**Drexel University**

**College of Computing and Informatics**

**INFO 442 – Data Science Project**

**Weekly Progress Report**

**Due Date: Sunday of Week 1-8**

**Each team is expected to submit a weekly progress report during week 1-8. The goal of these reports is not just to track your progress, but to facilitate learning. Be honest about your challenges and don't be afraid to ask questions**.

**Team Name**:

**Team Members**:

**Week Number**:

1. **Accomplishments**: Detail what your team has achieved this week. This could include any new models you've tried, any new techniques you've learned, or any significant steps forward in your understanding of the problem.

This week:

* We studied few machine and deep learning models that could be used to analyze protein embedding data. Our aim was to figure out model could handle sequential and high-dimensional nature of data. We opted LSTM model for this week due to its ability to handle sequential plus high-dimensional data along with that it also provides memory and context preservation.
* We learned about the feature engineering principles and its techniques. We have applied PCA for this week which will essentially reduce the dimension of our dataset while preserving variance as much as possible.

1. **Challenges and Solutions**: Describe any challenges or roadblocks you encountered this week. This could be technical (issues with the code or the tools), conceptual (struggling to understand certain aspects of protein function), or related to your modeling approach. Also, detail the solutions you found for these challenges.

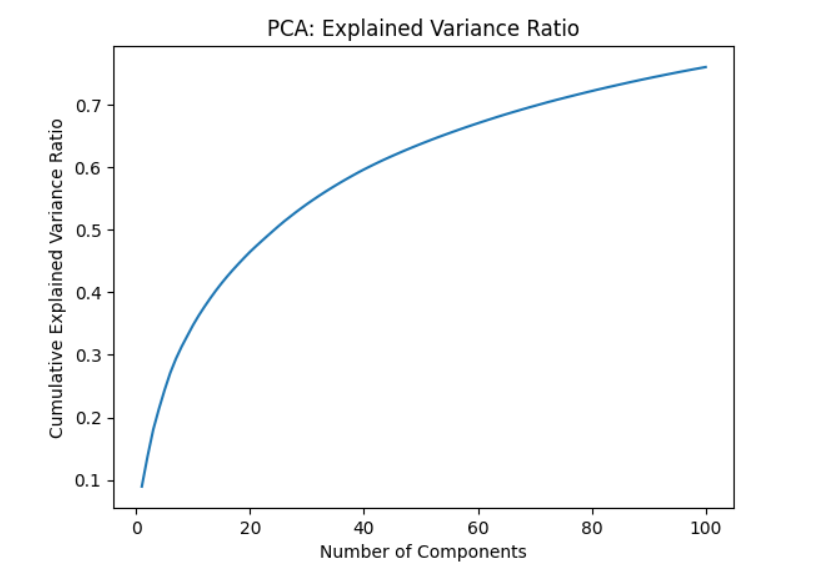
* Selecting the optimal learning rate for our protein embedding LSTM model. Learning rate is a type of hyper parameter that controls how quickly model learns. If it is too high, then the model will over-fit and if it is too low then the model will under-fit. We did few experiments with that and found that 0.001 learning rate works well.
* The another challenge was setting the number of components in PA that could capture 75% of variance in protein embedding training data. We have decided 75% as our threshold point and found that 100 leads us to have 75% variance.

1. **Data Understanding**: Discuss any new insights or understanding about the data you've gained this week. Include any relevant data visualization or statistics.

We have gained understanding about different types of representation for protein data.

* Protein data can be represented as 20 alphabets. There are twenty amino acids that constitute proteins so they can be conveniently represented by alphabets as symbols.
* Another powerful way to represent proteins is through graphs or networks. A node in a protein-protein interaction (PPI) network corresponds to a protein and each link represents an event that the two proteins physically bind (or interact) to carry out a biologically meaningful function.
* Gene regulatory networks (usually directed graphs) can be formed based on the expression patters between of the particular groups of genes.
* There could also be geometric representations of proteins using the 3D coordinates of each atom in its structure which could also be useful for the application of techniques from computer vision; e.g. geometric hashing.
* Protein can also be viewed as time-series data.

One of the important insight for this week is:



The graph above provides insights into the amount of variance that is captured by principal component.

1. **Feature Engineering**: Describe any steps you've taken towards feature engineering. What changes did you make to the input data, and why? What impact have these changes had on your model performance?

We have applied Principal Component Analysis (PCA) to reduce the dimensionality of protein embedding while retaining the most significant features of our dataset. By reducing dimensionality, we have removed redundant information or we can say that we have only kept the significantly contributing features. After applying feature engineering, the accuracy during each epoch is increased.

1. **Modeling**: Detail any new models you've tried this week. Include the type of model, the specific parameters you used, and why you chose this model and these parameters.

This week, we tried Long Short-Term Memory (LSTM) model. LSTM is a recurrent neural network model. We have chosen this because protein sequences data is sequential in nature and LSTM are known for effectively capturing and determining dependencies in sequential data along with memory and context preservation.

We have set the batch size of 5120. The model layers include:

* A Batch normalization layer for faster convergence after normalization
* A Reshape layer to ensure compatibility with LSTM layers
* Three LSTM layers with 512 units each. We have set the return\_sequence to true for these three layers to capture dependencies in sequential data,
* A Flattening layer to flatten the output into 1D array
* A Dense layer for final prediction.

We have used the Adam as optimizer with learning\_rate set to 0.001.

The loss function we opted is binary crossentropy, which is suitable for binary classification tasks.

To monitor model performance, we have used binary accuracy and AUC metrics.

The model is trained for 5 epochs.

1. **Model Evaluation**: Describe how you evaluated your models. What metrics did you use? What are your scores? How do they compare to the starter code score of 0.39? Provide an explanation if your score is higher or lower.

After training, model is evaluated using evaluate() function. The evaluation is performed on dimensionally reduced data. We have used the binary accuracy metric for evaluation. The accuracy score was 0.9799.

1. **Leaderboard Position**: Mention your current position on the Kaggle leaderboard and how it has changed over the week.
2. **Next Steps**: What are your plans for next week? What strategies, models, or techniques will you try? What are your specific goals?

Next week plans include:

* Apply another feature engineering technique to see if it can improve accuracy. We will be specifically focused on technique that can reduce dimensionality (1024 columns) and extract more meaningful features only.
* We will incorporate a protein BERT which is a deep-learning model specifically designed for protein related problems.