भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Identifying fraudulent taxpayers using variational autoencoders

Fraud Analytics Assignment 2

Jatin Sharma
CS17BTECH11020

Anjani Kumar
CS17BTECH11002

Dhananjay Raut
CS17BTECH11014

Vijay Tadikamalla
CS17BTECH11040

(Teamsize of 4 with special permission from sir)

Objective

The objective of this assignment is to identify fraudulent taxpayers by training variational autoencoders on the given dataset. The idea being the algorithm will be able to separate genuine taxpayers from fraudulent ones into separate clusters. To create effective clusters we want to learn good representations of the data points which will be used to separate out fraudulent taxpayers with techniques similar to anomaly detection. We use variational autoencoders

---

Introduction

Tax fraud occurs when an individual or business entity willfully and intentionally falsifies information on a tax return to limit the amount of tax liability. Tax fraud essentially entails cheating on a tax return in an attempt to avoid paying the entire tax obligation. Given the data with various features observed for taxpayers identification for fraudulent taxpayers comes down to an anomaly detection problem. Fraudulent taxpayers will be separate from genuine users and so they are expected to cluster separately by various clustering algorithms. They work best when given good small dimensional representations of data points. We know Neural networks learn non-linear representations of input so we force the network to learn smaller dimensional representations of the input in variational autoencoders. The model is trained using its ability to reconstruct the input from smaller dimensional representations at intermediate nodes.

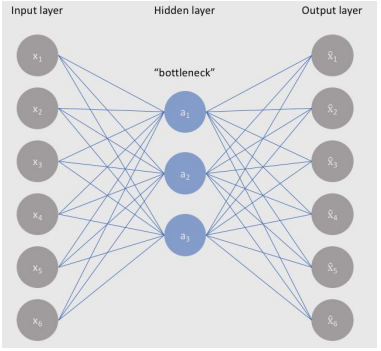

Figure : bottleneck layer to learn compressed representation.

Dataset

The dataset we got for the assignment consists Of 1163 data points each with 9 real-valued features P1-9. We did not get any feature descriptions and also there was no target column given. It is clearly an unsupervised learning task. We aim to cluster data points in the dataset to possibly identify fraudulent taxpayers which most like be quite separate From other genuine users.

	P1	P2	P3	P4	P5	P6	P7	P8	P9
0	0.595378	-0.531958	0.679654	-0.126799	0.432046	0.988092	-0.029813	0.768742	-0.054167
1	0.982237	0.991481	0.337646	0.228144	0.920032	0.999985	-0.032259	2.161651	-0.054350
2	0.996162	0.893987	0.767413	0.606840	0.970808	0.882602	-0.032267	0.369607	-0.054157
3	0.999928	0.922748	-0.444438	-0.371287	0.528038	-0.221645	-0.032692	-1.065439	0.381914
4	0.985838	0.937512	0.699592	0.585263	0.838804	0.999602	-0.033713	2.089720	-0.054379

Figure 1: Sample data points from the dataset

We did some data analysis on the provided Dataset and its features.

Dataset statistics

Number of variables	9
Number of observations	1163
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	81.9 KiB
Average record size in memory	72.1 B

Figure 2: Various statistics for our dataset

Figure 3 shows the interaction between two sample features in our dataset which clearly shows all the features are not independent. can be effectively represented in smaller dims.

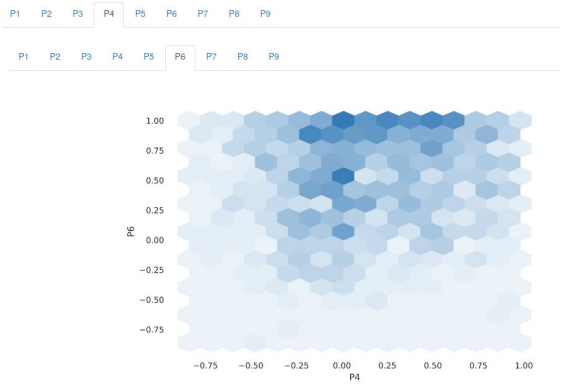

Figure 3: Interaction between P4 and P6.

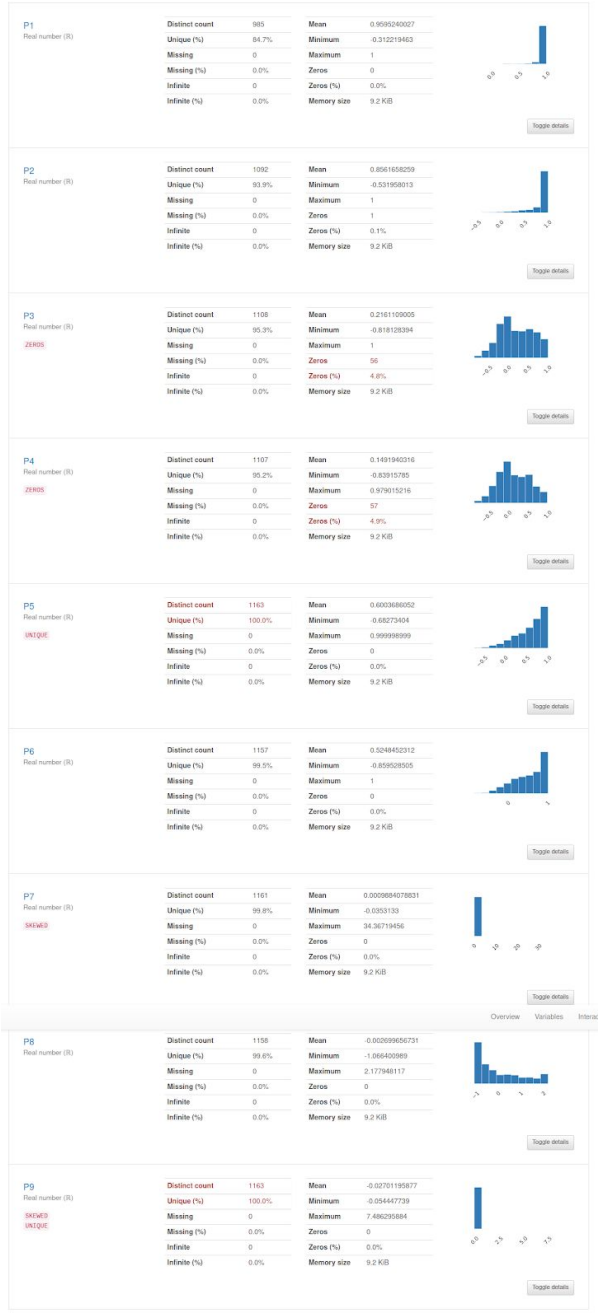

Figure 4: Distribution of all the features of our dataset.

Figure 4 shows various statistics for all the features and their individual distributions. We see some features that are highly skewed. which indicate the need for feature scaling and normalization in our dataset.

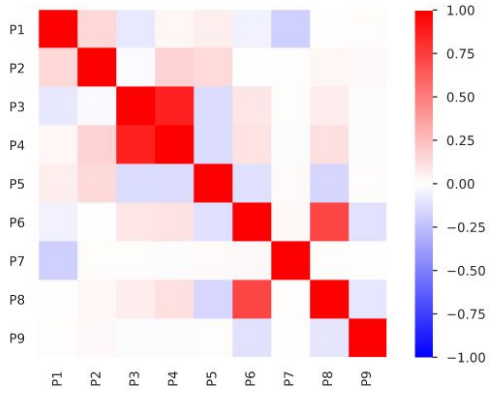

Figure 5: Pearson's r correlation of features.

Figure 5 show Pearson's r correlation which again indicates the presence of lower-dimensional Representation of our dataset.

Now to visualize our data we have to project it in smaller dimensions from current 9 dimensions. So we used the t-SNE technique to get the good visualizable 3-dimensional representation of our dataset. Figure 6 shows The same visualization using Plotly library. This is actually an interactive 3-dimensional plot in the attached notebook.

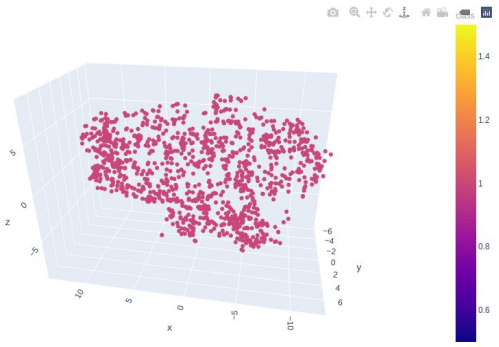

Figure 5: Data points visualized in 3D.

Figure 6 shows why k-means is not a good algorithm in our case. It has clustered our data In nearly equal parts which are not that helpful. In our task to find rare fraudulent taxpayers. And so we use variational autoencoder to solve our problem.

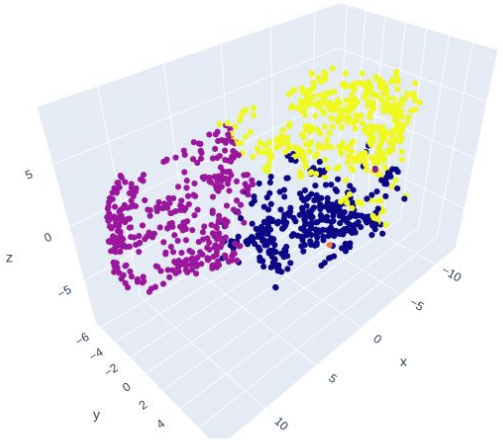

Figure 6: Problem with K-means clustering.

Variational autoencoders

Autoencoders are unsupervised learning models which are helpful when we want to automatically useful representation of the data. These are Neural network models designed for compressing the representations of datapoints and trained by trying to reconstruct the original data from the compressed one.

The main idea of a variational autoencoder as compared to standard autoencoder is that it embeds the input *X* to distribution rather than a point. And then a random sample is taken from the distribution rather than generated from encoder directly.

Advantage of the variational approach is that then we can sample new points from the output distribution. The loss function for variational autoencoder is as follows.

$$l_i(\theta, \phi) = -E_{z \sim q_\phi(z|x_i)}[log_{p_\theta}(x_i|z)] + KL(q_\phi(z|x_i)||p(z))$$

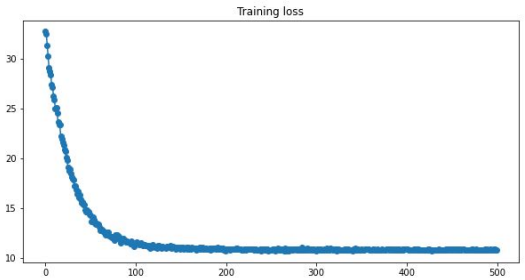

Figure 7: Loss plot for the training of our variational auto encoder.

Algorithm for VAE:

- Scale and normalize the data (this is good for training any neural network)
- Create an encoder that outputs mean and variance of intermediate representations.
- Sample data points from the same distribution and pass them through the decoder.
- calculate the reconstruction loss between the input to encoder and output of the decoder.
- Train the model to minimize reconstruction loss.
- Use the trained encoder to encode the data in a lower-dimensional space. (for example, take only the output corresponding to means).
- Use standard clustering algorithms to this non-linear smaller representation of the dataset.
- Identify fraudulent taxpayers using the clusters.

Results

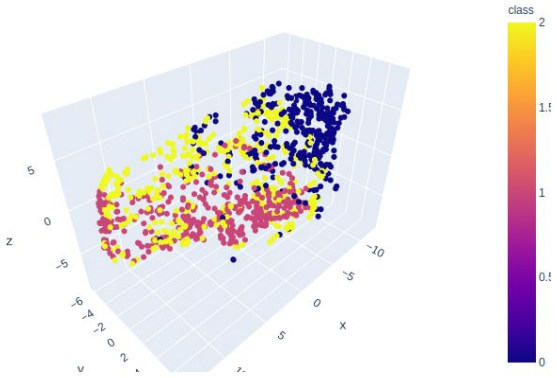

Figure 8: Results showing clusters using k=3.

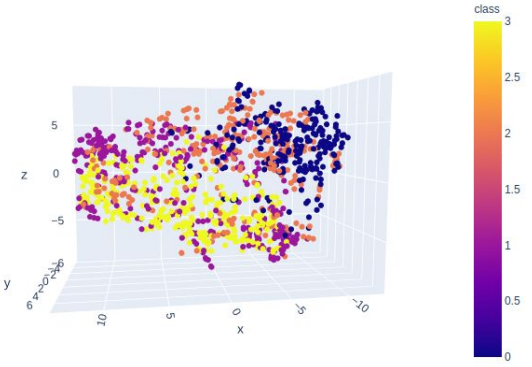

Figure 9: Results showing clusters using k=4.

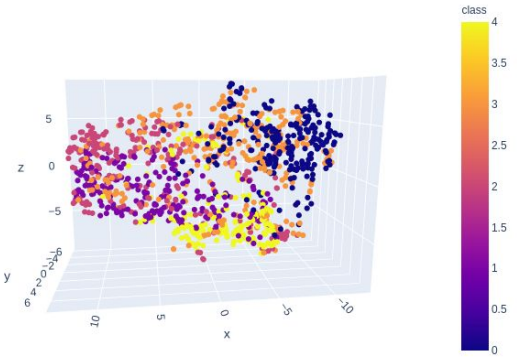

Figure 10: Results showing clusters using k=5.

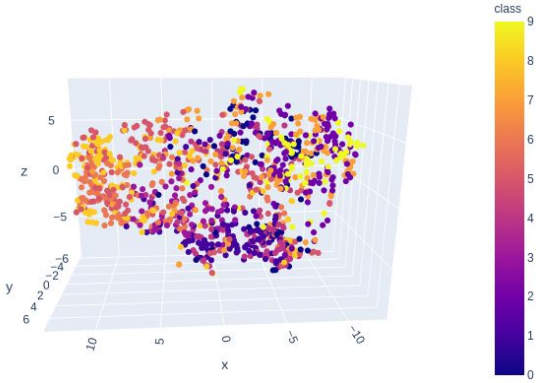

Figure 10: Results showing clusters using k=3.

Conclusion

For this particular dataset variational autoencoders did not create many clusters with the small number of data points. The reason for that is quite clear if we think about it. While training each datapoint had the same weight to it so the model has no incentive to learn a very unique representation for a rare fraudulent data point. So as results, when clustered most of the clusters, are of the nearly the same size. As compared to the spectral clustering approach from the first assignment

References

- https://www.jeremyjordan.me/autoencoders/
- https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73
- https://towardsdatascience.com/tutorial-on-variational-graph-auto-encoders-da9333281129