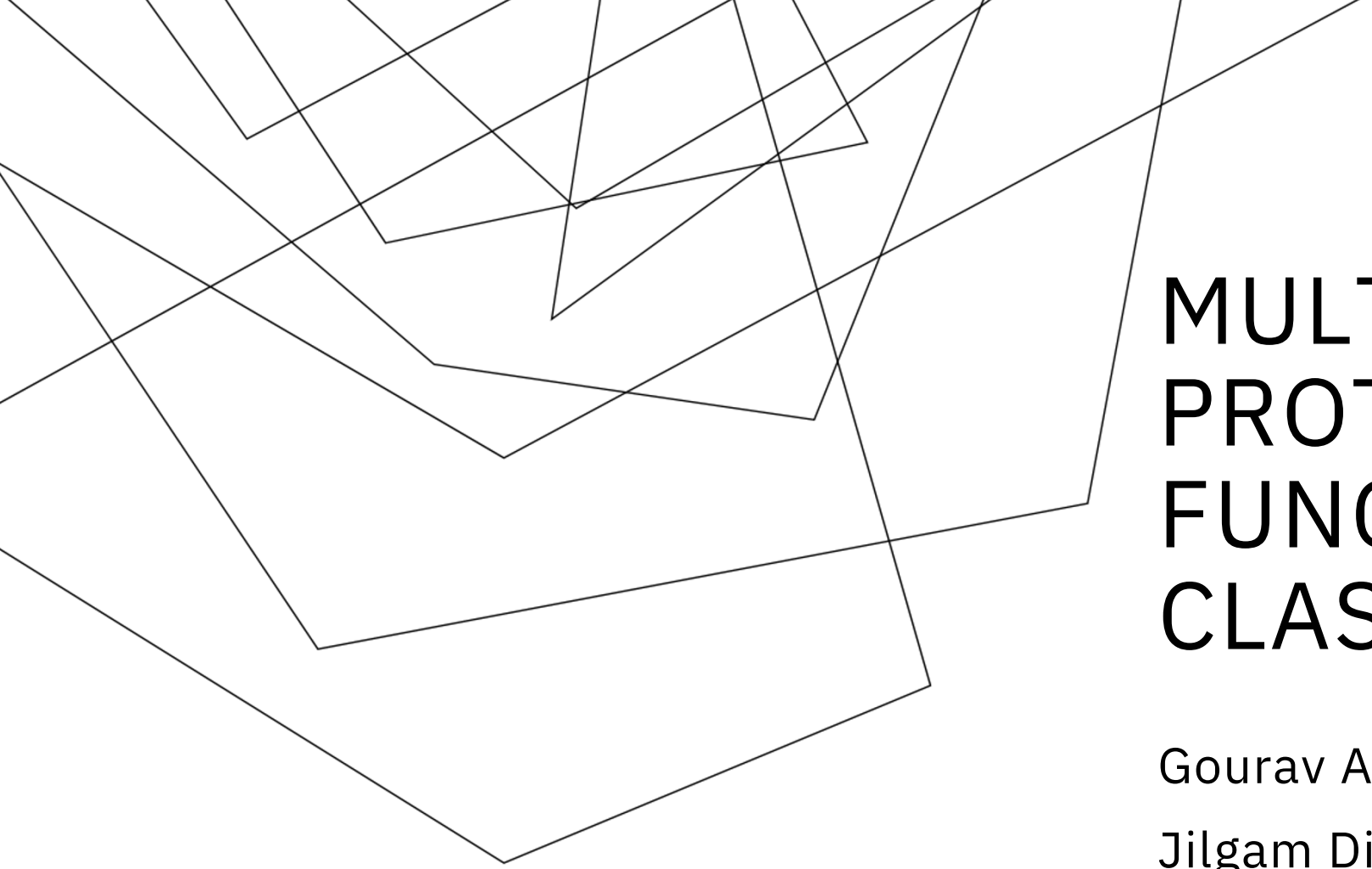


MULTI-LABEL PROTEIN FUNCTION CLASSIFICATION

Gourav Ahlawat(210001019)

Jilgam Dinesh(210001026)

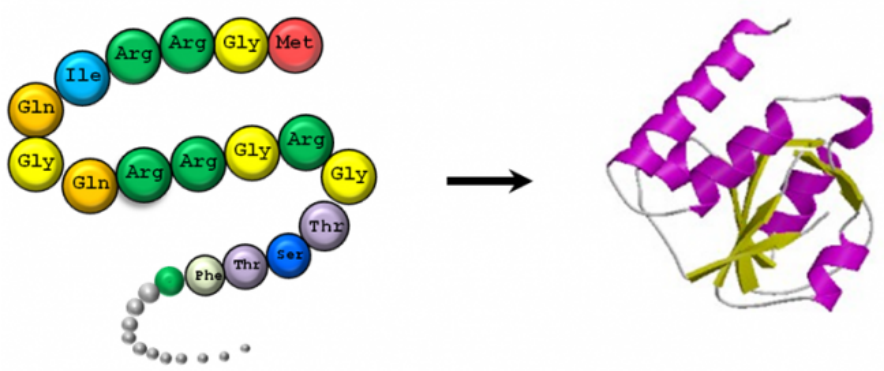
Yash Vashistha(210001082)

AGENDA

- Introduction
- Dataset
- Model
- Approach
- Experimentation
- Results
- References

INTRODUCTION

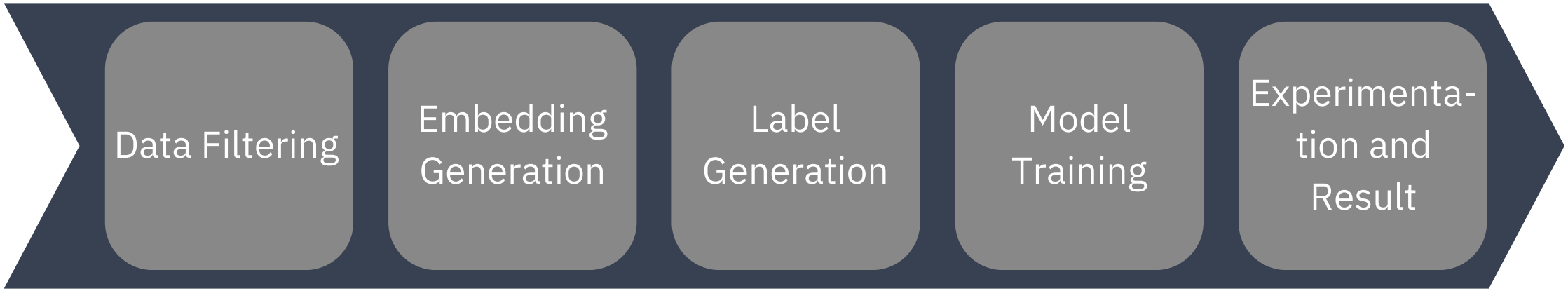
Our project focuses on protein function prediction (GO term Id) through machine learning. Our task is to predict a protein sequence's terms (functions). One protein sequence can have many functions and can thus be classified into several terms. Each term is uniquely identified by a GO Term ID. Thus our model has to predict all the GO Term IDs for a protein sequence. This means that the task at hand is a multi-label classification problem.

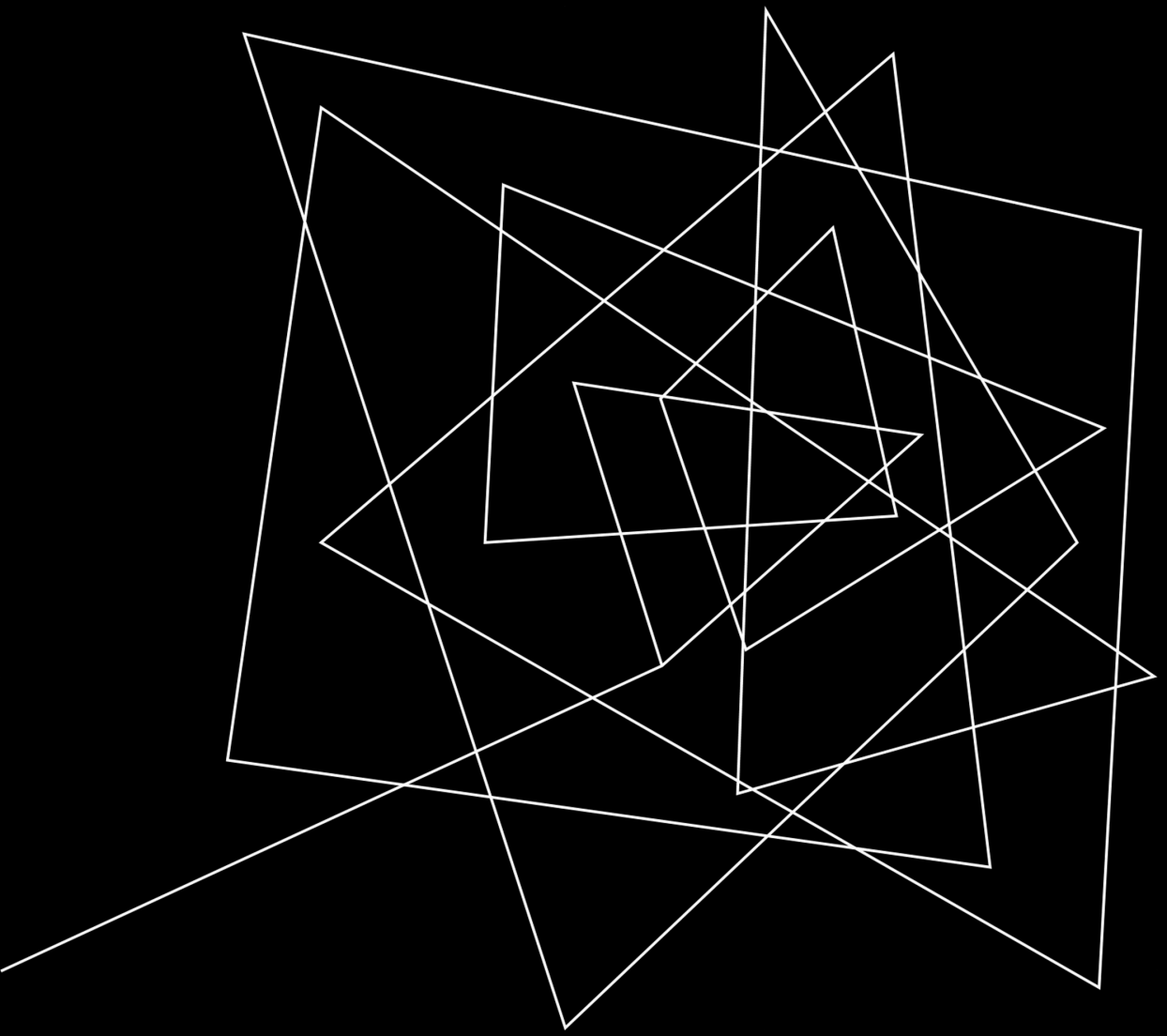


THE GENE ONTOLOGY(GO)

- Gene Ontology (GO) is a major bioinformatics initiative to unify the representation of gene and gene product attributes across all species.
- The Gene Ontology project provides an ontology of defined terms representing gene product properties. The ontology covers three domains:
- Cellular Component(CCO), the parts of a cell or its extracellular environment.
- Molecular Function(MFO), the elemental activities of a gene product at the molecular level, such as binding or catalysis.
- Biological Process(BPO), operations or sets of molecular events with a defined beginning and end, pertinent to the functioning of integrated living units: cells, tissues, organs, and organisms.
- Each GO term is associated to one of the three sub-ontologies.

PROJECT PIPELINE





DATASET

Used a dataset from a Kaggle Competition

DATASET DESCRIPTION

The dataset contains the following files:

- **go-basic.obo -** ontology graph structure.
- **train_sequeces.fasta -** amino acid sequences for proteins in training set.
- **train_taxonomy.tsv -** taxon ID for proteins in training set.
- **train_terms.tsv -** the training set of proteins and corresponding annotated GO terms.

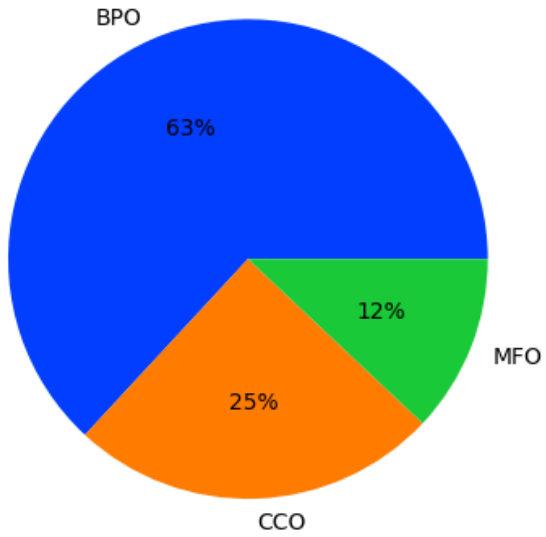
EMBEDDING GENERATION

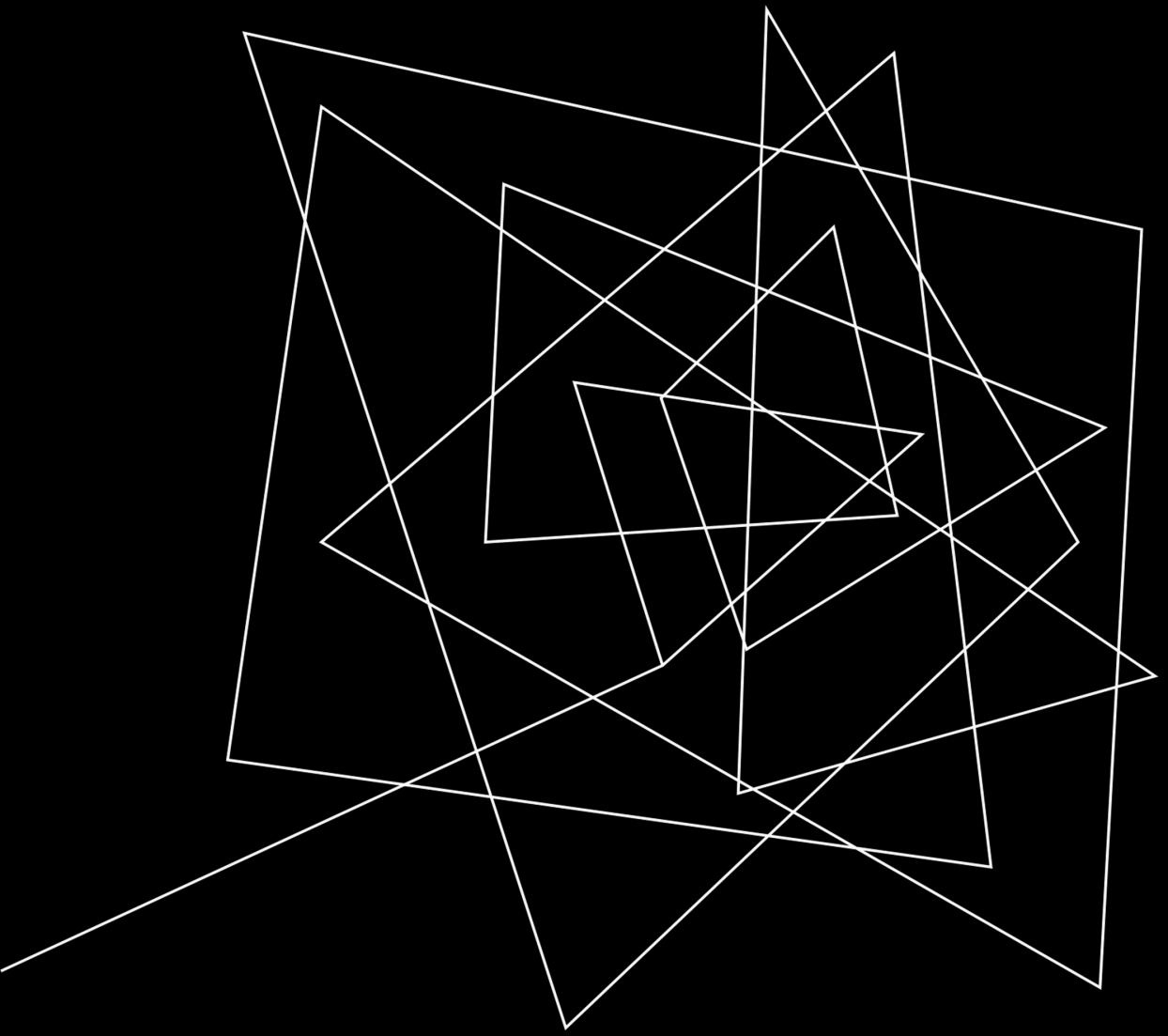
To train a machine learning model, we cannot directly use the alphabetical protein sequences in training-fasta file. They have to be converted into a vector format. In our project, we will use embeddings of the protein sequences to train the model. You can think of protein embeddings to be similar to word embeddings used to train NLP models. Protein embeddings are a machine-friendly method of capturing the protein's structural and functional characteristics, mainly through its sequence. One approach is to train a custom ML model to learn the protein sequences' protein embeddings in the dataset used in this notebook.

DATASET PREPARATION

First, we extracted all the needed labels(GO term ID) from the training-tsv file. There are more than 40,000 labels. To simplify our model, we will choose the most frequent 1500 GO term IDs as labels. We decided to work with the top 1500 most common features since otherwise, the feature space would begin too large and sparse. Next, we created a new dataframe by filtering the train terms with the selected GO Term IDs.The final labels dataframe is composed of 1500 columns and 142246 entries

On plotting the most frequent 1500 GO term IDs, we see that majority GO term IDs have BPO(Biological Process Ontology) as their aspect. Since this is a multi-label classification problem, in the labels array we will denote the presence or absence of each Go Term Id for a protein id using a 1 or 0.





MODEL

T5 and Deep Neural Network based Model

Text-to-Text Transfer Transformer (T5)

Overview

T5 is an encoder-decoder model pre-trained on a multi-task mixture of unsupervised and supervised tasks and for which each task is converted into a text-to-text format. T5 works well on a variety of tasks out-of-the-box by prepending a different prefix to the input corresponding to each task, e.g., for translation: translate English to German: ..., for summarization: summarize: ....

Unified Framework

T5 adopts a single architecture for all NLP tasks, simplifying model development and deployment by eliminating the need for task-specific architectures.

Text-to-Text Format

By framing tasks as text input and output, T5 provides a consistent and intuitive way to represent diverse NLP tasks, promoting simplicity and versatility in model training and usage.

Encoder-Decoder Architecture

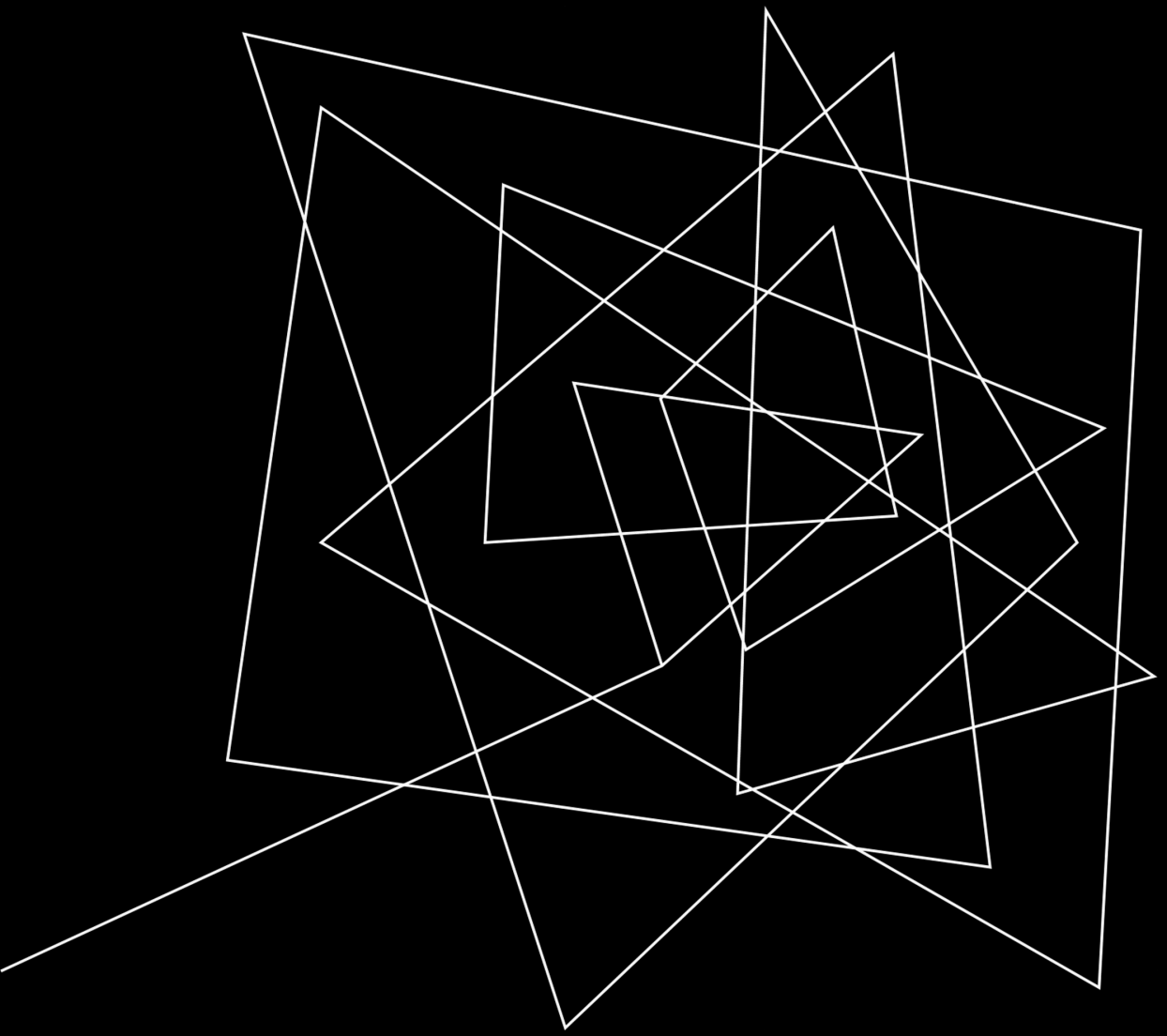
T5 leverages the Transformer architecture, comprising encoder and decoder components, allowing it to capture complex linguistic patterns and relationships in both input and output sequences.

Text-to-Text Transfer Transformer (T5)

Pre-training and Fine-tuning	Task-agnostic Objective	Flexible Decoding	Usecase
Through pre-training on large text corpora and fine-tuning on task-specific data, T5 harnesses both unsupervised and supervised learning to acquire broad linguistic knowledge and adapt to specific tasks effectively.	T5 learns to generate target text from input text during pre-training, irrespective of the task, facilitating the acquisition of a versatile language understanding that can be fine-tuned for various downstream applications.	T5 employs different decoding strategies like greedy decoding, beam search, and sampling during inference, enabling it to generate diverse and high-quality output sequences tailored to specific task requirements and preferences.	Protein sequences are converted to vector embedding using T5Tokenizer and T5Encoder from transformers available on hugging face. The result is a vector embedding of length 1024 which was used as input for the DNN model training.

PERFORMANCE METRICS

Binary Accuracy	Binary accuracy is a simple and commonly used metric for evaluating classification models. It measures the proportion of correct predictions made by the model among all predictions made. In binary classification problems, where there are only two possible outcomes (e.g., true or false, positive or negative), binary accuracy calculates the ratio of correct predictions to the total number of predictions.
Binary CrossEntropy	Binary crossentropy, also known as log loss, is a loss function used in binary classification tasks to measure the difference between predicted probabilities and true binary labels. It quantifies the difference between the predicted probability distribution and the actual distribution of the labels. Lower values of binary crossentropy indicate better model performance.
Area Under the ROC Curve (AUC)	AUC evaluates binary classification models by measuring the area under the ROC curve. This curve plots sensitivity against 1-specificity at various thresholds. A higher AUC indicates better discrimination between positive and negative classes, providing a single scalar value to represent overall model performance.
Precision and Recall	Precision measures the accuracy of positive predictions, while recall measures the ability of the model to correctly identify all positive instances. They are crucial in evaluating model performance, especially in imbalanced datasets. The F1 score, the harmonic mean of precision and recall, is commonly used to balance them.

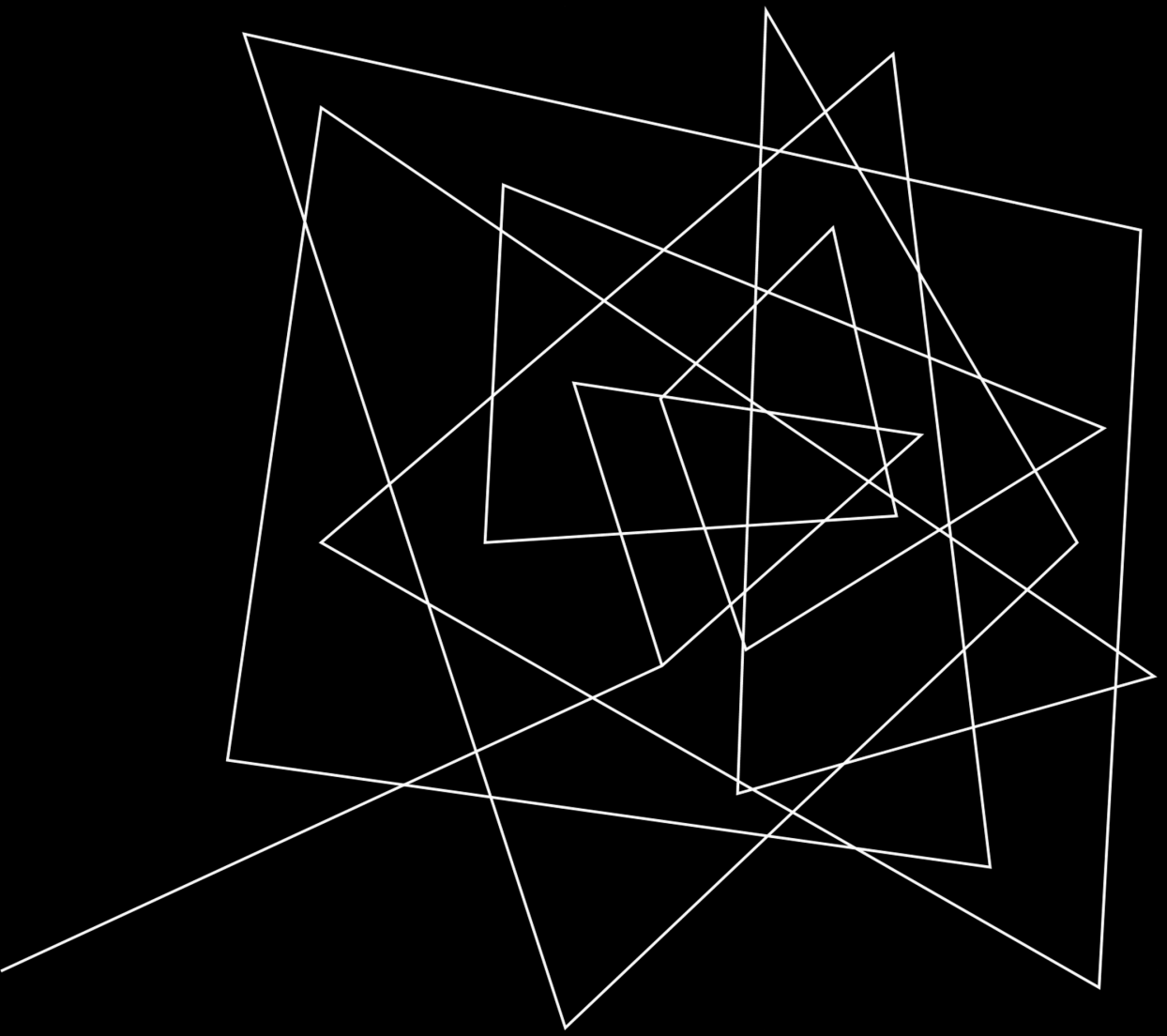


APPROACH

The Strategy

APPROACH FOLLOWED

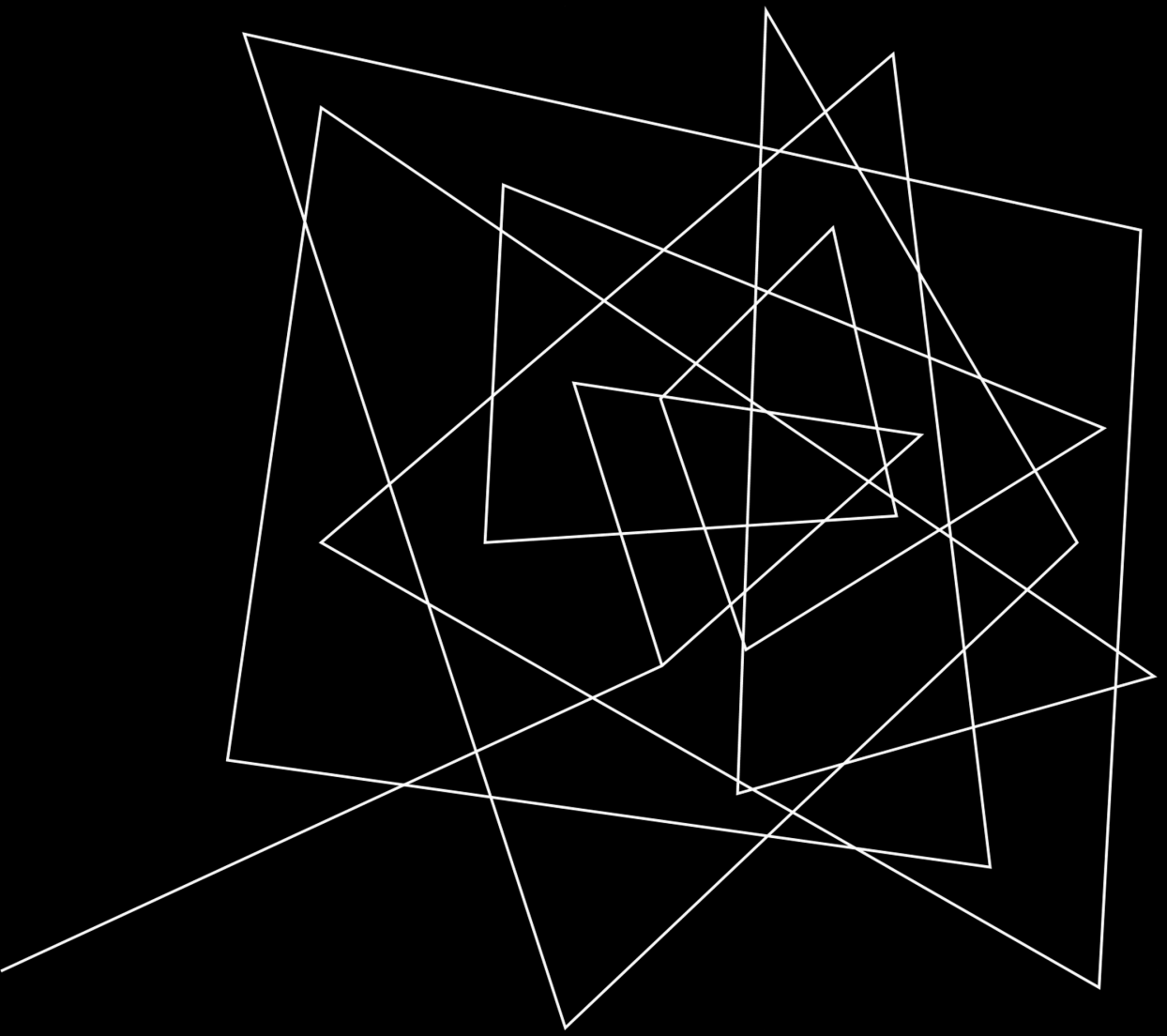
- First, we created embeddings from the fasta files using transformer-based models, then used a Deep Neural Net(DNN) based model and used the embeddings as input.
- Used dropout technique for reducing overfitting.
- Used larger batch size for faster training.
- Tried various thresholds to optimize for recall/precision lower for higher recall and vice versa



EXPERIMENTATION

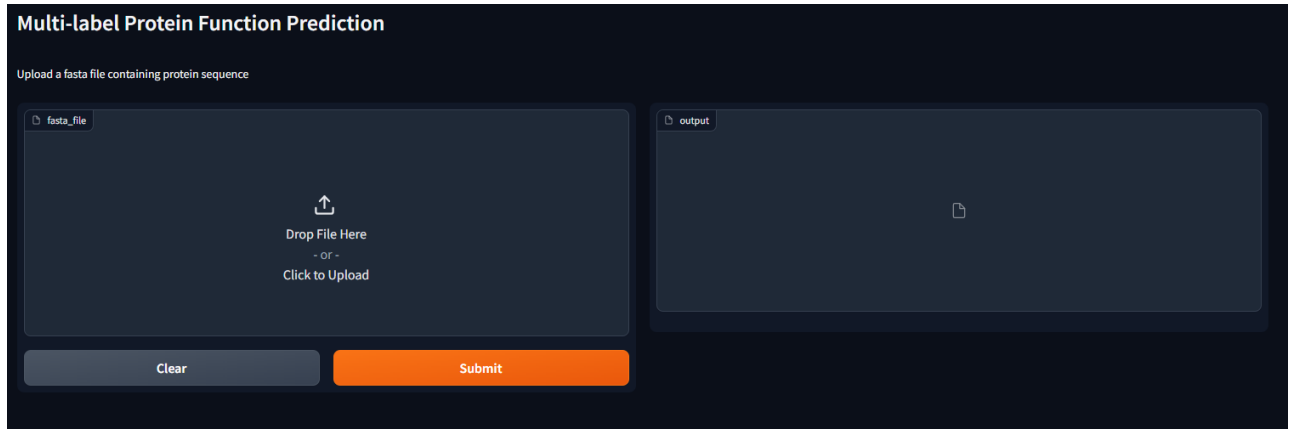
THINGS WE TRIED

- Tried different text embedding models like T5, Probert, esm2 etc.
- Among all the embedding models, T5 gave better results, and so we went ahead with it.
- Tried different model architectures by varying the number of layers and the neuron count of hidden layers.
- Checked various metrics like accuracy, loss, AUC, and precision-recall for all of them and used a validation dataset to check for overfitting.



RESULTS

Deployed Model




```
>Q9CQV8 10090
MTMDKSELVQKAKLAEQAERYDDMAAAMKAVTEQGHELSNEERNLLSVAYKNVVGARRSS
WRVISSIEQKTERNEKKQQMGKEYREKIEAELQDICNDVLELLDKYLILNATQAESKVFY
LKMKGDYFRYLSEVASGENKQTTVSNSQQAYQEAFEISKKEMQPTHPIRLGLALNFSVFY
YEILNSPEKACSLAKTAFDEAIAELDTLNEESYKDSTLIMQLLRDNLTLWTSENQGDEGD
AGEGEN
```

Input fasta file

	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	
65	GO:0050	GO:0050	GO:0050	GO:0008	GO:0032	GO:0005	GO:0032	GO:0071	GO:0048	GO:0016	GO:0003	GO:0044	GO:0044	GC
1	1	1	1	0	1	1	1	0	1	0	0	0	0	

Output csv file generated by model

PERFORMANCE METRICS

```
Final Training Metrics:
Loss: 0.05103842914104462
Binary Accuracy: 0.9834503531455994
AUC: 0.9628538489341736

Final Validation Metrics:
Loss: 0.058653220534324646
Binary Accuracy: 0.9814625382423401
AUC: 0.9393486380577087
```

ACKNOWLEDGEMENT

We are grateful to Dr. Aruna Tiwari for her guidance and Teaching Assistants for their support during our Computational Intelligence lab project. Their expertise and encouragement were invaluable.

REFERENCES

- https://www.kaggle.com/competitions/cafa-5-protein-function-prediction/data
- https://medium.com/analytics-vidhya/t5-a-detailed-explanation-a0ac9bc53e51
- https://huggingface.co/docs/transformers/en/model_doc/t5
- https://www.kaggle.com/datasets/sergeifironov/t5embeds
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6438694/#:~:text=The%20Gene%20Ontology%20(GO)%20considers,model%20assumed%20in%20GO%20annotations

THANK YOU