১) **One-Hot Encoding** কী এবং কেন প্রয়োজন?

One-Hot Encoding হলো এমন একটি Encoding টেকনিক, যেটা মূলত **Nominal categorical data** (যেখানে কোনো natural order নেই) কে সংখ্যায় রূপান্তর করে।

কারণ:

- মডেল টেক্সট/স্ট্রিং বোঝে না
- কিন্তু Nominal ক্যাটেগরিতে "বড়-ছোট" নেই
- তাই Ordinal/Label encoding দিয়ে 0,1,2 দিলে মডেল ভুলভাবে ধরে নেবে 2 > 1 > 0 অর্থাৎ একটা অর্ডার আছে

উদাহরণ:
Color = Red, Blue, Yellow
যদি Red=2, Blue=1, Yellow=0 দেন, মডেল ভুলভাবে ভাবতে পারে Red "বড়" বা "মোর ইম্পর্ট্যান্ট"।

One-Hot Encoding এই ভুল অর্ডার তৈরি হওয়া আটকায়।

২) **One-Hot Encoding** কীভাবে কাজ করে?

ধরুন একটি কলাম: Color
এর ইউনিক ভ্যালু তিনটি: Yellow, Blue, Red

One-Hot Encoding করলে:

- প্রতিটি ইউনিক ক্যাটেগরির জন্য আলাদা কলাম তৈরি হবে:
    - Color_Yellow
    - Color_Blue
    - Color_Red

তারপর প্রতিটি row-তে:

- যে ক্যাটেগরি আছে, তার কলামে 1
- বাকিগুলোতে 0

উদাহরণ:

- Yellow → [1, 0, 0]
- Blue → [0, 1, 0]
- Red → [0, 0, 1]

এভাবে একটি ক্যাটেগরি একটি "ভেক্টর" হিসেবে represent হয়। এবং কোনো অর্ডার তৈরি হয় না।

৩) **Dummy Variable Trap** এবং “একটা কলাম কেন ড্রপ করা হয়?”

One-hot encoding থেকে যে নতুন কলামগুলো তৈরি হয়, এগুলোকে অনেক সময় dummy variables বলা হয়।

সমস্যা কোথায়?
ধরুন তিনটা কলাম:

- Color_Yellow
- Color_Blue
- Color_Red

এখন লক্ষ্য করুন:
একটা row-তে এই তিনটার যোগফল সবসময় 1। অর্থাৎ:
Color_Red = 1 − (Color_Yellow + Color_Blue)

এতে কলামগুলো একে অপরের সাথে গণিতগতভাবে নির্ভরশীল হয়ে যায়। এটাকে বলা হয় **multicollinearity**।
এর ফলে কিছু মডেলে (বিশেষ করে linear/logistic regression) সমস্যা হতে পারে:

- coefficient estimate unstable
- interpretation কঠিন
- numerical issues

সমাধান:

- k ক্যাটেগরি থাকলে k-1 কলাম রাখা
- একটি reference category বাদ দেওয়া

এটাকে সাধারণভাবে **n−1 encoding** বলা হয়।

উদাহরণ:
Red, Blue, Yellow (3 ক্যাটেগরি)
তাহলে রাখবেন 2টা কলাম।
যদি দুইটাই 0 হয়, তাহলে মডেল বুঝবে “বাদ দেওয়া ক্যাটেগরি”।

গুরুত্বপূর্ণ:
Tree-based model এ multicollinearity সাধারণত বড় সমস্যা না, কিন্তু linear family-এর জন্য এটা best practice।

৪) ক্যাটেগরি অনেক বেশি হলে সমস্যা কী এবং কী করা হয়?

High-cardinality feature মানে:

- একটি কলামে অনেক ইউনিক ক্যাটেগরি
- যেমন brand 50টা, city 300টা

One-hot encoding করলে:

- অনেকগুলো নতুন কলাম তৈরি হবে
- dataset হয়ে যাবে খুব wide
- memory এবং training time বেড়ে যাবে
- overfitting-এর ঝুঁকিও বাড়ে

প্র্যাকটিক্যাল সমাধান :

1. top frequent ক্যাটেগরি আলাদা রাখবেন
2. rare/uncommon ক্যাটেগরি একত্র করে “Others” বানাবেন

উদাহরণ:
Brand এ 50টা থাকলে:

- Top 10 ব্র্যান্ড রাখলেন
- বাকিগুলো “Others”

এতে:

- কলাম সংখ্যা কমে
- মডেল বেশি generalize করতে পারে

আরেকটি বাস্তব নিয়ম:

- frequency threshold সেট করা (যেমন <100 occurrences হলে Others)

৫) Pandas বনাম Scikit-learn: কোনটা কখন?

(A) Pandas `get_dummies`

সুবিধা:

- খুব দ্রুত, এক লাইনে কাজ হয়
- exploratory কাজের জন্য ভালো

ঝুঁকি/সীমাবদ্ধতা (প্রোডাকশন/ML workflow এ):

- train এবং test এ category set আলাদা হলে কলাম mismatch হতে পারে

- pipeline-এর অংশ হিসেবে consistent column ordering/handling কঠিন
- unseen categories handle করা কঠিন

সোজা কথা:

- শেখা/EDA তে ঠিক আছে
- কিন্তু robust ML pipeline এ সাধারণত সুপারিশ করা হয় না

(B) Scikit-learn `OneHotEncoder`

সুবিধা:

- train এ fit করে mapping শেখে
- test এ transform করলে consistent কলাম তৈরি করে
- pipeline/ColumnTransformer এর সাথে সুন্দরভাবে বসে
- unseen category handle করার অপশন থাকে (handle_unknown)

Common parameters:

- `drop='first'` → dummy trap এড়াতে 1টা কলাম বাদ
- `sparse_output=True/False` → sparse matrix বা dense array
    - বড় ডেটাতে sparse ভালো (কম মেমোরি লাগে)

প্রোডাকশন/রিয়েল প্রজেক্টে সাধারণত এটিই ব্যবহার করা হয়।

৬) Fit/Transform নিয়ম (অবশ্যই মানতে হবে)

Encoding-এর ক্ষেত্রেও একই নিয়ম:

1. আগে train-test split
2. encoder কে শুধু train এ fit
3. train এবং test উভয় জায়গায় transform

কারণ:

- test data দিয়ে আগে mapping শিখে গেলে leakage হয়
- real-world performance ভুলভাবে বেশি দেখাতে পারে

৭) ColumnTransformer কেন বলা হয়?

বাস্তবে dataset এ একসাথে থাকে:

- কিছু numeric column (এগুলোতে scaling লাগবে)
- কিছু nominal categorical (এগুলোতে one-hot লাগবে)
- কিছু ordinal categorical (এগুলোতে ordinal encoding লাগবে)

সবগুলোকে আলাদা আলাদা করে handle করতে গেলে জটিল হয়।
ColumnTransformer দিয়ে:

- এক লাইনে নির্দিষ্ট কলামে নির্দিষ্ট preprocessing বসানো যায়
- পুরো preprocessing pipeline consistent থাকে

এটা বাস্তব ML workflow-এর স্ট্যান্ডার্ড পদ্ধতি।

৮) সিদ্ধান্ত

1. Nominal categorical feature (order নেই) → One-Hot Encoding
2. Dummy trap এড়াতে linear/logistic regression এ সাধারণত drop one category (n−1)
3. High-cardinality হলে rare categories → “Others”
4. Pandas get_dummies → শেখা/EDA
5. Scikit-learn OneHotEncoder + ColumnTransformer → প্রোডাকশন/প্রজেক্ট
6. split আগে, fit শুধু train এ, transform train+test এ