2 prediksi salah sebagai '0'.
 - 0 prediksi salah sebagai '1'.
 - 0 prediksi salah sebagai '3'.
- d. Baris 3: Kelas 3 (aktual)
 - 177 prediksi benar sebagai '3' (*True Positive*).
 - 3 prediksi salah sebagai '0'.
 - 0 prediksi salah sebagai '1'.
 - 0 prediksi salah sebagai '2'.

Berdasarkan analisa diatas, terlihat bahwa model memiliki performansi yang sangat baik dan dengan hanya sedikit kesalahan prediksi. Kesalahan utamanya berada di kelas 0 dan 3, di mana sebagian kecil sampel diprediksi menjadi kelas yang salah.

```

## MELAKUKAN UJI MODEL
# Hasil Uji Coba
# Misalnya kita ingin menggunakan model Random Forest (rf_model)
# Data yang ingin diprediksi
Data = pd.DataFrame({
    'salesqty': [100, 100, 300], # Contoh jumlah penjualan
    'price': [500, 5000, 10000], # Contoh harga produk
    'valuedisc': [5, 5, 15] # Contoh diskon nilai produk
})

# Prediksi dengan model Random Forest
rf_prediction = rf_model.predict(Data)

# Menampilkan hasil prediksi
print("Hasil Prediksi dengan Random Forest:")
print(rf_prediction)

# Prediksi dengan model Decision Tree (jika Anda ingin menggunakan dt_model)
dt_prediction = dt_model.predict(Data)
print("\nHasil Prediksi dengan Decision Tree:")
print(dt_prediction)
    
```

```

Hasil Prediksi dengan Random Forest:
['Low' 'High' 'Very High']

Hasil Prediksi dengan Decision Tree:
['Low' 'High' 'Very High']
    
```

Gambar 9. Hasil uji model

Dalam gambar 8. Hasil uji model, dapat dilihat bahwa *salesqty*, *price* dan *valuedisc* secara praktis dapat menjadi model penentuan *netamount*, dengan begitu perusahaan dapat memprediksi berapa jumlah yang harus dijual dengan harga berapa dan jumlah diskon yang tidak merugikan atau dengan hasil maksimal.

4. Hasil dan Pembahasan

Secara keseluruhan, penerapan *machine learning* dalam analisis data penjualan menawarkan peluang besar untuk meningkatkan efisiensi dan akurasi analisis. Dengan menggunakan alat seperti *PyCaret*, perusahaan dapat mengotomatisasi proses analisis dan mendapatkan wawasan yang lebih dalam dari data mereka. Namun, penting untuk mengatasi tantangan terkait kualitas data agar hasil analisis dapat diandalkan dan mendukung pengambilan keputusan yang strategis.

4.1 Eksplorasi Data

Hasil eksplorasi menunjukkan pola musiman dalam data penjualan dengan puncak penjualan terjadi pada kuartal akhir setiap tahun.

Dengan model yang mampu memprediksi nilai bersih penjualan (*netamount*) secara akurat, PT. Sinar Megartha Perkasa dapat menentukan kombinasi optimal antara harga, jumlah penjualan, dan diskon.

4.2 Evaluasi Model

Kesalahan prediksi dari hasil umumnya sangat rendah dan model klasifikasi mempermudah penyampaian informasi dengan mengelompokkan data ke dalam kelas-kelas tertentu, seperti "low", "medium", "high", dan "very high". Dengan pembagian kelas tersebut, diharapkan pemegang keputusan dapat dengan lebih mudah memahami informasi yang disajikan, sehingga mengurangi potensi kesalahan interpretasi dan memfasilitasi pengambilan keputusan yang lebih tepat.

4.3 Implementasi

Hasil analisis diterapkan dalam *dashboard* interaktif untuk memantau performa penjualan secara *real-time*.

5. Kesimpulan

Bagian terdiri atas simpulan dan saran atas penelitian hasil penelitian.

5.1 Simpulan

Penggunaan *PyCaret* terbukti meningkatkan efisiensi dan akurasi analisis data penjualan. Implementasi ini mendukung pengambilan keputusan yang lebih cepat dan berbasis data.

Dari simpulan tersebut wawasan strategis dari Hasil, dapat disampaikan sebagai berikut :

- a. Keunggulan kompetitif, dengan memanfaatkan model prediksi yang akurat, PT. Sinar Megartha Perkasa dapat lebih responsif terhadap perubahan tren pasar, memberikan keunggulan kompetitif dibandingkan pesaing.
- b. Perencanaan jangka panjang, data prediksi yang dapat diandalkan memungkinkan perusahaan membuat keputusan strategis, seperti investasi dalam produk baru atau ekspansi pasar.
- c. Identifikasi peluang pasar, dimana analisis data dapat mengungkap pola penjualan musiman yang sebelumnya tidak terlihat, membantu perusahaan memanfaatkan peluang peningkatan pendapatan pada periode tertentu.

Melalui penelitian ini, kami berharap dapat menginspirasi perusahaan lain untuk mengadopsi pendekatan berbasis teknologi seperti *PyCaret*. Automasi, dalam konteks ini, tidak hanya tentang efisiensi proses tetapi juga sebagai katalisator inovasi yang dapat mengubah tantangan menjadi peluang.

5.2 Saran

Penelitian lanjutan dapat menambahkan variabel eksternal seperti data ekonomi atau demografi untuk meningkatkan prediksi dan gunakan *Google Colab* atau *Jupyter Notebook* untuk pengembangan model berbasis *PyCaret* dengan sumber daya yang efisien dan hemat biaya.

Penelitian selanjutnya dapat memanfaatkan *PyCaret* untuk membangun model *time series* yang memprediksi tren musiman dalam penjualan. Misalnya, prediksi peningkatan penjualan pada kuartal terakhir tahun ini

Ucapan Terima Kasih

Penulis berterima kasih kepada PT. Sinar Megartha Perkasa dan Universitas Indo Global Mandiri atas dukungan penelitian ini.

Daftar Rujukan

- [1] DQLab, "Keunggulan Hingga Kelemahan Machine Learning yang Wajib Diketahui," 23 12 2024. [Online]. Available: <https://dqlab.id/keunggulan-hingga-kelemahan-machine-learning-yang-wajib-diketahui>.
- [2] F. Febriansyah, "Analisis dan Peramalan Pendapatan Penjualan Korporat di KC Sumedang Menggunakan PyCaret untuk Algoritma Time Series," 2024.
- [3] M. I. Baidowi, "ANALISIS RAMALAN PENJUALAN MENGGUNAKAN METODE TIME SERIES DALAM," *MANAGEMENT SCIENCE*, vol. 1, no. 1, p. 33, 2020.
- [4] N. C. a. J. Ferreira, "Salespeople performance evaluation with predictive analytics in B2B," *Appl.*, vol. 10, no. 11, p. 4036, 2020.
- [5] A. R. Abidin and I. K. D. Nuryana, "Perbandingan Metode Klasifikasi Data Mining untuk Mengukur Tingkat Kepuasan Mahasiswa terhadap Sistem Informasi Penilaian Nonakademik UNESA (SIPENA)," *Journal of Emerging Information Systems and Business Intelligence (JEISBI)*, vol. 4, no. 4, p. 129–138, 2023.
- [6] F. Rahmawati and A. Setiawan, "Analisis dan Prediksi Data Penjualan Menggunakan Machine Learning," *Data Science Indonesia Journal*, vol. 4, no. 1, p. 45–56, 2022.
- [7] M. Ali, *PyCaret*, 2024. [Online]. Available: <https://pyparet.gitbook.io/docs>.
- [8] A. R. Abidin, I. Kadek dan D. Nuryana, "Perbandingan Metode Klasifikasi Data Mining Untuk Mengukur Tingkat Kepuasan Mahasiswa Terhadap Sistem Informasi Penilaian Nonakademik UNESA (SIPENA)," 2023.
- [9] Terttiaavini et al., *FUNDAMENTAL ALGORITMA*, Sada Kurnia Pustaka, 2023.
- [10] S. Elvina, "Model Prediksi Penjualan Multi-Item Time Series Berbasis Machine Learning," *Universitas Islam Indonesia*, 2021.
- [11] D. Suryadi and R. Kurniawan, "Penerapan Algoritma Random Forest untuk Prediksi Penjualan dan Pengelolaan Persediaan," *Jurnal Resolusi*, vol. 5, p. 2149–2156, 2022.
- [12] Terttiaavini, "Analisis Perbandingan Algoritma Machine Learning untuk Prediksi Stunting pada Anak," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 1, p. 257–265, 2024.