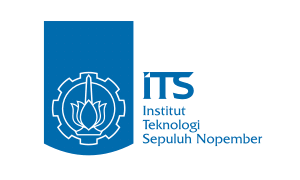
**TUGAS BESAR DATA MINING**

**Analisis Data Penjualan pada Bisnis *E-Commerce* dengan Menggunakan Pendekatan *Market Basket Analysis***



**Disusun Oleh:**

**Adrian Maulana Muhammad 06111540000099**

**Admiral Nelson S. A. Litik 06111840000086**

**Dosen Pengampu:**

Dr. Imam Mukhlash, S.Si, MT

**DEPARTEMEN MATEMATIKA**

**FAKULTAS SAINS DAN ANALITIKA DATA**

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**SURABAYA**

**2021**

**DAFTAR ISI**

**HALAMAN JUDUL**i

**DAFTAR ISI**ii

**BAB 1 PENDAHULUAN**1

**1.1 Latar Belakang**1

**1.2 Rumusan Masalah**3

**1.3 Batasan Masalah**3

**1.4 Tujuan Penelitian**3

**1.5 Manfaat Penelitian**3

**BAB 2 ANALISIS DAN PEMBAHASAN**4

**2.1 *Data Pre*-*processing***4

**2.1.1 *Data Cleaning***4

**2.1.2 *Data Transformation***4

**2.1.3 *Data Integration***5

**2.2 *Exploratory Data Analysis***6

**2.2.1 *Sales Trend Analysis***6

**2.3 *Market Basket Analysis***9

**2.3.1 Algoritma Apriori**9

**BAB 3 KESIMPULAN**12

**DAFTAR PUSTAKA**12

**LAMPIRAN**13

# BAB 1

**PENDAHULUAN**

* 1. **Latar Belakang**

Pada saat ini, bisnis ritel besar telah menawarkan skema *membership card* atau kartu keanggotaan kepada pelanggan mereka, hal tersebut membuat proses pengumpulan data pelanggan sekarang menjadi praktik komersial rutin. Pada awalnya, skema kartu keanggotaan diperkenalkan untuk menghargai pelanggan setia dan mendorong pelanggan tersebut untuk meningkatkan pembelian mereka pada bisnis, bisa dengan skema potongan harga atau *discount*.

Namun, bisnis tidak hanya ingin mengidentifikasi pelanggan setia untuk memberikan mereka penghargaan (atau menarik kembali pelanggan yang tidak lagi setia), tetapi mereka juga ingin memahami kebiasaan belanja semua pelanggan mereka untuk lebih menargetkan pemasaran dan promosi bisnis. Dengan cara ini, bisnis dapat mengidentifikasi hubungan antara produk yang dibeli dengan pelanggan mereka.

Tentu saja, pengelompokan pada setiap produk bisa jadi tidak praktis, sehingga bisnis perlu untuk melakukan pengelompokkan *itemset* yang dibeli pelanggan, biasa disebut *basket* atau keranjang. Satu keranjang mungkin berisi barang-barang yang mewakili kesenangan, sementara yang lainnya mewakili pembelian massal atau penghematan dari pelanggan. Berdasarkan pembelian mereka, pelanggan dapat ditautkan ke keranjang yang relevan, dengan memeriksa dari keranjang mana mereka telah membeli, berapa banyak dan berapa sering. Bisnis dapat memperoleh wawasan berharga tentang preferensi belanja pelanggan tersebut. Sementara data pelanggan sebagian besar digunakan untuk lebih memahami kebutuhan konsumen, ada manfaat operasional tambahan yang dapat diperoleh dari menganalisis data ini. Misalnya, data ini juga dapat digunakan untuk memahami permintaan dan untuk memastikan bahwa bisnis memiliki persediaan yang tepat.

Bagaimana dengan pelanggan yang tidak memiliki kartu keanggotaan? Semakin banyak bisnis yang mengumpulkan data pelanggan ini melalui sumber lain. Pada bisnis *e*-*commerce*, data pelanggan tersimpan dalam *database* mereka, data pelanggan yang tersimpan tersebut berupa informasi pembeli *online* dan kegiatan pembelian *online* mereka. *E-commerce* adalah perdagangan elektronik, yaitu penyebaran, pembelian, penjualan, pemasaran barang dan jasa melalui sistem elektronik seperti internet, televisi, dan jaringan komputer lainnya. Terdapat banyak cara yang berbeda di mana bisnis *e-commerce* dapat mengumpulkan data dari konsumen, seperti dengan survey, *e-mail subscriptions*, *web analytics*, dan cara lainnya. Penting bahwa bisnis *e-commerce* mengumpulkan data tersebut secara bertanggung jawab.

Proses analisis yang mengidentifikasi hubungan yang kuat antara produk yang dijual, disebut *market basket analysis*. *Market basket analysis* merupakan salah satu teknik dari *data mining* yang digunakan untuk menentukan produk-produk manakah yang akan dibeli oleh pelanggan secara bersamaan dengan melakukan analisis terhadap daftar transaksi pelanggan. Dengan mengetahui produk-produk tersebut, maka sebuah sistem *e-commerce* dapat membuat atau mengembangkan sebuah sistem *customer profiles* dan dapat menentukan *layout catalog* pelanggannya tersebut.

Terdapat beberapa keuntungan dalam menganalisis data pembelian untuk memahami perilaku konsumen, seperti sebagai berikut:

1. *Behavioral Targeting*

*Big data* dapat membantu bisnis memahami pelanggan mereka dengan lebih baik, seperti wawasan tentang produk yang mereka sukai, alasan mereka meninggalkan troli tanpa meneruskan ke proses pembayaran, waktu dalam sehari dan bulan dalam setahun yang paling banyak dikunjungi pelanggan, dan berbagai keputusan bisnis penting lainnya. Bisnis dapat merencanakan *marketing campaign* berdasarkan wawasan ini.

1. *E-Commerce Personalization*

Pelanggan suka melihat diskon, produk yang direkomendasikan, dan yang terpenting, mereka menyukai “*personal touch*” dari bisnis. Hal ini tidak mungkin terjadi tanpa ketersediaan data: Apa yang cenderung mereka beli, berapa banyak waktu yang mereka habiskan untuk menelusuri produk, berapa banyak waktu yang mereka habiskan per-transaksi, seberapa sering mereka mengunjungi situs ­­*e-commerce*, seberapa sering mereka meninggalkan keranjang belanja mereka dan banyak informasi lainnya.

1. *Dynamic Pricing*

Dengan data yang cukup, pelanggan dapat diukur untuk pola pengeluaran dan perilaku mereka di tempat. Setelah persona dikaitkan dengan pelanggan, harga dapat bervariasi untuk menarik mereka kembali ke situs. Jika mereka pergi, diskon dinamis adalah alat yang ampuh tetapi hanya berhasil jika dijalankan dengan data yang cukup.

1. *End-To-End Analytics*

Bisnis dapat menggunakan *big data* untuk memahami bukan hanya pelanggan mereka, tetapi juga bagaimana kinerja seluruh bisnis: persediaan, pengiriman, inventaris, dan penjualan. Wawasan ini dapat digunakan untuk memperkirakan arus kas masa depan, volume barang dagangan, tingkta akuisisi pelanggan, dan parameter lain yang penting bagi keberhasilan bisnis.

Berdasarkan latar belakang yang telah diuraikan di atas, penulis akan melakukan penelitian tentang analisis data penjualan pada bisnis *e-commerce* dengan menggunakan pendekatan *market basket analysis*. Data yang digunakan pada penelitian ini adalah data penjualan ­*website e-commerce* dari India yang diperoleh dari situs Kaggle. Pada situs tersebut terdapat tiga *dataset* yaitu *List of Orders, Order Details,* dan *Sales Target*. Tabel 1.1 menunjukkan atribut beserta deskripsi dari atribut pada data yang akan digunakan pada penelitian ini.

|  |  |
| --- | --- |
| **Atribut** | **Deskripsi** |
| *Order* ID | ID pembelian produk |
| *Order Date* | Tanggal pembelian produk |
| *Amount* | Jumlah harga pembelian |
| *Profit* | Keuntungan yang diperoleh dari pembelian |
| *Quantity* | Jumlah pembelian |
| *Category* | Kategori dari produk |
| *Sub-Category* | Sub-kategori dari produk |
| *Customer Name* | Nama dari pembeli |
| *State* | Negara asal pembeli |
| *City* | Kota asal pembeli |
| *Month of Order Date* | Tanggal pembelian per-bulan |
| *Target* | Target penjualan bulanan |

**Tabel 1.1** Atribut dan deskripsi atribut dari data

* 1. **Rumusan Masalah**

Berdasarkan latar belakang yang telah diuraikan di atas, permasalahan yang akan dibahas dalam penelitian ini adalah sebagai berikut:

1. Bagaimana implementasi *data mining* dalam menganalisis transaksi pembelian konsumen pada bisnis *e-commerce*?
2. Bagaimana implementasi *data mining* dalam mengevaluasi performa penjualan pada bisnis *e-commerce*?
   1. **Batasan Masalah**

Penulis membatasi permasalahan pada penelitian ini sebagai berikut:

1. Data yang akan dianalisis adalah data penjualan ­*website e-commerce* dari India yang diperoleh dari situs Kaggle.
2. Proses analisis menggunakan bahasa pemograman Python.
3. Metode yang digunakan adalah *basket market analysis* dan *exploratory data analysis.*
   1. **Tujuan Penelitian**

Tujuan dalam penelitian ini adalah sebagai berikut:

1. Menganalisis transaksi pembelian konsumen pada bisnis *e-commerce* dengan metode *basket market analysis*.
2. Mengevaluasi performa penjualan pada bisnis ­*e-commerce* dengan metode *exploratory data analysis*.
   1. **Manfaat Penelitian**

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Bisnis *e-commerce* dapat memperoleh wawasan berharga tentang preferensi belanja pelanggan mereka.
2. Bisnis *e-commerce* dapat mengevaluasi performa penjualan mereka.
3. Dalam bidang akademik, penelitian ini mempunyai manfaat untuk mengetahui dan memahami pendekatan *data mining* dalam bisnis *e-commerce*.

# BAB 2

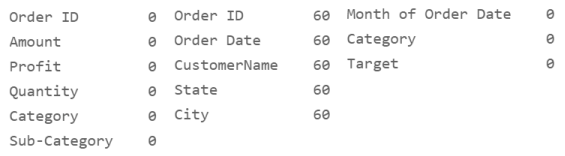
**ANALISIS DAN PEMBAHASAN**

**2.1 *Data Pre-Processing***

Pada tahap ini, penulis akan melakukan *data pre-processing*, yaitu tahap di mana data disiapkan terlebih dahulu sebelum dilanjutkan pada tahap analisis. Tahapan pada *data pre-processing* meliputi *data cleaning, data integration,* dan *data transformation*. Terdapat tiga *dataset* dengan tipe CSV pada penelitian ini, yaitu *List of Orders, Order Details,* dan *Sales Target*.

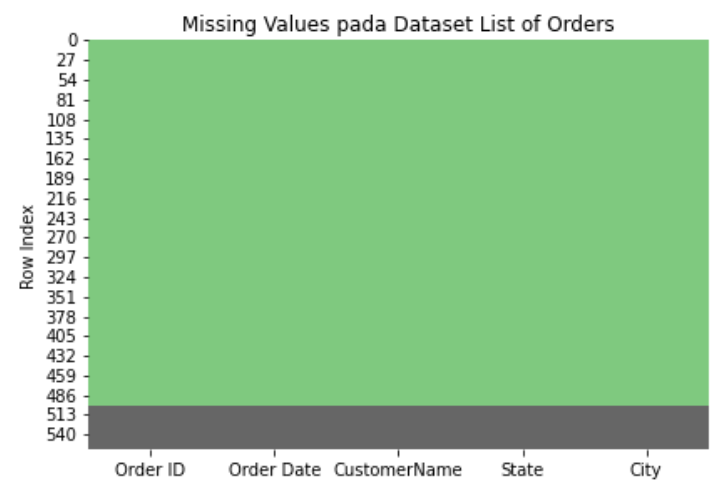
**2.1.1 *Data Cleaning***

Pertama, penulis akan mencari keberadaan *missing values* pada masing-masing *dataset*. Pada Gamber 2.1 ditunjukkan jumlah *missing values* pada setiap kolom.



**Gambar 2.1** Jumlah missing values pada setiap kolom

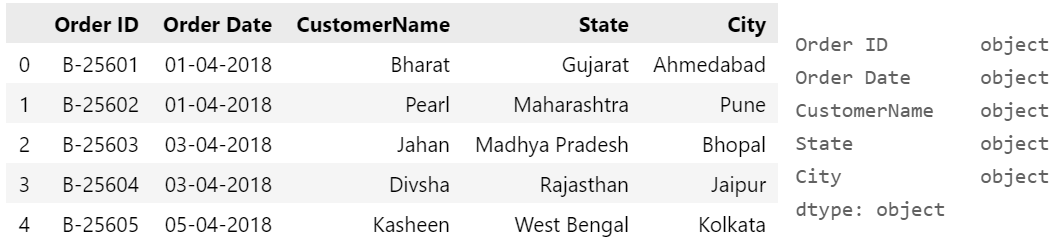
Terlihat bahwa, *missing values* hanya terdapat pada *dataset List of Orders* sebanyak 60 *missing values* pada masing-masing kolom. Selanjutnya, diketahui bahwa posisi indeks baris pada *missing values* tersebut adalah sama, yaitu pada baris 500 sampai dengan 559, maka kita bisa *drop missing values* pada setiap kolom tersebut. Gambar 2.2 menunjukkan indeks *missing values* pada *dataset List of Orders*.



**Gambar 2.2** Indeks baris missing values pada setiap kolom

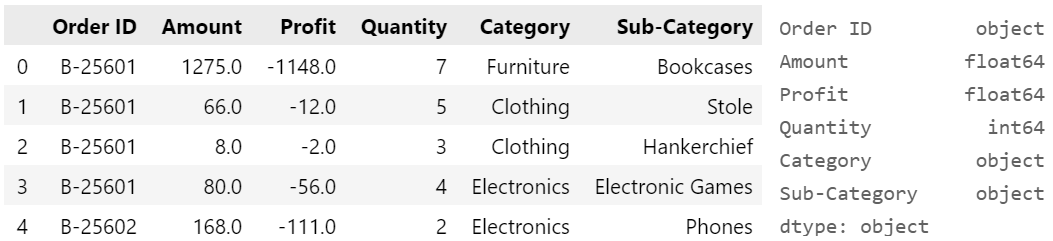
**2.1.2 *Data Transformation***

Tahap yang selanjutnya adalah *data transformation*,yaitu di mana data diubah dan dikonsolidasikan ke dalam bentuk yang sesuai untuk tahap berikutnya. Kita akan melihat tipe data dari setiap kolom pada seluruh *dataset*. Pertama kita lihat *dataset* *List of Orders*. Pada Gambar 2.3 menunjukkan bahwa terdapat kolom yang belum mempuyai tipe data yang sesuai. Oleh karena itu, kami akan mengkonversi kolom *Order Date* menjadi bertipe *datetime*.



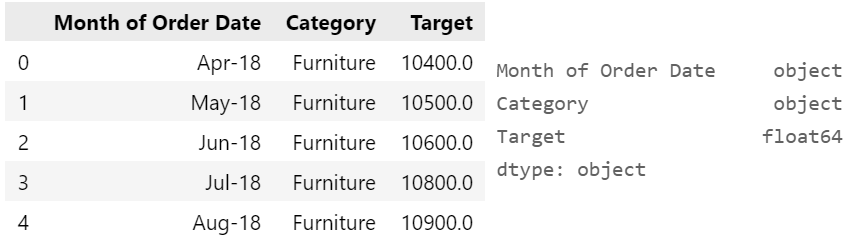
**Gambar 2.3** Lima baris teratas dataset List of Orders berserta tipe data mula-mula

Selanjutnya, kita lihat *dataset* *Order Details*. Pada Gambar 2.4 menunjukkan bahwa semua kolom sudah mempuyai tipe data yang sesuai. Oleh karena itu, kami tidak mengkonversi kolom apapun dari *dataset* ini.



**Gambar 2.4** Lima baris teratas dataset Order Details berserta tipe data mula-mula

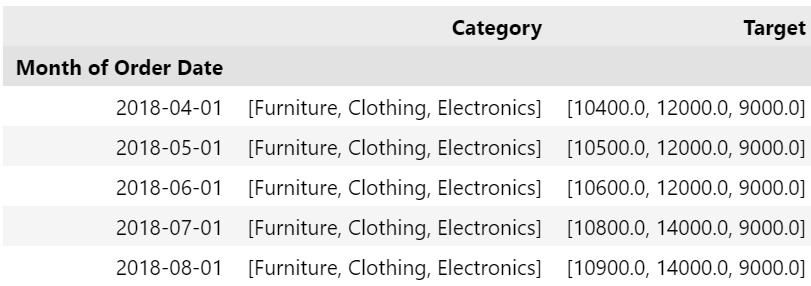
Selanjutnya, Pada Gambar 2.5 menunjukkan bahwa terdapat kolom yang belum mepunyai tipe data yang sesuai pada *dataset* *Sales Target*. Pada *dataset* ini, kami akan mengkonversi kolom *Month of Order Date* menjadi bertipe *datetime*.



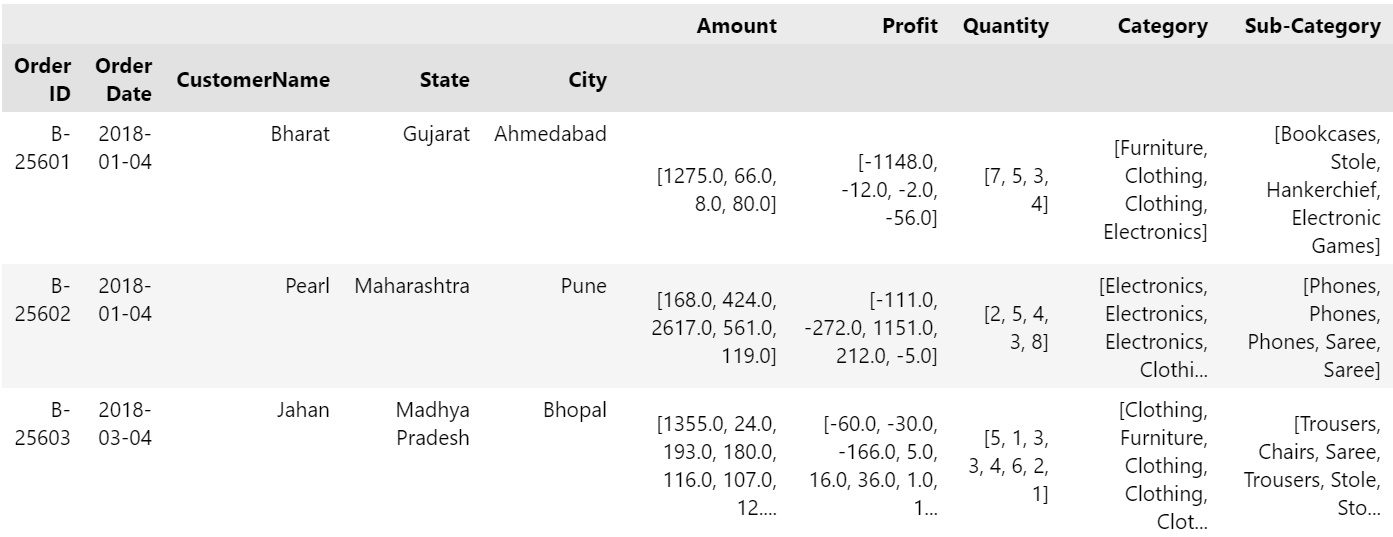
**Gambar 2.5** Lima baris teratas dataset Sales Target berserta tipe data mula-mula

**2.1.3 *Data Integration***

Tahap selanjutnya adalah *data integration*, yaitu di mana beberapa sumber data dapat digabungkan. Jika kita lihat ketiga *dataset* yang digunakan pada penelitian ini, ketiga *dataset* tersebut dapat dibagi menjadi dua kategori. Pertama, *dataset List of Orders* dan *Order Details* dapat dimasukkan ke dalam kategori data pemesanan, sedangkan *dataset Sales Target* termasuk kategori data evaluasi penjualan. Oleh karena itu, penulis akan menggabungkan *dataset* *List of Orders* dan *Order Details* menjadi satu *dataset* yang bernama *Orders*. Pada Gambar 2.6 dan Gambar 2.7 menunjukkan lima baris teratas pada *dataset Sales Target* dan *Orders* yang akan digunakan pada tahap analisis selanjutnya pada penelitian ini.



**Gambar 2.6** Lima baris teratas dari dataset Sales Target



**Gambar 2.7** Tiga baris teratas dari dataset Orders

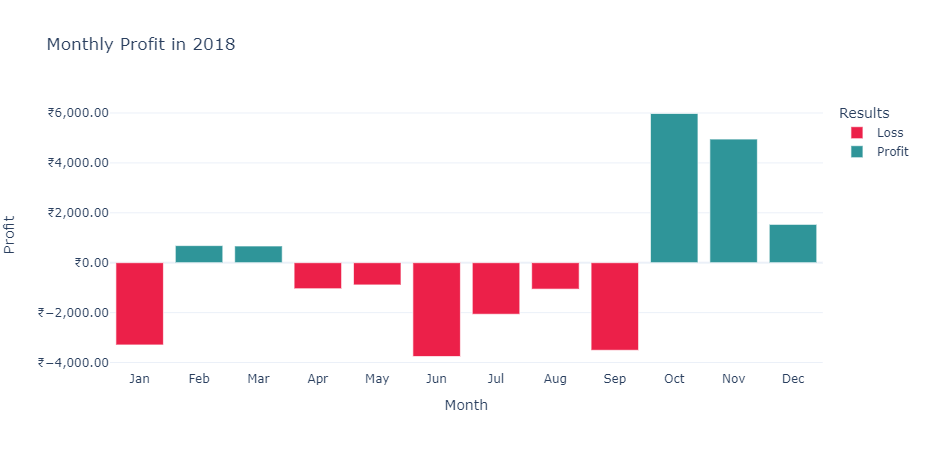
**2.2 *Exploratory Data Analysis* (EDA)**

Tahap selanjutnya adalah *Exploratory Data Analysis* (EDA) di mana merupakan pendekatan menganalisis kumpulan data untuk meringkas karakteristik utama dari data tersebut, sering dengan menggunakan grafik statistik dan metode visualisasi data lainnya. EDA dapat melihat apa yang data dapat beritahu di luar pemodelan formal atau *hypothesis testing*.

Pada konteks *dataset* yang telah kita miliki, EDA yang dapat kita lakukan adalah berupa *sales analysis*. *Sales analysis* adalah salah satu implementasi EDAyang bertujuan untuk mengevaluasi kinerja penjualan pada suatu perusahaan terhadap *goals* yang dimiliki perusahaan tersebut. Hal ini dapat memberikan *insights* tentang produk-produk yang mempunyai kinerja yang baik atau buruk terkait aktivitas penjualan. Wawasan atau *insights* tersebut penting bagi perusahaan agar dapat membuat suatu keputusan bisnis yang lebih bersifat ­*data-driven*, mengidentifikasi pelanggan yang paling *profitable*, mendapatkan wawasan atau kesadaran tentang tren pasar, dan memperluas jangkauan pasar.

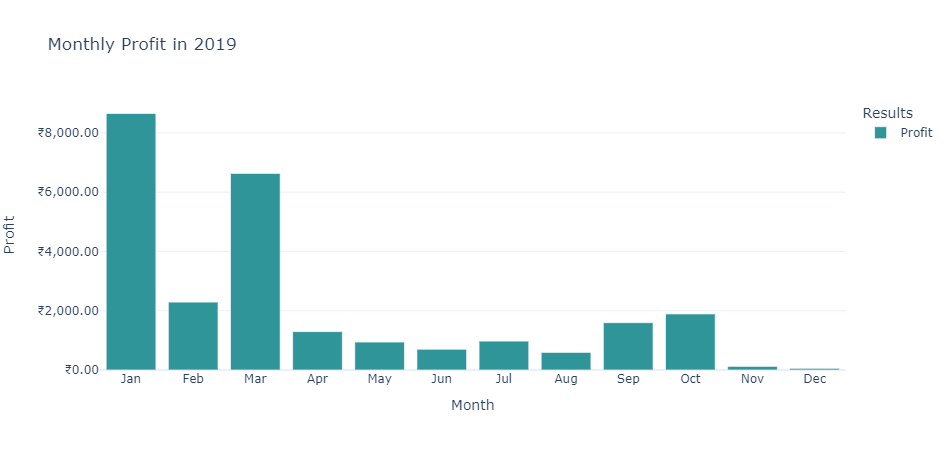
**2.2.1 *Sales Trend Analysis***

Analisis tren bertujuan untuk menemukan pola dalam data, sebuah tren adalah pergeseran ke atas atau ke bawah dalam kumpulan data dari waktu ke waktu. Dalam bisnis, analisis ini dapat berguna untuk *budgeting*. Kita akan melihat tren keuntungan bulanan di tahun 2018, tren tersebut ditunjukkan pada Gambar 2.8. Terlihat bahwa keuntungan yang tertinggi pada tahun 2018 adalah pada bulan Oktober, dan terendah pada bulan Juni.



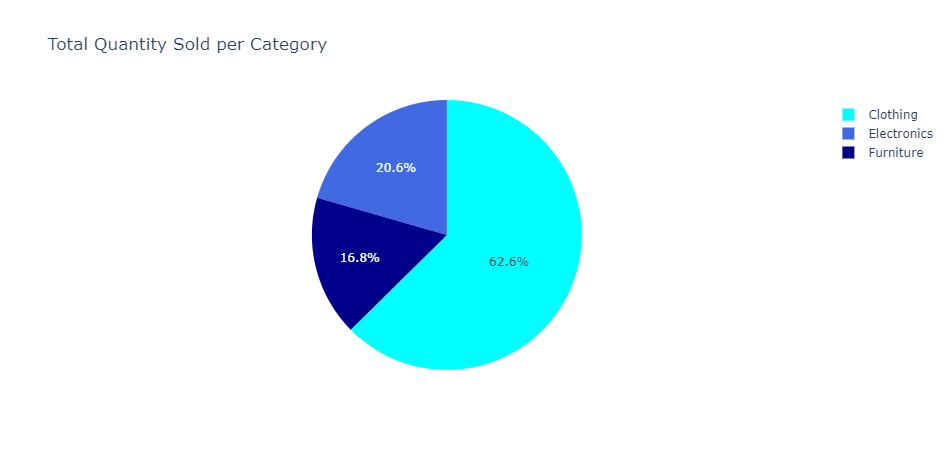
**Gambar 2.8** Profit pada tahun 2018

Lalu, tren keuntungan bulanan pada tahun 2019, ditunjukkan pada Gambar 2.9. Terlihat bahwa keuntungan tertinggi pada 2019 adalah pada bulan Januari dan terendah adalah pada Desember.



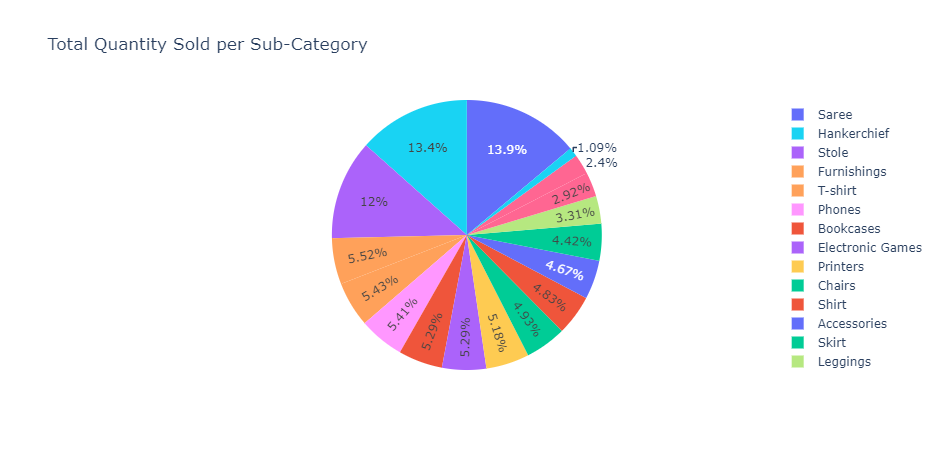
**Gambar 2.9** Profit pada tahun 2019

Selanjutnya, kita ingin mengetahui jenis produk apa yang paling sering dibeli pelanggan, *insights* tersebut dapat ditunjukkan pada Gambar 2.10. Terlihat bahwa jenis produk yang paling banyak dibeli adalah produk dengan jenis *clothing* atau baju.



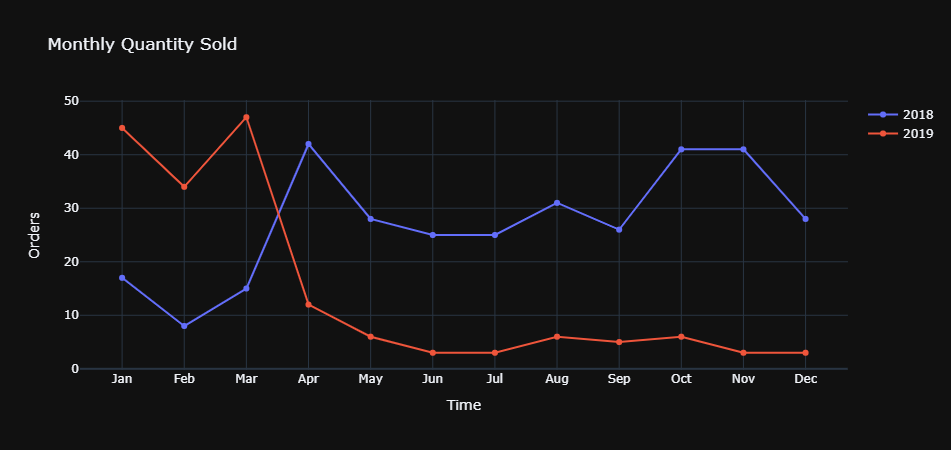
**Gambar 2.10** Jenis produk yang paling banyak dibeli

Kalau sebelumnya telah ditunjukkan jenis produk yang sering dibeli pelanggan, sekarang kita ingin mengetahui produk apa yang paling sering dibeli oleh pelanggan, *insights* tersebut ditunjukkan pada Gambar 2.11. Terlihat bahwa produk yang paling sering dibeli oleh pelanggan adalah berupa *saree*, dan yang paling sedikit dibeli oleh pelanggan adalah *tables* atau meja.



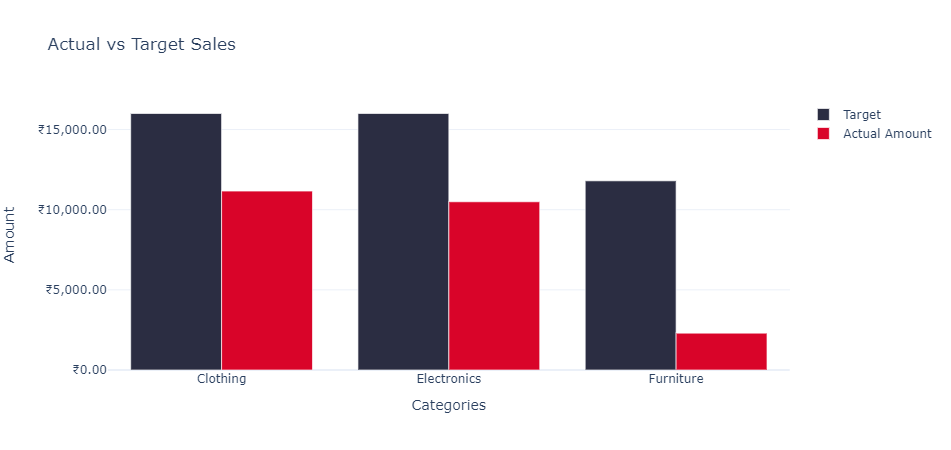
**Gambar 2.11** Produk yang paling sering dibeli pelanggan

Selanjutnya, kami ingin mengetahui tren banyaknya produk yang dibeli pada tahun 2018 dan 2019, *insights* tersebut ditunjukkan pada Gambar 2.12. Terlihat bahwa pada bulan Januari sampai Maret 2019 tren banyaknya produk yang dibeli lebih tinggi daripada tahun sebelumnya. Sedangkan, pada bulan April sampai Desember 2018 tren banyaknya produk yang dibeli lebih tinggi daripada tahun 2019.



**Gambar 2.12** Tren banyaknya produk yang dibeli pada tahun 2018 dan 2019

Selanjutnya, kami ingin mengetahui apakah banyaknya produk yang dibeli sudah mencapai target dari perusahaan, *insights* tersebut ditunjukkan pada Gambar 2.13. Terlihat bahwa seluruh target pada setiap kategori masih belum tercapai.



**Gambar 2.13** Banyak produk yang dibeli dan target perusahaan

**2.3 *Market Basket Analysis***

Diambil dari buku *“Discovering Knowledge in Data: An Introduction to Data Mining”* yang ditulis oleh Larose, ­*data mining* dapat dibagi menjadi beberapa kelompok berdasarkan tugas atau pengerjaan yang dilakukan. Salah satu contoh pengerjaannya yaitu asosiasi. Asosiasi, atau yang lebih umum disebut sebagai *market basket analysis* di dunia bisnis, bertugas dalam menemukan atribut yang muncul dalam satu waktu.

Contoh pengimplementasian asosiasi adalah meneliti jumlah pelanggan yang diharapkan untuk memberikan respon positif terhadap suatu produk atau menentukan barang supermarket yang perlu untuk direkomendasikan dan tidak direkomendasikan. Informasi dari *market basket analysis* membantu menganalisis kemungkinan produk yang dibeli bersama dari kategori yang berbeda. Pengetahuan ini memperbesar kemungkinan para manajer untuk mengembangkan intervensi guna mempengaruhi perilaku pembelian, mempromosikan kategori produk tertentu, atau penawaran promosi untuk penjualan produk yang cenderung meningkatkan angka keuntungan.

**2.3.1 Algoritma Apriori**

Algoritma apriori adalah salah satu algoritma dasar yang diusung oleh Agrawal dan Srikan pada tahun 1994 untuk menemukan *frequent itemset* pada aturan asosiasi *Boolean*. *Frequent itemset* merupakan sekumpulan *item* yang sering muncul secara bersamaan. Ide utama pada algoritma ini adalah dimulai dengan mencari *frequent itemset* dari basis transaksi, dilanjutkan dengan menghilangkan *itemset* dengan frekuensi yang rendah berdasarkan level minimum *support* yang telah ditentukan sebelumnya. Dua tolok ukur dari asosiasi, *support* dan *confidence*, menentukan penting atau tidaknya suatu asosiasi. *Support* adalah nilai penunjang atau persentase kombinasi sebuah *item* dalam basis data, sementara *confidence* adalah nilai kepastian; kuatnya hubungan antar *item* dalam aturan asosiasi. Nilai *support* dari sebuah *item* dapat diperoleh dengan rumus berikut:

(2.1)

Sedangkan untuk nilai *support* dari dua *item* diperoleh dari rumus berikut:

(2.2)

Nilai *confidence* dari aturan “jika A maka B” diperoleh dengan rumus berikut:

(2.3)

Selain *support* dan *confidence*, *lift ratio* juga merupakan parameter penting dalam asosiasi. *Lift ratio* mengukur seberapa penting *rule* yang telah terbentuk berdasarkan nilai dari *support* dan *confidence*. *Lift ratio* menunjukkan validitas proses transaksi dan memberikan informasi jika *item* A direkomendasikan untuk dibeli bersamaan dengan *item* B. Nilai *lift* dapat diperoleh dari rumus berikut:

(2.4)

Dari seluruh atribut yang tersedia, diambil atribut *“Sub-Category”* yang menjelaskan tentang produk yang dibeli oleh tiap konsumen. Kolom akan dikonversikan menjadi beberapa *list* yang merepresentasikan suatu *itemset*.

Graphical user interface, text, application

Description automatically generated

**Gambar 2.14** Konversi kolom menjadi list

Tahapan dilanjutkan dengan mengimplementasikan algoritma apriori untuk menghilangkan *infrequent itemsets*, sehingga perlu menerapkan *one-hot-encoder* terhadap data yang sudah ada. Proses ini dapat dilakukan dengan modul *mlxtend* yang sudah tersedia di *Python*.

Table

Description automatically generated

**Gambar 2.15** one-hot-encoder terhadap data

Secara spesifik, algoritma apriori akan digunakan untuk mengidentifikasi *itemsets* yang memiliki nilai *support* lebih dari sama dengan 0,1.

Table

Description automatically generated

**Gambar 2.16** itemset yang memiliki support lebih atau sama dengan 0,1

Setelah itu, fungsi *association\_rules* dari modul *mlxtend* akan membantu menemukan produk dengan tipe asosiasi yang menemui kebutuhan. Nilai *lift* yang digunakan adalah 1 sebagai persyaratan menentukan *itemset* yang diharapkan.

Table

Description automatically generated

**Gambar 2. 17** Association rules

Berdasarkan hasil yang didapat, *itemset* *“handkerchief”* dan *“stoles”* memiliki nilai *lift* yang tertinggi. Kedua produk ini patut ditelaah lebih lanjut untuk mengidentifikasi alasan peningkatan angka penjualannya. Meskipun data menunjukkan bahwa kedua produk ini memiliki potensi yang tinggi di pasar, perlu dilakukan eksplorasi lebih lanjut mengenai korelasi antara kedua produk sebelum diambil keputusan akhir.

**BAB 3**

**KESIMPULAN**

1. Berdasarkan hasil yang didapatkan dari metode *Exploratory Data Analysis* (EDA):
   1. Profit paling tinggi pada tahun 2018 adalah pada bulan Oktober dan paling rendah pada bulan Juni. Sedangkan pada tahun 2019, profit paling tinggi terdapat pada bulan Januari dan paling rendah pada bulan Desember.
   2. Jenis produk yang paling banyak dibeli adalah *clothing* atau pakaian dan yang paling rendah adalah *furniture*.
   3. Produk yang paling banyak dibeli pelanggan adalah *saree* dan yang paling sedikit dibeli adalah *tables* atau meja.
   4. Tren banyaknya produk terjual pada bulan Januari sampai Maret 2018 lebih unggul daripada tahun 2019, namun setelahnya tren banyaknya produk yang terjual pada tahun 2018 terus menurun, hingga dikalahkan oleh tren pada tahun 2019.
   5. Banyaknya penjualan produk dari seluruh kategori masih belum sesuai oleh target perusahaan.
2. Berdasarkan hasil yang didapatkan dasri metode *Market Basket Analysis*:
   1. Berdasarkan hasil yang didapat, *itemset* *“handkerchief”* dan *“stoles”* memiliki nilai *lift* yang tertinggi. Kedua produk ini patut ditelaah lebih lanjut untuk mengidentifikasi alasan peningkatan angka penjualannya.
   2. Meskipun data menunjukkan bahwa kedua produk ini memiliki potensi yang tinggi di pasar, perlu dilakukan eksplorasi lebih lanjut mengenai korelasi antara kedua produk sebelum diambil keputusan akhir.

**DAFTAR PUSTAKA**

1. Jiawei Han And Micheline Kamber. (2006). “ Data Mining : Concepts and Techiques ”. San Fransisco : Morgan Kaufmann Publishers
2. Putro, A. N. S., Ernawati, & Wisnubhadra, I. (2016). Market Basket Analysis Pada Magister Teknik Informatika , Universitas Atma Jaya Yogyakarta, 978–979
3. Solnet, D., Boztug, Y., & Dolnicar, S. (2016). An untapped gold mine? Exploring the potential of market basket analysis to grow hotel revenue. International Journal of Hospitality Management, 56, 119–125.
4. Muflikhah, L., Ratnawati, D. E., & Putri, R. R. M. (2018). Data Mining (Pertama). Malang: UB Press.
5. Badrul, M. (2016). Algoritma Asosiasi Dengan Algoritma Apriori Untuk Analisa Data Penjualan. Pilar Nusa Mandiri, XII(2), 121–129.
6. Widiati, E., & Evita Dewi, K. (2014). Implementasi Association Rule Terhadap Penyusunan Layout Makanan Dan Penentuan Paket Makanan Hemat Di RM Roso Echo Dengan Algoritma Apriori. Jurnal Ilmiah Komputer Dan Informatika (KOMPUTA), 3(2), 96–101.
7. Listriani, D., Setyaningrum, A. H., & M.A, F. E. (2016). Penerapan Metode Asosiasi Menggunakan Algoritma Apriori Pada Aplikasi Pola Belanja Konsumen ( Studi Kasus Toko Buku Gramedia Bintaro ). Jurnal Teknik Informatika Vol 9 No. 2, Universitas Islam Negeri Jakarta, 9(2), 120–127.

**LAMPIRAN**

# Import libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from IPython.display import display, Markdown

import plotly.express as px

import plotly.graph\_objs as go

# Import dataset

df\_LOD = pd.read\_csv(

    r"D:/Python Projects/Data Mining-Tubes/data/raw/List of Orders.csv"

)

df\_OD = pd.read\_csv(r"D:/Python Projects/Data Mining-Tubes/data/raw/Order Details.csv")

df\_ST = pd.read\_csv(r"D:/Python Projects/Data Mining-Tubes/data/raw/Sales Target.csv")

# Data cleaning: remove missing values

df\_LOD.dropna(inplace=True)

# Data transformation

df\_LOD["Order Date"] = pd.to\_datetime(df\_LOD["Order Date"])

df\_ST["Month of Order Date"] = pd.to\_datetime(

    df\_ST["Month of Order Date"], format="%b-%y"

)

# Data integration

df\_orders = (

    pd.merge(df\_LOD, df\_OD, on="Order ID")

    .groupby(["Order ID", "Order Date", "CustomerName", "State", "City"])

    .agg(

        {

            "Amount": lambda x: x.tolist(),

            "Profit": lambda x: x.tolist(),

            "Quantity": lambda x: x.tolist(),

            "Category": lambda x: x.tolist(),

            "Sub-Category": lambda x: x.tolist(),

        }

    )

)

df\_target = df\_ST.groupby(["Month of Order Date"]).agg(

    {"Category": lambda x: x.tolist(), "Target": lambda x: x.tolist()}

)

# Export the output to csv files

df\_orders.to\_csv(

    "D:\\Python Projects\\Data Mining-Tubes\\data\\post-preprocessing/orders.csv"

)

df\_target.to\_csv(

    "D:\\Python Projects\\Data Mining-Tubes\\data\\post-preprocessing/targets.csv"

)

df\_orders.reset\_index(inplace=True)

df\_STA = df\_orders[['Order Date', 'Amount', 'Profit', 'Quantity']].copy()

df\_STA.insert(loc=1, column='Year', value=pd.DatetimeIndex(df\_orders['Order Date']).year)

df\_STA.insert(loc=2, column='Month', value=pd.DatetimeIndex(df\_orders['Order Date']).month)

for col in ['Amount', 'Profit', 'Quantity']:

    df\_STA[col] = df\_STA[col].apply(sum)

year\_month = df\_STA.groupby(['Year', 'Month']).sum().sort\_values(['Year','Month'])

year\_month = year\_month.reset\_index()

year\_month["Color"] = np.where(year\_month["Profit"]<0, 'Loss', 'Profit')

year\_month\_2018 = year\_month[year\_month['Year']==2018]

fig = px.bar(year\_month\_2018, x='Month', y='Profit', color='Color',

             title="Monthly Profit in 2018",

             labels=dict(Month\_Number="Month", Profit="Profit", Color="Results"),

             color\_discrete\_map={

                 'Loss': '#EC2049',

                 'Profit': '#2F9599'},

             hover\_data=["Month", "Profit"],

             template='plotly\_white')

fig.update\_layout(yaxis\_tickprefix = '₹', yaxis\_tickformat = ',.2f')

fig.update\_layout(

    xaxis = dict(

        tickvals = [1, 2, 3, 4, 5, 6, 7,8 ,9, 10, 11, 12],

        ticktext = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']

    )

)

fig.show()

year\_month\_2019 = year\_month[year\_month['Year']==2019]

fig = px.bar(year\_month\_2019, x='Month', y='Profit', color='Color',

             title="Monthly Profit in 2019",

             labels=dict(Month\_Number="Month", Profit="Profit", Color="Results"),

             color\_discrete\_map={

                 'Loss': '#EC2049',

                 'Profit': '#2F9599'},

             hover\_data=["Month", "Profit"],

             template='plotly\_white')

fig.update\_layout(yaxis\_tickprefix = '₹', yaxis\_tickformat = ',.2f')

fig.update\_layout(

    xaxis = dict(

        tickvals = [1, 2, 3, 4, 5, 6, 7,8 ,9, 10, 11, 12],

        ticktext = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']

    )

)

fig.show()

details\_category = df\_OD.groupby('Category').sum().reset\_index()

fig = px.pie(details\_category, values='Quantity', names='Category', color='Category',

             color\_discrete\_map={'Clothing':'cyan',

                                 'Electronics':'royalblue',

                                 'Furniture':'darkblue'},

            title='Total Quantity Sold per Category')

fig.show()

details\_subcategory = df\_OD.groupby('Sub-Category').sum().reset\_index()

fig = px.pie(details\_subcategory, values='Quantity', names='Sub-Category', color='Sub-Category',

            title='Total Quantity Sold per Sub-Category')

fig.show()

date\_orders = df\_LOD.groupby('Order Date').size().reset\_index(name="Orders")

date\_orders['Month'] = pd.DatetimeIndex(date\_orders['Order Date']).month

date\_orders['Year'] = pd.DatetimeIndex(date\_orders['Order Date']).year

date\_orders\_2018 = date\_orders[date\_orders['Year']==2018]

date\_orders\_2019 = date\_orders[date\_orders['Year']==2019]

month\_2018 = date\_orders\_2018.groupby('Month').sum().reset\_index()

month\_2019 = date\_orders\_2019.groupby('Month').sum().reset\_index()

fig = go.Figure()

fig.add\_trace(go.Scatter(

    name='2018',

    x=month\_2018['Month'],

    y=month\_2018['Orders'],

    connectgaps=True # override default to connect the gaps

))

fig.add\_trace(go.Scatter(

    name='2019',

    x=month\_2019['Month'],

    y=month\_2019['Orders'],

    connectgaps=True # override default to connect the gaps

))

fig.update\_layout(title\_text='Monthly Quantity Sold',

                 template='plotly\_dark')

fig.update\_xaxes(title\_text='Time')

fig.update\_yaxes(title\_text='Orders')

fig.update\_layout(

    xaxis = dict(

        tickvals = [1, 2, 3, 4, 5, 6, 7,8 ,9, 10, 11, 12],

        ticktext = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']

    )

)

fig.layout.xaxis.fixedrange = True

fig.layout.yaxis.fixedrange = True

fig.show()

target\_category = df\_ST.groupby('Category').max().reset\_index()

details\_category = df\_OD.groupby('Category').sum().reset\_index()

target\_category['Actual\_Amount'] = details\_category['Profit']

fig = go.Figure(data=[

    go.Bar(name='Target', x=target\_category['Category'], y=target\_category['Target'],

          marker\_color='#2b2d42'),

    go.Bar(name='Actual Amount', x=target\_category['Category'], y=target\_category['Actual\_Amount'],

          marker\_color='#d90429')

])

fig.update\_layout(title\_text='Actual vs Target Sales',

                 template='plotly\_white')

fig.update\_xaxes(title\_text='Categories')

fig.update\_yaxes(title\_text='Amount')

fig.update\_layout(yaxis\_tickprefix = '₹', yaxis\_tickformat = ',.2f')

fig.layout.xaxis.fixedrange = True

fig.layout.yaxis.fixedrange = True

fig.show()