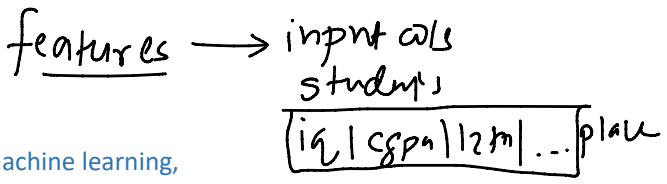
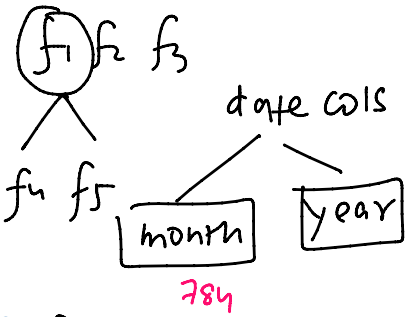
What is Feature Engineering?

07 February 2024 10:31

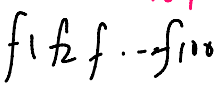


Feature engineering is a crucial step in the data pre-processing phase of machine learning, where data scientists and machine learning engineers create new features or modify existing ones to improve the performance of machine learning models. The goal of feature engineering is to provide the model with informative, non-redundant, and interpretable data that captures the underlying structure of the dataset. This process can significantly enhance model accuracy and performance by leveraging domain knowledge and mathematical transformations.

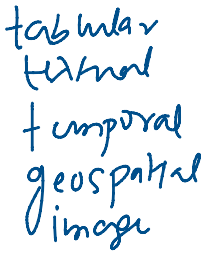
Key Aspects of Feature Engineering Include:



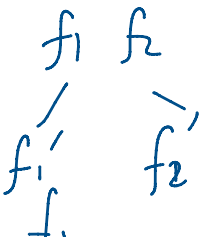
1. Creation of New Features: Involves generating new features from the existing data, which might be more relevant to the prediction task. This could include combining two or more features, extracting parts of a date-time stamp (like the day of the week, month, or year), or creating interaction terms that capture the relationship between different variables.



2. Feature Transformation: Applying transformations to features to change their distribution or scale. Common transformations include normalization, standardization, log transformation, and power transformations. These are especially important for algorithms that assume data is normally distributed or algorithms sensitive to the scale of features, like k-nearest Neighbors (KNN) and gradient descent-based algorithms.

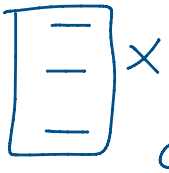


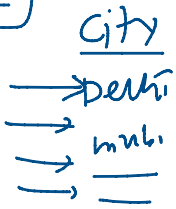
3. Feature Selection: Identifying the most relevant features to use in model training. This involves removing irrelevant, redundant, or noisy data that can detract from model performance. Techniques for feature selection include filter methods, wrapper methods, and embedded methods.



4. Feature Extraction: Techniques like Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and t-distributed Stochastic Neighbour Embedding (t-SNE) are used to reduce the number of features in a dataset while retaining as much of the variance in the data as possible. This is particularly useful for high-dimensional data.

5. Handling Missing Values: Developing strategies for dealing with missing data, such as imputation (filling in missing values with the mean, median, mode, or using more complex algorithms), or creating binary indicators that signal whether data was missing.





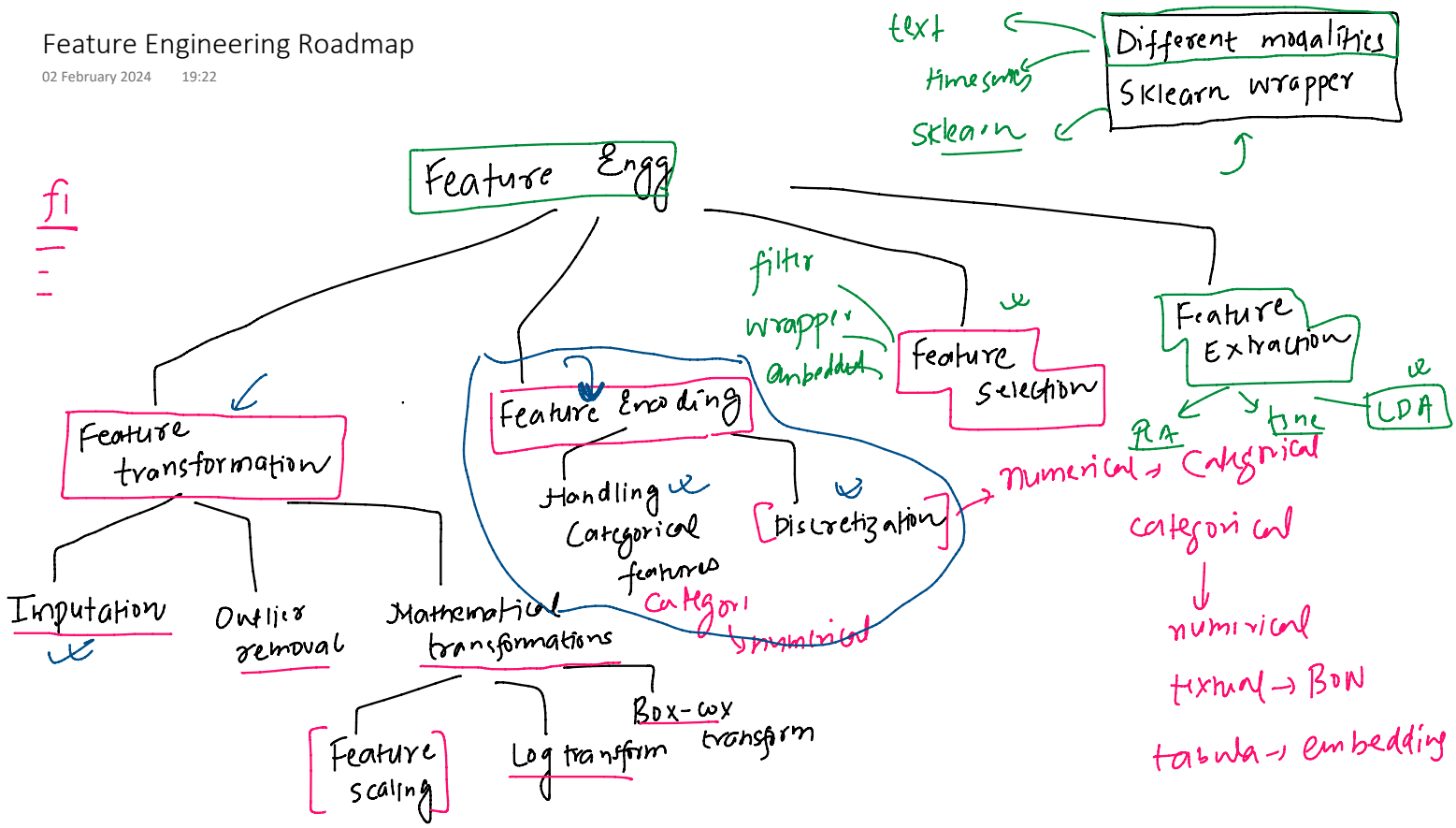
6. Encoding Categorical Variables: Converting categorical variables into a form that can be provided to ML models to improve performance. This includes using techniques like one-hot encoding, label encoding, and target encoding.

7. Working with different modalities: Feature engineering also includes applying all the above techniques to different modalities of data like temporal, textual and geospatial data.

Feature engineering is often considered more of an art than a science, requiring creativity, intuition, and domain knowledge. The quality and relevance of the features used can often make a more significant difference in the performance of a machine learning model than the choice of model itself. It enables models to learn better from the data, leading to more accurate predictions.

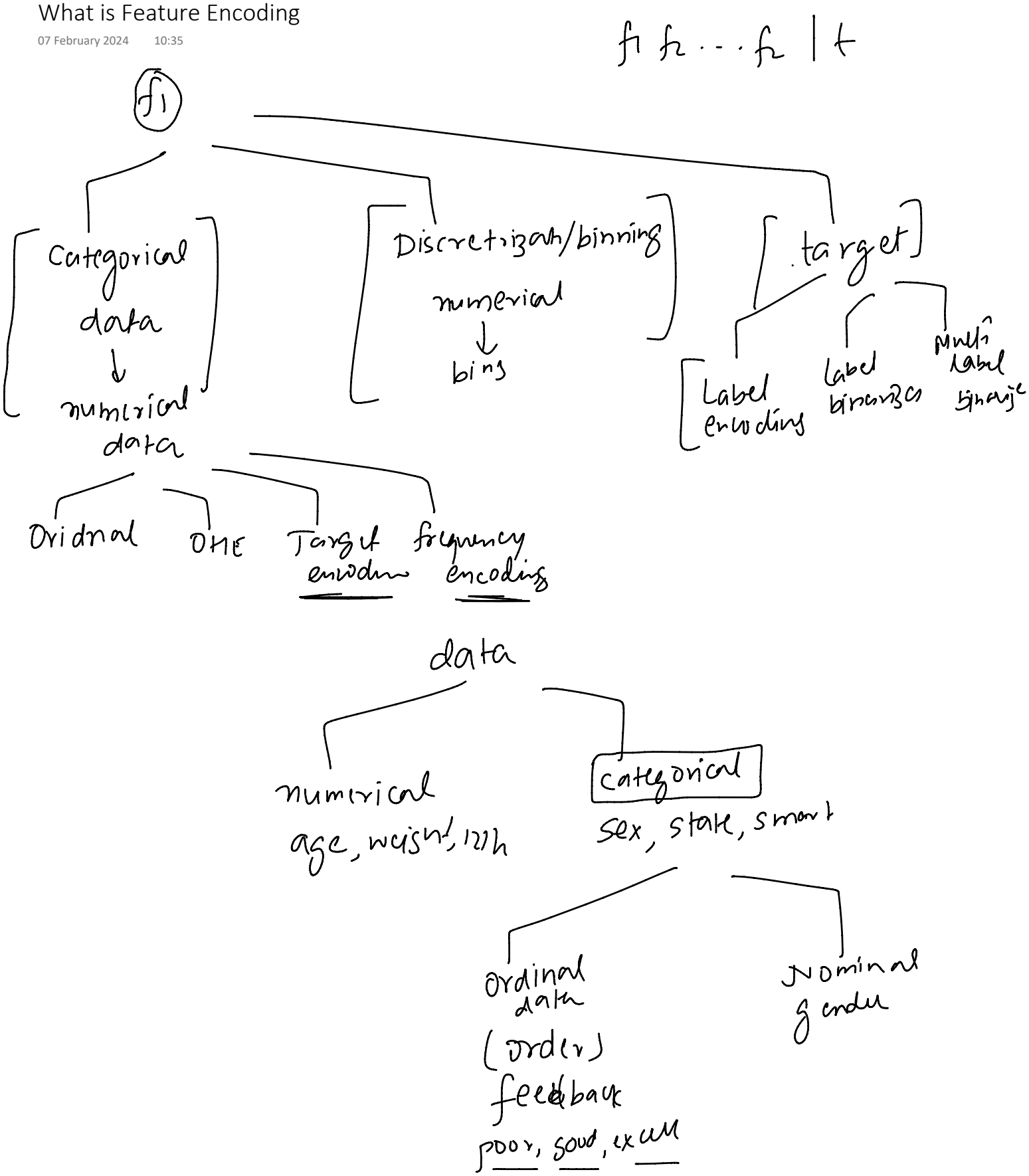
Feature Engineering Roadmap

02 February 2024 19:22



What is Feature Encoding

07 February 2024 10:35

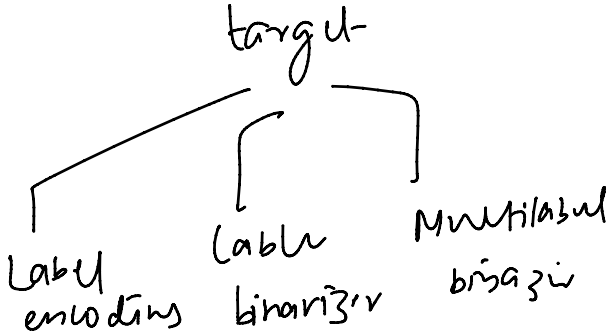


Ordinal encoding

(ordinal)

OHE

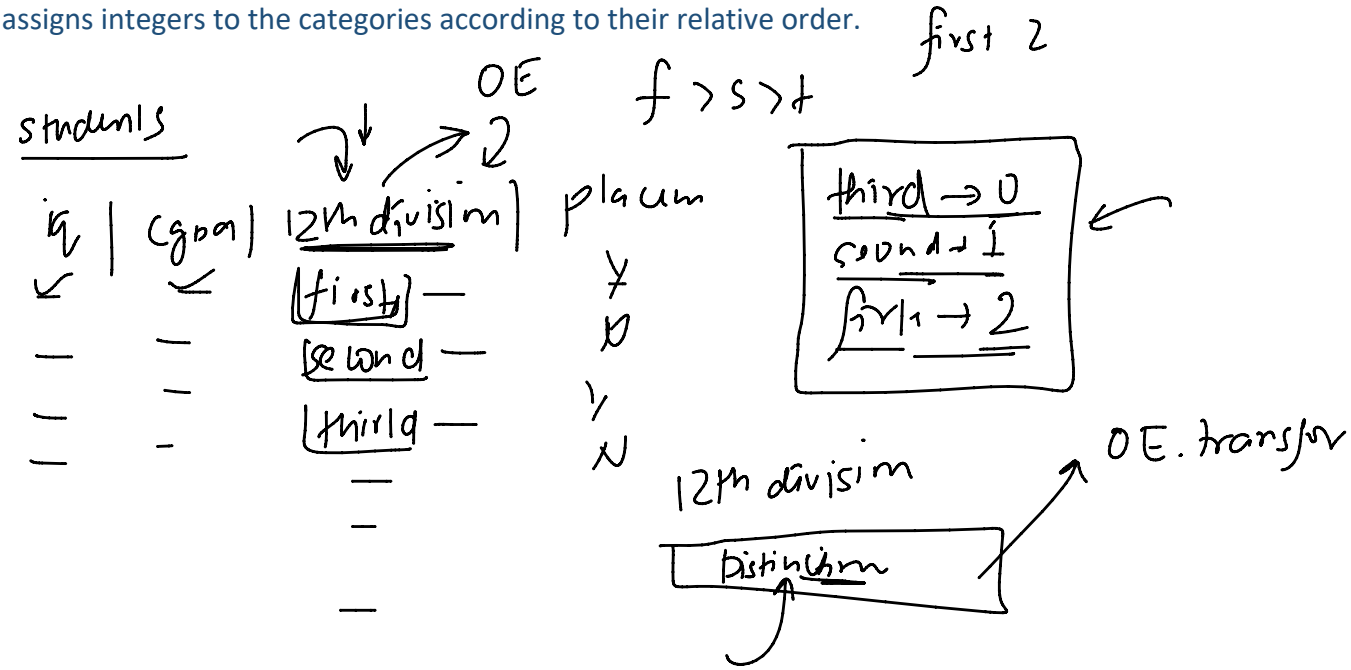
(nominal)



Ordinal Encoding

07 February 2024 10:35

Ordinal encoding is a method for transforming categorical variables that have a natural order or ranking among the categories into numerical values. This technique assigns integers to the categories according to their relative order.



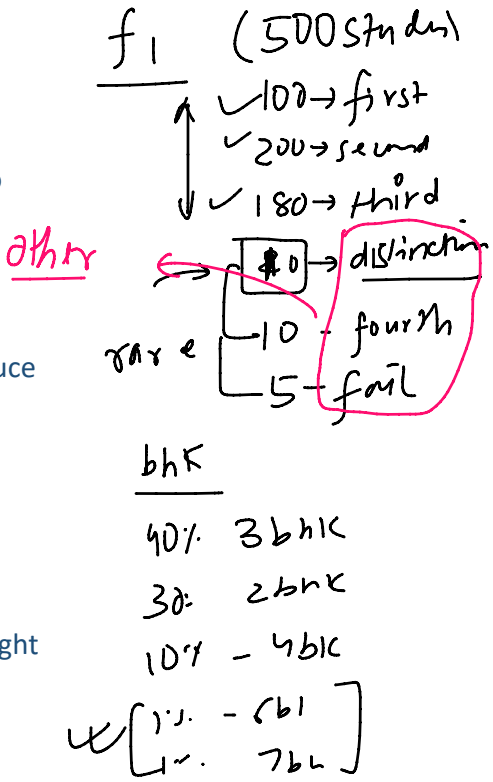
Rare Categories

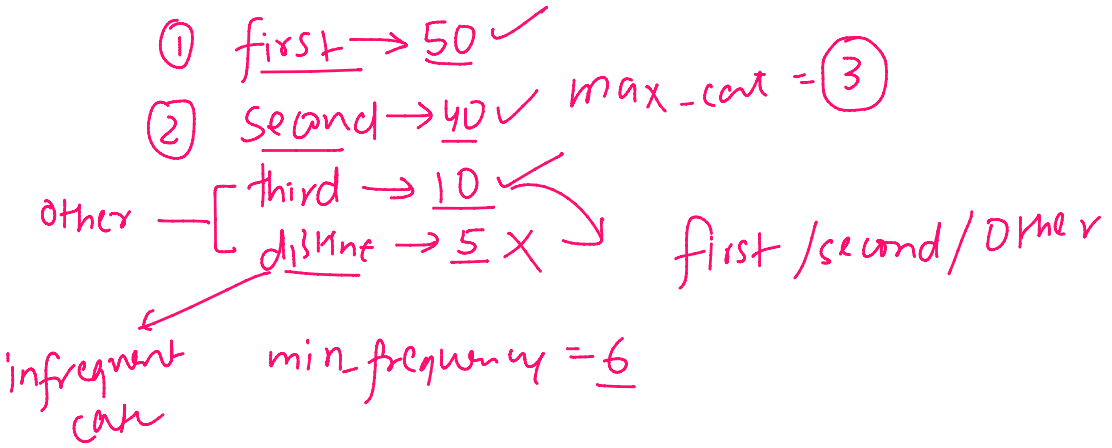
07 February 2024 11:55

Infrequent categories, often referred to as "rare categories," are categories within a categorical variable that appear very seldom in the dataset. These categories are characterized by having a low frequency or count compared to other categories within the same feature.

How to handle:

1. Aggregation: Combining rare categories into a single "Other" category to reduce the feature's cardinality and simplify the model.
2. Encoding with Special Treatment: Using encoding techniques that specifically account for the rarity of categories, such as setting a min_frequency or max_categories threshold in OrdinalEncoder, or employing target encoding where the influence of rare categories is mitigated.
3. Exclusion: In some cases, particularly when a category is extremely rare, it might be justified to exclude those data points from the analysis if it's believed they do not add value or could introduce noise.





Label Encoding

07 February 2024 10:37

→ input

→ target

feature encoding X

f1 f2 ... f3 | Y → encode

Yes → 1

No → 0

target col encodings

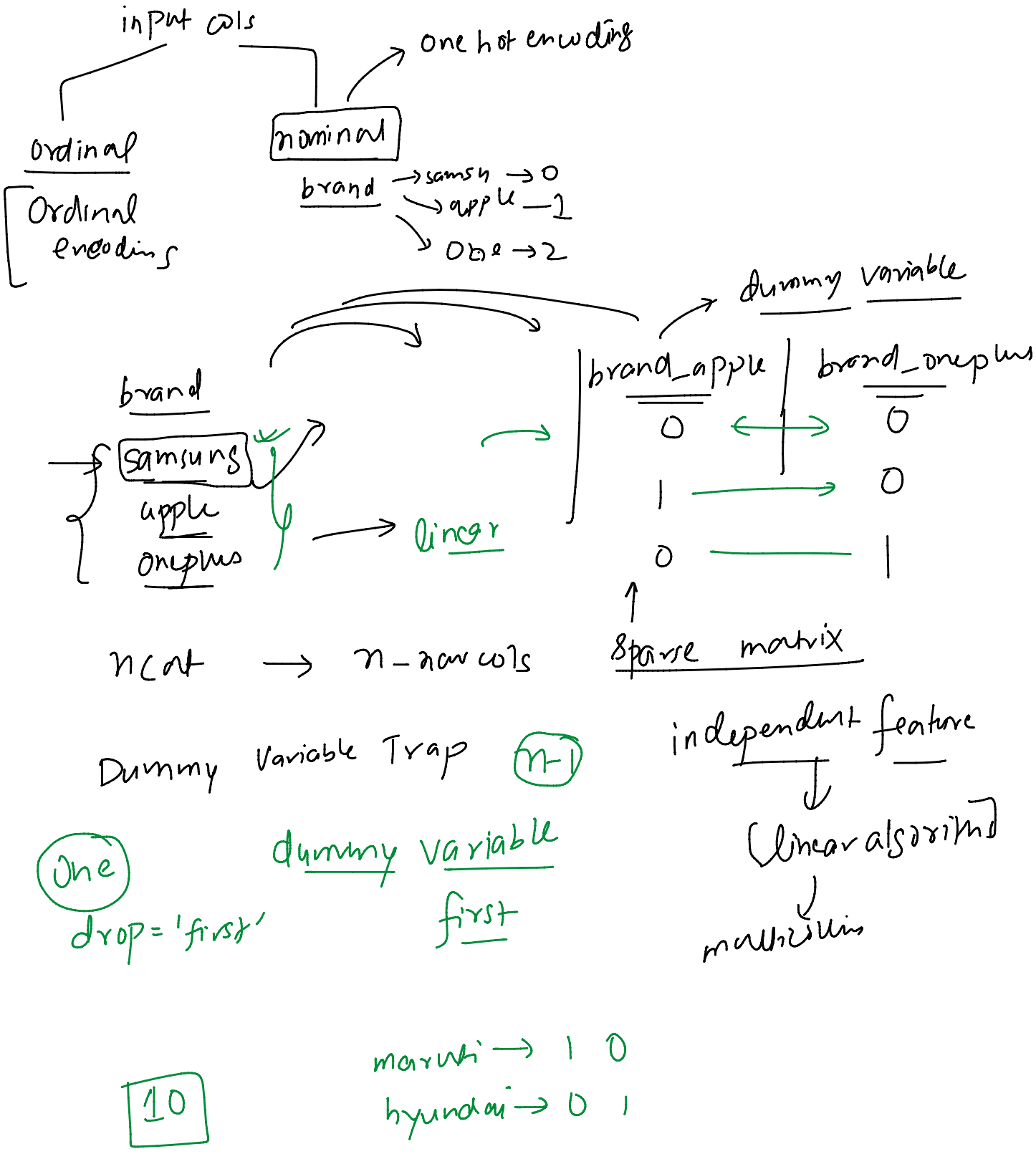
input → ordinal

output → label encoder

One Hot Encoding

07 February 2024 10:37

1. Why cant we use Ordinal Encoder
2. How OHE works?
3. Dummy Variable Trap



brand

→ honda 0 0

LabelBinarizer

07 February 2024 10:38

The LabelBinarizer in scikit-learn is a class used to transform multi-class labels to binary labels (1 vs all). Essentially, it's used for one-hot encoding of labels. This is particularly useful in classification tasks where you need to convert categorical target variables into a format that's suitable for machine learning models, especially those that require a binary or one-hot encoded format for the target variable.

Sepal Length (cm)	Sepal Width (cm)	Petal Length (cm)	Petal Width (cm)	Species
5.1	3.5	1.4	0.2	Setosa
7.0	3.2	4.7	1.4	Versicolor
6.3	3.3	6.0	2.5	Virginica
5.8	2.7	5.1	1.9	Virginica
5.7	2.8	4.1	1.3	Versicolor
5.0	3.6	1.4	0.2	Setosa

→ setosa | versico | virg

→ 1 0 0

0 1 0

0 0 1

multi label classification

Yes

No

Movie Title	Genres
The Matrix	Action, Sci-Fi
Inception	Action, Sci-Fi, Thriller
Pride & Prejudice	Drama, Romance
Toy Story	Animation, Comedy, Family

deep learning

softmax

target

[Label Encoder]

y
Yes
No
Maybe

y_yes	y_no	y_maybe
1	0	0
0	1	0
0	0	1

classification

outcol

classification

Y

Yes → 1

No → 0

outcol

↓

OHE