Project Name: Feature-Engineering NYC-Taxi-Fare-Dataset

Table of Contents

Demo

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 120000 entries, 0 to 119999
Data columns (total 8 columns):
 #   Column             Non-Null Count   Dtype              
---  ------             --------------   -----              
 0   pickup_datetime    120000 non-null  datetime64[ns, UTC]
 1   fare_amount        120000 non-null  float64            
 2   fare_class         120000 non-null  int64              
 3   pickup_longitude   120000 non-null  float64            
 4   pickup_latitude    120000 non-null  float64            
 5   dropoff_longitude  120000 non-null  float64            
 6   dropoff_latitude   120000 non-null  float64            
 7   passenger_count    120000 non-null  int64              
dtypes: datetime64[ns, UTC](1), float64(5), int64(2)
memory usage: 7.3 MB
```

	fare_amount	fare_class	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count	Year	Month	Day	Hours	Minutes	mornig
0	6.5	0	-73.992365	40.730521	-73.975499	40.744746	1	2010	4	19	4	17	
1	6.9	0	-73.990078	40.740558	-73.974232	40.744114	1	2010	4	17	11	43	
2	10.1	1	-73.994149	40.751118	-73.960064	40.766235	2	2010	4	17	7	23	
3	8.9	0	-73.990485	40.756422	-73.971205	40.748192	1	2010	4	11	17	25	
4	19.7	1	-73.990976	40.734202	-73.905956	40.743115	1	2010	4	16	22	19	

In [28]:
```
df.drop(["pickup_longitude","pickup_latitude","dropoff_longitude","dropoff_latitude"],axis=1,inplace=True)
```

In [29]:
```
df.head()
```

Out[29]:

	fare_amount	fare_class	passenger_count	Year	Month	Day	Hours	Minutes	mornight	Total distance
0	6.5	0	1	2010	4	19	4	17	0	2.126312
1	6.9	0	1	2010	4	17	11	43	0	1.392307
2	10.1	1	2	2010	4	17	7	23	0	3.326763
3	8.9	0	1	2010	4	11	17	25	1	1.864129
4	19.7	1	1	2010	4	16	22	19	1	7.231321

Overview

This is diving into Feature Engineering Concept.
Feature Engineering is the act of extracting important features from raw data and transforming them into formats that are suitable for machine learning. Feature Engineering is a data preparation process.
This repository contains the code for Feature Engineering using python's various libraries.
It used numpy, pandas, sklearn and datetime libraries.
These libraries help to perform individually one particular functionality.
Numpy is used for working with arrays. It stands for Numerical Python.
Pandas objects rely heavily on Numpy objects.
Sklearn has 100 to 200 models.
Datetime module supplies classes to work with date and time.
The purpose of creating this repository is to gain insights into how to extract and transform data.
These python libraries raised knowledge in discovering these libraries with practical use of it.
It leads to growth in my ML repository.
These above screenshots will help you to understand flow of output.

Motivation

The reason behind building this is, of course all businesses want that their final result should be accurate therefore need to learn Feature Engineering which has a major influence on the performance of machine learning models and even the quality of insights derived during EDA. The models take features as input. A feature is generally a numeric representation of an aspect of real-world phenomena or data. My job as a Data Scientist is to find a clear path to the end goal of insights. Feature engineering helps extract information from raw data, i.e., it has created a lot of features. This means we need to find the main features of the whole lot. This is also known as the Curse of Dimensionality. I sometimes try to relate things of both the world that is IT and Life Philosophy, In a normal life also it is not possible for one person to study and grab all degrees in the world or all the knowledge in the world so people extract degrees as per their interests so similar analogy is here I have to extract relevant features from raw data as per requirement. Through feature engineering, you can isolate key information, highlight patterns, and bring in domain expertise. If I wish to convey my point to buyer than this can be helpful as I can highlight relevant pattern and isolate key information, which is manipulation of data and hence of decision which is a tricky strategy which is what is call for apart from just a normal coder. I believe that, it's not all always about coding but also about convincing or confusing or ponder them to imagination or influence which is also a must which can be achieved through Feature Engineering. Feature engineering, also known as feature creation, is the process of

constructing new features from existing data to train a machine learning model. This step can be more important than the actual model used because a machine learning algorithm only learns from the data, we give it, and creating features that are relevant to a task is absolutely crucial. Statistically, 8 out of 10 people does not like to perform Feature Engineering as it is time-consuming but what fascinates me is, it is art of direction. You know that, master plan can make it to millions too. Hence, I continue to gain knowledge while practicing the same and spread literary wings in tech-heaven.

Technical Aspect

Numpy contains a multi-dimensional array and matrix data structures. It works with the numerical data. Numpy is faster because is densely packed in memory due to its homogeneous type. It also frees the memory faster.
Pandas module mainly works with the tabular data. It contains Data Frame and Series. Pandas is 18 to 20 times slower than Numpy. Pandas is seriously a game changer when it comes to cleaning, transforming, manipulating and analyzing data.
Sklearn is known as scikit learn. It provides many ML libraries and algorithms for it. It provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python.
Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps. Working with dates and times is one of the biggest challenges in programming. Between dealing with time zones, daylight saving time, and different written date formats, it can be tough to keep track of which days and times you're referencing. Fortunately, the built-in Python datetime module can help you manage the complex nature of dates and times. For example, one great example of this irregularity is daylight saving time. In the United States and Canada, clocks are set forward by one hour on the second Sunday in March and set back by one hour on the first Sunday in November. However, this has only been the case since 2007. Prior to 2007, clocks were set forward on the first Sunday in April and set back on the last Sunday in October. Things get even more complicated when you consider time zones. Ideally, time zone boundaries would follow lines of longitude exactly.

Installation

Using intel core i5 9th generation with NVIDIA GFORECE GTX1650.
Windows 10 Environment Used.
Already Installed Anaconda Navigator for Python 3.x
The Code is written in Python 3.8.
If you don't have Python installed then please install Anaconda Navigator from its official site.
If you are using a lower version of Python you can upgrade using the pip package, ensuring you have the latest version of pip, *python* -m pip install --*upgrade pip and press Enter*.

Run/How to Use/Steps

Keep your internet connection on while running or accessing files and throughout too.
Follow this when you want to perform from scratch.

Open Anaconda Prompt, Perform the following steps:

cd <PATH>
pip install numpy
pip install pandas
pip install sklearn
pip install datetime

You can also create requirement.txt file as, pip freeze > requirements.txt

run files.

Follow this when you want to just perform on local machine.
Download ZIP File.
Right-Click on ZIP file in download section and select Extract file option, which will unzip file.
Move unzip folder to desired folder/location be it D drive or desktop etc.
Open Anaconda Prompt, write cd <PATH> and press Enter.

eg: cd C:\Users\Monica\Desktop\Projects\Python Projects 1\15)Feature_Engineering+Selection+Importance\ Feature_Engineering_3Different_Datasets\FE_NYCTaxiFareDataset

In Anconda Prompt, pip install -r requirements.txt to install all packages.
Open in Jupyter Notebook, <filename>.ipynb
That is,

Open in Jupyter Notebook, 1)Feature_Engineering_for_NYCTaxiFareDataset.ipynb
This takes taxifare.csv file as input dataset.
Please be careful with spellings or numbers while typing filename and easier is just copy filename and then run it to avoid any silly errors.
Note: cd <PATH>
[Go to Folder where file is. Select the path from top and right-click and select copy option and paste it next to cd one space <path> and press enter, then you can access all files of that folder] [cd means change directory]

Directory Tree/Structure of Project

Folder: 15)Feature_Engineering+Selection+Importance > Feature_Engineering_3Different_Datasets > FE_NYCTaxiFareDataset

1)Feature_Engineering_for_NYCTaxiFareDataset.ipynb

To Do/Future Scope

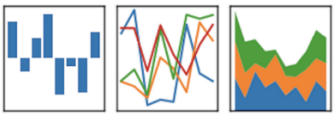
Can do visualization after it.

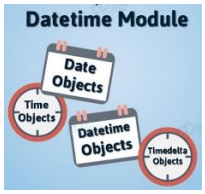
Technologies Used/System Requirements/Tech Stack

NumPy

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$





Credits

Krish Naik Channel