Assignment 9

1. **Explain One-Hot Encoding and Label Encoding.** **Does the dimensionality of the dataset increase or decrease after encoding, if yes then how?**

Ans:

**One-Hot Encoding:** One-Hot Encoding is popular technique for treating categorical variables. It simply creates additional features based on the number of unique values in the categorical feature. Every unique value in the category will be added as a feature.

One-Hot Encoding is the process of creating dummy variables.

In this encoding technique, each category is represented as a one-hot vector.

**Label Encoding:** Label Encoding is a popular encoding technique for handling categorical variables. In this technique, each label is assigned a unique integer based on alphabetical ordering.

Label Encoding refers to **converting the labels into a numeric form so as to convert them into the machine-readable form**. Machine learning algorithms can then decide in a better way how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.

**Does the dimensionality of the dataset increase or decrease after encoding, if yes then how?**

When we use **one hot encoding**, there is an increase in the dimensionality of a dataset. The reason for the increase in dimensionality is that, for every class in the categorical variables, it forms a different variable.

**Example**: Suppose, there is a variable ‘Colour.’ It has three sub-levels as Yellow, Purple, and Orange. So, one hot encoding ‘colour’ will create three different variables as Colour. Yellow, colour. Purple, and colour Orange.

In **label encoding**, the sub-classes of a certain variable get the value as **0** and **1**. So, we use label encoding only for binary variables.

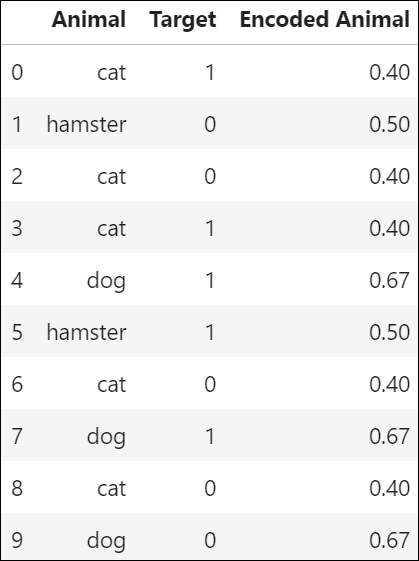
This is the reason that one hot encoding increases the dimensionality of data and label encoding does not.

1. What is Target Encoding and how it is different from one hot encoding?

Ans:

Target encoding is **the process of replacing a categorical value with the mean of the target variable**. Any non-categorical columns are automatically dropped by the target encoder model. Note: You can also use target encoding to convert categorical columns to numeric.

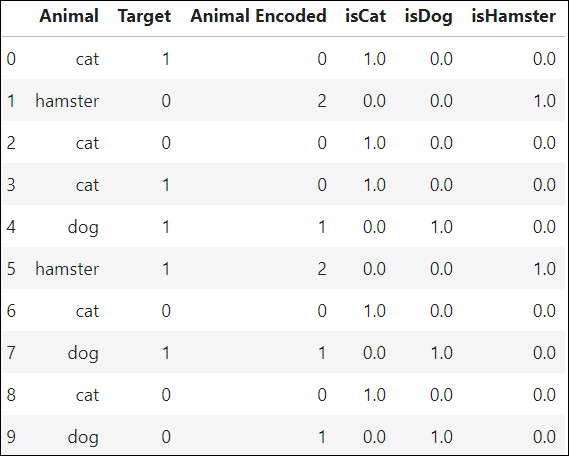
Example :



# how it is different from one hot encoding

# **One-hot Encoding**

One-hot encoding is easier to conceptually understand. This type of encoding simply “produces one feature per category, each binary.” Or for the example above, creating a new feature for cat, dog, and hamster. In the column cat, for example, we show that a cat exists with a 1, and it doesn’t exist with a 0. Let’s look at the same example to make more sense of this:



Hence the difference , because sometimes we have more features in the dataset , in that case we can’t apply one-hot encoding there. Hence we are going with target encoding. In that we find the mean of each categorical variable and then rank the variables on the basis of their means.

1. **If you have a date column in our dataset, then how will you perform Feature Engineering in pandas?**

Ans:

The following steps we have to follow for performing feature engineering in pandas on date column :

1.Get our environment set up.

2.Check the data type of our date column.

3.If it is not in datetime ,convert date columns to datetime by using code pd.to\_datetime(series).

4.Select just year from our column by using code series.dt.year .

5. Select just date from our column by using code series.dt.date.

6. Select just month from our coulumn by using code series.dt.month .

7.Get the week of year, the day of week, and leap year.

8.Get the age from the date of birth.

9. Improve performance by setting date column as the index.

1. How do you perform feature selection with Categorical Data?

Ans:

[**Feature selection**](https://machinelearningmastery.com/an-introduction-to-feature-selection/) is the process of identifying and selecting a subset of input features that are most relevant to the target variable.

Feature selection is often straightforward when working with real-valued data, such as using the Pearson’s correlation coefficient, but can be challenging when working with categorical data.

The two most commonly used feature selection methods for categorical input data when the target variable is also categorical (e.g. classification predictive modelling) are the [chi-squared statistic](https://machinelearningmastery.com/chi-squared-test-for-machine-learning/) and the [mutual information statistic](https://machinelearningmastery.com/information-gain-and-mutual-information).

### Chi-Squared Feature Selection

Pearson’s chi-squared statistical hypothesis test is an example of a test for independence between categorical variables.

You can learn more about this statistical test in the tutorial:

* [A Gentle Introduction to the Chi-Squared Test for Machine Learning](https://machinelearningmastery.com/chi-squared-test-for-machine-learning/)

The results of this test can be used for feature selection, where those features that are independent of the target variable can be removed from the dataset.

The scikit-learn machine library provides an implementation of the chi-squared test in the [chi2() function](https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.chi2.html). This function can be used in a feature selection strategy, such as selecting the top k most relevant features (largest values) via the [SelectKBest class](https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html).

For example, we can define the SelectKBest class to use the chi2() function and select all features, then transform the train and test sets.

**Mutual Information Feature Selection**

Mutual information from the field of information theory is the application of information gain (typically used in the construction of decision trees) to feature selection.

Mutual information is calculated between two variables and measures the reduction in uncertainty for one variable given a known value of the other variable.

You can learn more about mutual information in the following tutorial.

* [What Is Information Gain and Mutual Information for Machine Learning](https://machinelearningmastery.com/information-gain-and-mutual-information)

The scikit-learn machine learning library provides an implementation of mutual information for feature selection via the [mutual\_info\_classif() function](https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.mutual_info_classif.html).

Logistic regression is probably the major alternative (i.e. traditional statistical modelling method) to Random Forests or Decision Trees. That method can also handle categorical data, via dummy variables, but there is a certain amount of data re-coding needed, so it is somewhat more labor intensive.

1. **When would you remove Correlated Variables?**

**Ans:**

In a more general situation, when you have two independent variables that are very highly correlated, **you definitely should remove one of them** because you run into the multicollinearity conundrum and your regression model's regression coefficients related to the two highly correlated variables will be unreliable.