Once option to allow the mining of multiple occurrences is to simply treat each repeat item as a new occurrence. For example, instead of the set {beer, milk, beer}, we could mine it as {beer, milk, beer\_2}, allowing us to see how often someone bought a second beer after buying a first beer.

Alternatively, we could use a system of weights to perform the calculations, with repeat items having a higher weight value the more repeat items there are. Using the previous example, we could transform {beer, milk, beer} into {beer: 2, milk: 1}

Fraudulent credit card transaction identification is a fairly common example use case for classification tasks, though the small dataset is certainly a complication. I would say the real-world answer to this problem is that we should first work to try to enlarge our dataset, probably by purchasing an anonymized dataset on the market, or just using one from the many Kaggle competitions that cover this. For the class answer, my first instinct is to use some sort of ensemble method, training numerous different models on the dataset and then somehow averaging the results at the end. As the task is fraudulent transaction identification, it is beneficial that we prioritize recall over precision, and we can do this by having a relatively low threshold for flagging a transaction. For example, if our ensemble consists of 10 different models, we could perhaps flag a transaction if two of them classify it as fraudulent.