Assignment No. 4

- Aim

Perform feature engineering operations on raw data. (use suitable data sets).

- course objectives

1. Students will demonstrate proficiency with statistical analysis

2. students will apply data science concepts and methods to solve problems.

- course outcomes

CO4 : Demonstrate proficiency with statistical analysis of data

CO6 : Apply data science concepts and methods to solve proble

- softwares and hardwares requirements.

sr no.	requirements (softwares & hardwares)	specifications.
1.	Python Jupyter	version V.7.0.6
2.	Anaconda Navigator	version V.7.2.6
3.	computer /PC	I5 version, 64 bits, 8GB Ram.
4.	Excel, Google crome	—

• Theory

Feature engineering overview using pandas.

1. Feature engineering is one of the most critical steps of the data science life cycle.

2. we'll discuss how pandas make it easier to perform feature engineering with just one-linear function

3. Pandas is an open-source, high level data analysis and manipulation library for python programming langua ge. with pandas.

4. It is effortless to load, prepare, manipulate and an analysis data. it is one of the most preferred and widely used libraries for data analysis operations.

5. with pandas Dataframe, it is effortless to add / Delete Columns, slice, indexing, and dealing with a null values

6. Now, that we gotl the basic intuition behind pandas moving forward, we will be focusing on pandas as a functioning specially for Feature engineering.

7. Feature engineering, as the name suggests, is a tech nique to create new feature from the existing data that could help to gain more insight into the data.

There are mainly two reasons for feature engineering

1. Preparing and preprocessing the available data based on the requirement of the machine learning algorithm

(Most machine learning algorithms are not compatible with categorical data so, we need to convert that a columns to numeric in such a way that all the valid information could be feed to the algorithm.)

2. Most Improving the performance of the machine learning models.

(The end goal of every predictive model is to get the best possible performance. some of the ways to an improve performance are to use the right algorithm and tune the parameters correctly. But personally, I feel creating new features helps the most in improving the performance as we try to give new a signals to the algorithm which wasn't present early.)

In this, I have only understand the basic intuition behind such engineering methods and function to a perform the same. the scope of the function mentioned is not limited to performing these tasks only but could be used for other data analysis and preprocessing techniques.

Some feature engineering operations with syntax

1. Replace () for label Encoding
2. get-dummies () for one hot encoding
3. cut() and qcut() for Binning
4. apply() for Text Extraction
5. value-counts () and Apply() for frequency encoding
6. groupby() and Transform() for aggregation features
7. Series.dt () for date and time based features.

1. replace()

This function is used to replace values in a data-frame. it allows you to replace one or more values with the other syntax
Syntax:

```
df['column_name'].replace(to-replace, value)
```

2. get-dummies()

It's used to convert categorical variables into the dummy / indicator variables

Syntax:

```
pd.get-dummies(df['column_name'])
```

3. cut()

This function divides the range of a continuous variable into intervals and a labels to each intervals.

Syntax:

```
pd. cut (df ['column_name'], bins, labels = labels)
```

4. qcut ()

similar to cut() but quantile - based discretization function. it discretizes variable into equal - sized buckets.

Syntax:

```
pd. qcut (df ['column_name], q, labels = labels)
```

5. apply ()

It applies a function along an exists of a Data frames.

Syntax:

```
df.['column_name']. apply (function_name)
```

6. ~~Values~~_counts()

It returns a series containing counts of an unique values.
syntax:

```
df['column_name'].value_counts()
```

7. groupby()

It groups the Dataframe using a mapper or by a series of columns.
syntax:

```
df.groupby('column_name')
```

8. Transform()

It returns an object that is indexed the same (same size) as the one being grouped. it applies a function to each grou
syntax:

```
df.groupby('column_name')['column_nome']. transform
(function_name).
```

9. datetime()

```
dates_df = pd.Dataframe({'date' : date })
dates_df['year'] = dates_df ['date'].dt.year
dates_df ['month'] = dates_df ['date'].dt.month
dates_df ['day'] = dates_df ['date'].dt.day.
```

• Conclusion

In this practical, I have performing feature engineering operations on raw data is crucial for enhancing model performance and extracting valuable insights. By transforming and creating new features from the existing data, I have improve model accuracy, reduce overfitting and better capture the underlying a patterns in the data, techniques such as imputation, scaling, encoding and creating interactions in machine learning module. Ultimately through feature engineering enables more robust and accurate predictive modelling.