Dimension Reduction – PCA & SVD

Instructions:
Please share your answers wherever applicable in line with the word document. Submit code separately wherever applicable.

Please ensure you update all the details:
Name: ULLI VENKATA SAI KUMAR Batch ID: 04072024HYD10AM
Topic: Dimension Reduction – PCA & SVD

Problem Statements:

The average retention rate in the insurance industry is 84%, with the top-performing agencies in the 93% - 95% range. Retaining customers is all about the long-term relationship you build. Offering a discount on the client's current policy will ensure he/she buys a new product or renews the current policy. Studying clients' purchasing behaviour to determine which products they're most likely to buy is essential.

The insurance company wants to analyze their customer's behaviour to strategies offers to increase customer loyalty.

CRISP-ML(Q) process model describes six phases:

1. Business and Data Understanding
2. Data Preparation
3. Model Building
4. Model Evaluation
5. Deployment
6. Monitoring and Maintenance

Objective: Maximize the Sales

Constraints: Minimize the Customer Retention

Success Criteria:

Business Success Criteria: Increase the Sales by 10% to 12% by targeting cross-selling opportunities on current customers.

ML Success Criteria: NA

Economic Success Criteria: The insurance company will see an increase in revenues by at least 8%

Data: Refer to the Autoinsurance.csv dataset.

Customer	State	Customer	Response	Coverage	Education	Effective To Date	Employme	Gender	Income	Location C	Marital Sta	Monthly Pi	Months Sir	Months Sir	Number o	Number o	Policy Typ	Policy	Renew Off	Sales Char	Total Clair	Vehicle Cl	Vehicle Siz
BU79786	Washingto	2763.519	No	Basic	Bachelor	2/24/2011	Employed	F	56274	Suburban	Married	69	32	5	0	1	Corporate	Corporate	Offer1	Agent	384.8111	Two-Door	Medsize
QZ44356	Arizona	6979.536	No	Extended	Bachelor	1/31/2011	Unemploy	F	0	Suburban	Single	94	13	42	0	8	Personal A	Personal L	Offer3	Agent	1131.465	Four-Door	Medsize
AI49188	Nevada	12887.43	No	Premium	Bachelor	2/19/2011	Employed	F	48767	Suburban	Married	108	18	38	0	2	Personal A	Personal L	Offer1	Agent	566.4722	Two-Door	Medsize
WW63253	California	7645.862	No	Basic	Bachelor	1/20/2011	Unemploy	M	0	Suburban	Married	106	18	65	0	7	Corporate	Corporate	Offer1	Call Cente	529.8813	SUV	Medsize
HB64268	Washingto	281.3.693	No	Basic	Bachelor	3/2/2011	Employed	M	43836	Rural	Single	73	12	44	0	1	Personal A	Personal L	Offer1	Agent	138.1309	Four-Door	Medsize
OC83172	Oregon	8256.298	Yes	Basic	Bachelor	1/25/2011	Employed	F	62902	Rural	Married	69	14	94	0	2	Personal A	Personal L	Offer2	Web	159.383	Two-Door	Medsize
XZ87318	Oregon	5380.899	Yes	Basic	College	2/24/2011	Employed	F	55350	Suburban	Married	67	0	13	0	9	Corporate	Corporate	Offer1	Agent	321.6	Four-Door	Medsize
CF85061	Arizona	7216.1	No	Premium	Master	1/18/2011	Unemploy	M	0	Urban	Single	101	0	68	0	4	Corporate	Corporate	Offer1	Agent	363.0297	Four-Door	Medsize
DY87989	Oregon	24127.5	Yes	Basic	Bachelor	1/26/2011	Medical Le	M	14072	Suburban	Divorced	71	13	3	0	2	Corporate	Corporate	Offer1	Agent	511.2	Four-Door	Medsize
BQ94931	Oregon	7388.178	No	Extended	College	2/17/2011	Employed	F	28812	Urban	Married	93	17	7	0	8	Special Au	Special L2	Offer2	Branch	425.5278	Four-Door	Medsize
SX51350	California	4738.992	No	Basic	College	2/21/2011	Unemploy	M	0	Suburban	Single	67	23	5	0	3	Personal A	Personal L	Offer1	Agent	482.4	Four-Door	Small
VQ65197	California	8197.197	No	Basic	College	6/1/2011	Unemploy	F	0	Suburban	Married	110	27	87	0	3	Personal A	Personal L	Offer2	Agent	528	SUV	Medsize
DP39365	California	8798.797	No	Premium	Master	6/2/2011	Employed	M	77026	Urban	Married	110	9	82	2	3	Corporate	Corporate	Offer2	Agent	472.0297	Four-Door	Medsize
SJ95423	Arizona	8819.019	Yes	Basic	High Schoo	10/1/2011	Employed	M	99845	Suburban	Married	110	23	25	1	8	Corporate	Corporate	Offer2	Branch	528	SUV	Medsize

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
df = pd.read_csv('AutoInsurance.csv')
print(df.info())
print(df.head())
print(df.describe())
df['Income'].hist()
plt.title("Income")
plt.show()
print(df.isnull().sum())
df.fillna(df.mean(), inplace=True)
df.fillna(df.mode().iloc[0], inplace=True)
df['Single_Product_Customer'] = df['Product_Count'].apply(lambda x: 1 if x == 1 else 0)
X = df.drop('Target', axis=1)
y = df['Target']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
clf = RandomForestClassifier()
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
```

```
print("Accuracy:", accuracy_score(y_test, y_pred))

cross_sell_customers = df[df['Target'] == 1]

estimated_revenue_increase = cross_sell_customers['Sales'].sum() * 0.1

print("Estimated Revenue Increase:", estimated_revenue_increase)
```

Questions to Trigger Your thoughts:

Q1. Which libraries are used in PCA to find the optimal number of PCA components?

You can use the following libraries in Python to find the optimal number of PCA components:

- **Scikit-Learn** (sklearn.decomposition.PCA) – Use the explained variance ratio to find the number of components.

```
from sklearn.decomposition import PCA

pca = PCA().fit(X)

explained_variance_ratio = pca.explained_variance_ratio_
```

Q2. Principal Component Analysis (PCA) is a _____ technique in Data Mining?

- PCA is a **dimensionality reduction** technique in Data Mining.

Q3. What is the importance of using PCA before the clustering?

PCA is important before clustering because:

- It reduces **dimensionality**, making the clustering algorithm more efficient.
- It removes **noise** and irrelevant features, helping improve cluster formation.
- It **projects data** into lower dimensions, which may make clusters more separable.

Q4. Can we perform PCA on categorical features?

- No, PCA is not suitable for categorical features. PCA relies on the covariance matrix, which requires numerical input. Categorical features need to be one-hot encoded or transformed using other techniques like **Factor Analysis** or **Multiple Correspondence Analysis (MCA)**.

Q5. Why is it important to create pipelines?

Pipelines in machine learning are important because they:

- **Ensure reproducibility** by chaining together multiple steps (preprocessing, feature selection, modeling).
- **Prevent data leakage** by applying transformations only on training data.
- Simplify code and make it more **readable** and **manageable**.

Q6. Which libraries we can use to save or dump pipelines?

You can use:

- **joblib** from Scikit-Learn to save and load models and pipelines.

from joblib import dump, load

dump(pipeline, 'pipeline.pkl')

- **pickle** can also be used to serialize pipelines.

Q7. Why it is important to standardize the data in PCA?

- Standardizing data (scaling) is important in PCA because:
- PCA is sensitive to the scale of the data since it maximizes variance. If features are on different scales, PCA will be biased toward features with larger magnitudes.
- Standardization ensures all features contribute equally to the variance.

Q8. How can you obtain the principal components and the eigenvalues from Scikit-Learn PCA?

You can obtain the principal components and eigenvalues as follows:

- Principal Components: pca.components_
- **Eigenvalues:** The eigenvalues are the variance explained by each component, which can be accessed using: pca.explained_variance_

Q9. What is sklearn.pipeline extension used for?

- The sklearn.pipeline module is used to create **sequential workflows** where steps such as data preprocessing, feature transformation, and model fitting are performed in sequence. It ensures that all transformations are applied consistently and that there is no data leakage.

Q10. Why do we use filterwarnings function? What library does it belong to and what are the uses of the library

- filterwarnings belongs to the **warnings** library.
- It is used to suppress or filter out unwanted warning messages in Python, allowing you to focus on important parts of the output during development or testing.
  import warnings
  warnings.filterwarnings("ignore")

Q11. What is the extension for the sklearn library to import TruncatedSVD?

- You can import TruncatedSVD from **sklearn.decomposition**:

  from sklearn.decomposition import TruncatedSVD

Q12. How to read only the first 30 data rows?

- df.head(30)

Q13. What are the common functions used from the joblib library? Why do we use this library?

Common functions in joblib:

- **dump**: Save a model or pipeline to a file.
- **load**: Load a saved model or pipeline.

We use joblib to efficiently save and load large models or objects.

Q14. How to drop columns in location [5] ?

- df.drop(df.columns[5], axis=1, inplace=True)

Q15. How to set the timeframe as an index?

- df.set_index('Date', inplace=True)

Q16. How to check what imputation is better for replacing nan/infinity values?

- You can check different imputations by using **cross-validation** or comparing metrics (e.g., MSE, accuracy) after applying various imputation techniques such as mean, median, or KNN
  imputati from sklearn.impute import SimpleImputer

```
imputer = SimpleImputer(strategy='mean')
imputed_data = imputer.fit_transform(df)on.
```

Q17. What does figsize(x, y) define in plotting?

- figsize(x, y) defines the **dimensions** of a plot in inches, where x is the width and y is the height.

Q18. Can we define the type of plot inside a plot() function?

- Yes, you can specify the type of plot using the kind parameter in pandas' plot() function:
- df.plot(kind='bar')

Q19. How is SVD different from PCA?

- **PCA**: Uses the covariance matrix, works on centered data, and is typically used for dimensionality reduction.
- **SVD**: Factorizes the data matrix directly (without centering) and can be used for both dimensionality reduction and matrix decomposition.

Q20. What are n_components in SVD?

- n_components in SVD refers to the **number of singular values** (or components) to keep, i.e., the number of reduced dimensions.

Q21. What does the fit function do? What does the transform function do?

- **fit()**: Learns the model's parameters from the training data.
- **transform()**: Applies the learned transformation to the data.

In PCA, fit() calculates the principal components, while transform() projects the data onto these components.