****

**(B.TECH) Trimester-XI AY 2020-21**

**PPL Lab Assignment No. 02**

**Faculty: Prof. Jayshree Aher**

**Problem Statement:** To study Feature Engineering and implement the dimensionality reduction techniques (PCA & TSNE)

**Name: Vishwa Mohan Singh**

**ERP No.: 1032170273**

**Panel: AML2**

**Date: 08/12/2020**

**Objectives:**

1. To understand Feature Engineering and learn feature selection techniques.
2. To understand the concept of dimensionality reduction.

**Theory:**

* Feature Engineering

Feature Engineering is the process of transforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data.

Some Feature Engineering Method:

* Binning
* Log Transform
* Scaling
* Feature Selection
* Feature Selection Techniques

Feature selection is the process where you automatically or manually select those features which contribute most to your prediction variable or output in which you are interested in. This method can deal with:

* Multicollinearity
* High Data Dimension
* High Training Time

Feature Selection Methods:

* Forward Selection
* Backward Elimination
* Dimensionality reduction (PCA, LDA and t-SNE)
  + Filter Method
    - ANOVA: Analysis of Variance is a method used to analyse the difference among the group means in a sample.
    - Chi\_square: This is defined as where Oi is the observation and Ei is the expected value. This value of chi can be used to derive the p-value that gives us the probability of independence. If a p value is high (>0.05), we can say that the attribute is not statistically significant to the target variable.
    - Pearson’s correlation: This is defined as . This value is the measure of the strength of a linear association between two variable, where r = 1 means a perfect positive correlation and the value r = -1 means a perfect negative correlation.
  + Wrapper Method
    - RFE: It is a wrapper-type feature selection algorithm. This means that a different machine learning algorithm is given and used in the core of the method, is wrapped by RFE, and used to help select features.
  + Intrinsic Method
    - DT: A decision tree is a method which can be traversed based on the attribute values and can give an intrinsic value at the leaf nodes.
* PCA

Principle Component Analysis is an algorithm that uses the eigen values derived from the correlation matrix in order to reduce the dimension of the dataset. The reduced features are representative of the dataset but does not hold any meaning on its own.

* t-SNE

t-distributed stochastic neighbor embedding is a machine learning algorithm that employees Stochastic Neighbor Embedding to reduce the number of attribute by projecting them on a low dimension space.

**Operations to be performed on dataset:**

**Steps in Preprocessing of Data**

**PCA** Steps for dimensionality reduction:

1. Read the .csv file of dataset
2. Display few observations
3. Create the independent and dependent variables
4. Standardization of data
5. Find Covariance matrix
6. Find eigen values and eigen vectors
7. Find principal components
8. Reducing dimensions of dataset and plot the graph

**t-SNE** Steps for dimensionality reduction:

1. Use TSNE function from sklearn
2. Implement the technique on sample dataset with different values of perplexity and iterations.
3. Plot atleast two dimensions

**Program code:**

1. PCA

import pandas as pd

import numpy as np

from sklearn.impute import SimpleImputer

from sklearn.preprocessing import LabelEncoder,OneHotEncoder,StandardScaler

from sklearn.compose import ColumnTransformer

from sklearn.model\_selection import train\_test\_split

import matplotlib.pyplot as plt

from scipy.linalg import eigh

## Importing the data

dataset = pd.read\_csv("train.csv")

print(dataset.head())

print()

print(dataset.info())

##Array Transformation

arr = dataset.iloc[:,[2,5,6,7,9,1]]

## Handeling NaN

impu = SimpleImputer(missing\_values=np.nan, strategy = "mean")

arr = arr.values

arr[:,[0,1,2,3,4]] = impu.fit\_transform(arr[:,[0,1,2,3,4]])

##X-Y Split

X = arr[:,:-1]

Y = arr[:,-1]

#Scaling

scaler = StandardScaler()

X = scaler.fit\_transform(X)

#Getting Covariance matrix

cov = np.matmul(X.T, X)

print("\nShape of variance matrix = ", cov.shape)

#Finding Eigen Values

val,eigen = eigh(cov,eigvals=[3,4])

print(val)

print(eigen)

#Taking transpose

eigen = eigen.T

print("\nShape of eigen vector ", eigen.shape)

#Reducing Dimensions

new\_coordinates = np.matmul(eigen, X.T)

new\_coordinates = new\_coordinates.T

print("\nReduced dimensions shape ",new\_coordinates.shape)

print("\nReduced data ", new\_coordinates)

#Ploting a scatter plot

plt.scatter(new\_coordinates[:,0],new\_coordinates[:,1])

plt.show()

1. t-SNE

import pandas as pd

import numpy as np

from sklearn.impute import SimpleImputer

from sklearn.preprocessing import LabelEncoder,OneHotEncoder,StandardScaler

from sklearn.compose import ColumnTransformer

from sklearn.model\_selection import train\_test\_split

import matplotlib.pyplot as plt

from sklearn.manifold import TSNE

## Importing the data

dataset = pd.read\_csv("train.csv")

print(dataset.head())

print()

print(dataset.info())

## Array Transformation

arr = dataset.iloc[:,[2,5,6,7,9,1]]

## Handeling NaN

impu = SimpleImputer(missing\_values=np.nan, strategy = "mean")

arr = arr.values

arr[:,[0,1,2,3,4]] = impu.fit\_transform(arr[:,[0,1,2,3,4]])

##X-Y Split

X = arr[:,:-1]

Y = arr[:,-1]

#Scaling

scaler = StandardScaler()

X = scaler.fit\_transform(X)

print("Data Shape ",X.shape)

#Creating t-SNE object

tsne = TSNE(n\_components = 2, random\_state = 0)

X = tsne.fit\_transform(X)

print("\nReduced Data Shape ",X.shape)

print("\nReduced Data ",X)

#Plotting

plt.scatter(X[:,0],X[:,1])

plt.show()

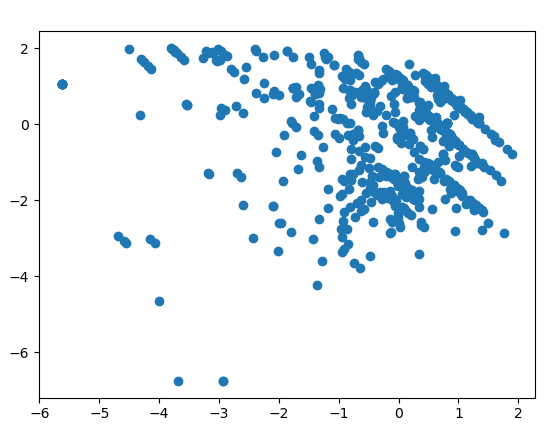
**Dataset used: (source link & description)**

**Titanic:** A Kaggle competition dataset that is used for a machine learning competition aimed at predicting who might have survived the sinking. It includes passenger information like class, embarkment port, fare, cabin etc.

**Link:** https://www.kaggle.com/c/titanic/data

**Output:**

**PCA:**



Scatter plot for PCA

PassengerId Survived Pclass ... Fare Cabin Embarked

0 1 0 3 ... 7.2500 NaN S

1 2 1 1 ... 71.2833 C85 C

2 3 1 3 ... 7.9250 NaN S

3 4 1 1 ... 53.1000 C123 S

4 5 0 3 ... 8.0500 NaN S

[5 rows x 12 columns]

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 891 entries, 0 to 890

Data columns (total 12 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 PassengerId 891 non-null int64

1 Survived 891 non-null int64

2 Pclass 891 non-null int64

3 Name 891 non-null object

4 Sex 891 non-null object

5 Age 714 non-null float64

6 SibSp 891 non-null int64

7 Parch 891 non-null int64

8 Ticket 891 non-null object

9 Fare 891 non-null float64

10 Cabin 204 non-null object

11 Embarked 889 non-null object

dtypes: float64(2), int64(5), object(5)

memory usage: 83.7+ KB

None

Shape of variance matrix = (5, 5)

[1452.24566121 1506.02210497]

[[ 0.05454005 0.68029285]

[ 0.32001975 -0.45274522]

[-0.61235522 0.12605198]

[-0.6193362 0.03563436]

[-0.36885838 -0.56130607]]

Shape of eigen vector (2, 5)

Reduced dimensions shape (891, 2)

Reduced data [[ 0.06919074 1.15080251]

[-0.14288499 -1.75860564]

[ 0.71829878 0.88943917]

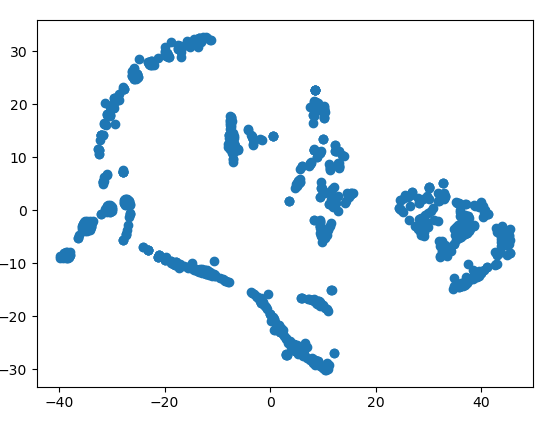
...

[-1.39908707 0.78793795]

[ 0.42381037 -0.9883165 ]

[ 0.86735995 0.68237268]]

**TSNE:**



Scatter plot for t-SNE

PassengerId Survived Pclass ... Fare Cabin Embarked

0 1 0 3 ... 7.2500 NaN S

1 2 1 1 ... 71.2833 C85 C

2 3 1 3 ... 7.9250 NaN S

3 4 1 1 ... 53.1000 C123 S

4 5 0 3 ... 8.0500 NaN S

[5 rows x 12 columns]

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 891 entries, 0 to 890

Data columns (total 12 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 PassengerId 891 non-null int64

1 Survived 891 non-null int64

2 Pclass 891 non-null int64

3 Name 891 non-null object

4 Sex 891 non-null object

5 Age 714 non-null float64

6 SibSp 891 non-null int64

7 Parch 891 non-null int64

8 Ticket 891 non-null object

9 Fare 891 non-null float64

10 Cabin 204 non-null object

11 Embarked 889 non-null object

dtypes: float64(2), int64(5), object(5)

memory usage: 83.7+ KB

None

Data Shape (891, 5)

Reduced Data Shape (891, 2)

Reduced Data [[ -7.528336 15.4489765]

[ 35.15682 -4.579014 ]

[-30.430494 17.936888 ]

...

[ 9.6989765 4.2026815]

[ 45.37128 -4.621011 ]

[-21.262516 -8.835775 ]]

**FAQs:**

1. List any two univariate feature selection techniques.

* Following methods are used to perform forward selection or backward elimination:
  + Regression F-Score
  + ANOVA
  + Chi-Squared

1. What is ‘curse of dimensionality’ in Machine Learning?

* The “Curse of Dimensionality” basically implies that the number of errors increases with the increase in the number of columns. This happens because it is harder to design machine learning algorithms for a high dimension space and often have a running time exponential in the dimension.

1. Comment on the statement: ‘Wrapper methods need more computational power’.

* In wrapper methods, the feature selection process is based on a specific machine learning algorithm that we are trying to fit on a given dataset. Using this, we can calculate values like R-Squared and adjusted R-Squared which can then be used in forward selection or backward elimination. The computational power required here is more since after every step, when we either add or remove a factor, we need to create and train a new model to get the adj R-Squared and p-values.

1. Compare PCA and t-SNE.

|  |  |
| --- | --- |
| PCA | t-SNE |
| It is a linear Dimensionality reduction technique. | It is a non-linear Dimensionality reduction technique. |
| It tries to preserve the global structure of the data. | It tries to preserve the local structure (cluster) of data. |
| PCA is a deterministic algorithm. | It is a non-deterministic or randomised algorithm. |
| It works by rotating the vectors for preserving variance. | It works by minimising the distance between the point in a guassian. |

1. Give the importance of perplexity in t-SNE technique.

* Perplexity tells how to balance attention between local and global aspects of your data and can have large effects on the resulting plot. The parament is a guess about the number of close neighbors each point has.

1. State the importance of heatmap.

* The heat map employees the visual tool of color to highlight important aspects in a plot. The primary purpose of heat map is to better visualize the volume of locations/events within a dataset and assist in directing viewers towards areas on data visualizations that matter most.

**Conclusion:**

Feature Engineering was studied and implemented the two dimensionality reduction techniques PCA & TSNE.