**Introduction:**

Data mining is the process of discovering patterns in large data sets involving methods of machine learning, statistics, and database systems.

Regression using PCA: It is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of linearly uncorrelated variables called principal components.

**Datasets used:**

1) Wine Dataset: These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines.

Source: https://archive.ics.uci.edu/ml/datasets/wine

**Analysis and conclusions:**

**Part 1:**

Wine Dataset:

1)Let us import the libraries and dataset required

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

# Importing the dataset

dataset = pd.read\_csv('Wine.csv')

X = dataset.iloc[:, 0:13].values

y = dataset.iloc[:, 13].values

A screenshot of a social media post

Description generated with very high confidence

We can see that the wine dataset contains features which describes the contents of the wine. The last column describes the type of wine which is found using clustering. We need to verify the results using logistic regression which is a supervised learning process. We will apply PCA to reduce the correlations within the variables. Let us see the accuracy of our predictions.

2) Let us divide the dataset into test and train sets. We keep the ratio of 80 and 20 percent between training and testing dataset respectively.

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.2, random\_state = 0)

3) We then perform feature scaling on the data:

We normalize the data to standardize the range of independent variable and features.

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X\_train = sc.fit\_transform(X\_train)

X\_test = sc.transform(X\_test)

A screenshot of a cell phone

Description generated with very high confidence

4) Applying PCA

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components.

from sklearn.decomposition import PCA

pca = PCA(n\_components = 2)

X\_train = pca.fit\_transform(X\_train)

X\_test = pca.transform(X\_test)

explained\_variance = pca.explained\_variance\_ratio\_

5) Fitting logistic regression to training set.

Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). [4]

from sklearn.linear\_model import LogisticRegression

classifier = LogisticRegression(random\_state = 0)

classifier.fit(X\_train, y\_train)

6) Predicting the test set results.

y\_pred = classifier.predict(X\_test)

# Making the Confusion Matrix

from sklearn.metrics import confusion\_matrix

cm = confusion\_matrix(y\_test, y\_pred)

A screenshot of a cell phone

Description generated with very high confidence

We can see the actual wine type values on the left and the predictions on the right.

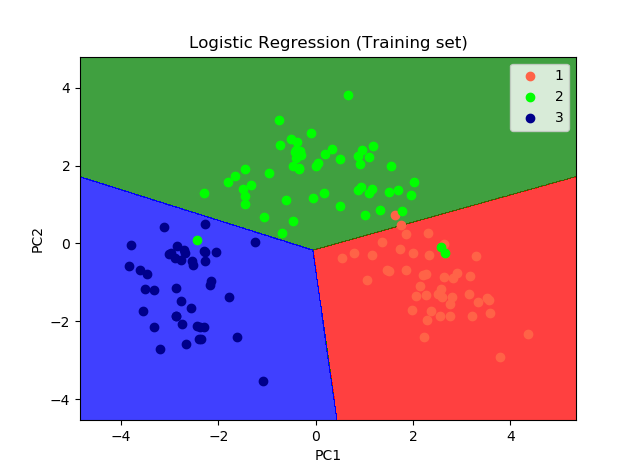
Let us see the accuracy using a confusion matrix.

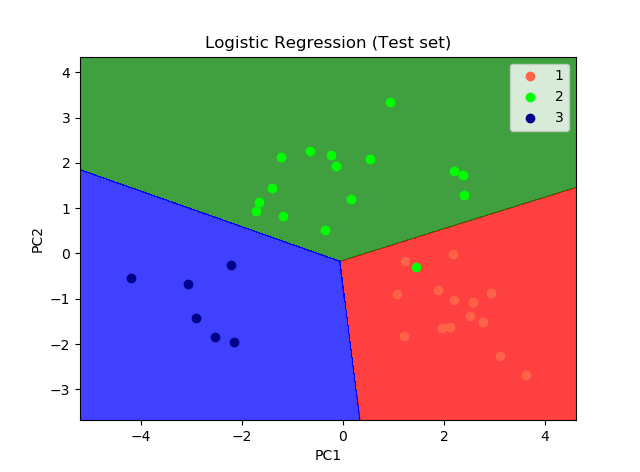
A picture containing object

Description generated with high confidence

We can see there was only one wrong prediction where the wine type was type 0 but was prediction as type 1.

7) Visualizing the results:





Hence, we have successfully implemented PCA and logistic regression in Python. We can see there are a very few wrong predictions and overall it has perfectly fitted the model.