

A Weekly Update on

DiabetIQ: An Intelligent Diabetes Management Application with LLM-Augmented Chatbot and ML-Based Early Risk Prediction

12 March, 2025

Group Information

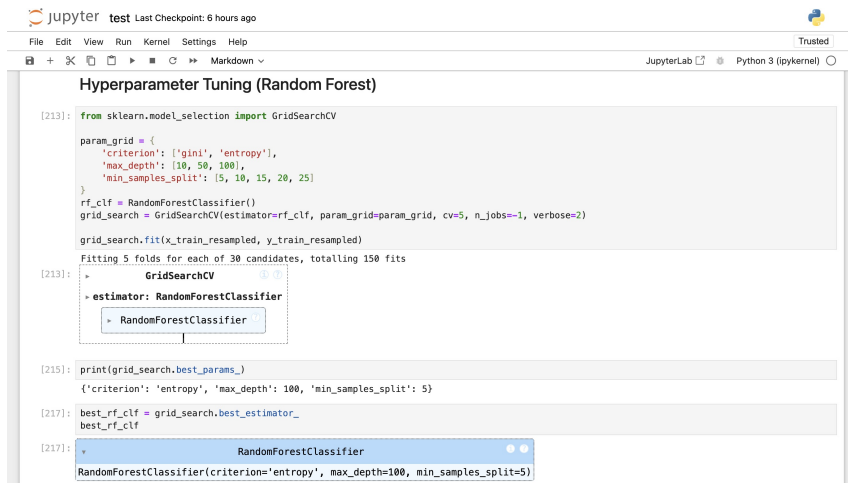
Group-01 CSE299 (Section-17)

1. Saif Mohammed - 2121913042
2. Nazibul Islam Nabil - 2222456642
3. Humayra Rahman Nipa - 2121128042
4. Umme Suraia Haque Setu - 2031278642

Weekly Update Brief

- ▶ **ML Model Train:** Applied Machine Learning Algorithms: Logistic Regression, SVM, Decision Tree, Random Forest, KNN with Hyperparameter Tuning (Grid Search Cross Validation).
- ▶ **LLM RAG Chatbot:** Processed BADAS Guideline 2019 PDF, implemented document chunking, embeddings, and retrieval for Q&A conversation.
- ▶ **Summary Documentation:** Completed a summary documentation on *Attention in Transformers: Concepts and Code in PyTorch*

Machine Learning (Model Train)



The screenshot shows a JupyterLab interface with a notebook titled "Hyperparameter Tuning (Random Forest)". The notebook contains four code cells. The first cell imports GridSearchCV and defines a parameter grid for a Random Forest classifier. The second cell creates a Random Forest classifier, sets up GridSearchCV, and fits it to the training data. The third cell prints the best parameters found. The fourth cell assigns the best estimator to a variable. The output of the second cell shows the GridSearchCV object and its estimator, which is a Random Forest classifier with the best parameters found.

```
[213]: from sklearn.model_selection import GridSearchCV

param_grid = {
    'criterion': ['gini', 'entropy'],
    'max_depth': [10, 50, 100],
    'min_samples_split': [5, 10, 15, 20, 25]
}

rf_clf = RandomForestClassifier()
grid_search = GridSearchCV(estimator=rf_clf, param_grid=param_grid, cv=5, n_jobs=-1, verbose=2)

grid_search.fit(x_train_resampled, y_train_resampled)

Fitting 5 folds for each of 30 candidates, totalling 150 fits

[213]: > GridSearchCV
> estimator: RandomForestClassifier
> RandomForestClassifier

[215]: print(grid_search.best_params_)

{'criterion': 'entropy', 'max_depth': 100, 'min_samples_split': 5}

[217]: best_rf_clf = grid_search.best_estimator_
best_rf_clf

[217]: RandomForestClassifier
RandomForestClassifier(criterion='entropy', max_depth=100, min_samples_split=5)
```

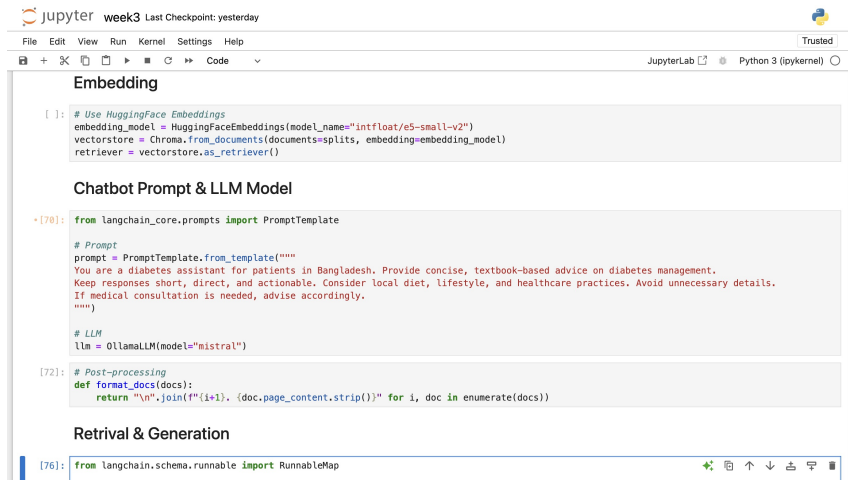
Figure: Hyperparameter Tuning (Random Forest)

Performance Metrics of ML Algorithms Using SMOTE

Serial	Classifier	Precision	Recall	F1-Score	Accuracy	AUC
1	Logistic Regression	0.32	0.77	0.45	88%	0.83
2	SVM	0.33	0.70	0.45	89%	0.80
3	Decision Tree	0.26	0.45	0.33	88%	0.68
4	Random Forest	0.46	0.45	0.46	93%	0.71
5	XGboost	0.56	0.39	0.46	94%	0.69
6	KNN	0.24	0.61	0.34	85%	0.74

Table: Performance Metrics of ML Algorithms Using SMOTE

LLM RAG Q&A Chatbot using LangChain



```
[ ]: # Use HuggingFace Embeddings
embedding_model = HuggingFaceEmbeddings(model_name="intfloat/e5-small-v2")
vectorstore = Chroma.from_documents(documents=splits, embedding=embedding_model)
retriever = vectorstore.as_retriever()

Chatbot Prompt & LLM Model

* [70]: from langchain_core.prompts import PromptTemplate

# Prompt
prompt = PromptTemplate.from_template("""
You are a diabetes assistant for patients in Bangladesh. Provide concise, textbook-based advice on diabetes management.
Keep responses short, direct, and actionable. Consider local diet, lifestyle, and healthcare practices. Avoid unnecessary details.
If medical consultation is needed, advise accordingly.
""")

# LLM
llm = OllamaLLM(model="mistral")

[72]: # Post-processing
def format_docs(docs):
    return "\n".join(f"{i+1}. {doc.page_content.strip()}" for i, doc in enumerate(docs))

Retrieval & Generation

[76]: from langchain.schema.runnable import RunnableMap
```

Figure: Chatbot Retrieval & Generation from textbook

Summary Documentation

Humayra Rahman Nipa

Id- 2121128042

Attention in Transformers: Concepts and Code

Early Translation Methods:

- Early translation methods faced issues with word order and sentence length.
- They used a single vector to represent whole sentences, which wasn't effective.
- The introduction of the attention mechanism improved translation quality.
 - It allows words to be considered individually, maintaining their meanings.
 - Helps in better understanding the context, making translations more accurate.
 - Contextual embeddings show how word meanings can change depending on surrounding words.

ChatGPT and the Transformer Architecture:

- ChatGPT is an AI model that answers questions and engages in

Figure: Summary Documentation on Transformers

Achievements

- ▶ Applied Machine Learning Algorithms for machine learning model development.
- ▶ Successfully implemented basic retrieval and generation from the textbook.
- ▶ Integrated HuggingFaceEmbeddings for improved text representation.
- ▶ Built an initial RAG pipeline using LangChain and ChromaDB.
- ▶ Summarized Documentation on Transformers.

Technology Stack

- ▶ **Programming Language:** Python
- ▶ **Framework:** LangChain
- ▶ **Libraries:** NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn
- ▶ **Embedding Model:** HuggingFaceEmbeddings: *intfloat/e5-small-v2*
- ▶ **Vector Database:** chromadb
- ▶ **LLM:** Ollama LLM: *mistral*
- ▶ **Document Processing:** PyPDF

Work Distribution (This Week)

- ▶ **Saif Mohammed - 2121913042**

- ▶ ML Algorithms Application
- ▶ ML Algorithms Evaluation

- ▶ **Nazibul Islam Nabil - 2222456642**

- ▶ LLM Q&A Chatbot Prototype





- ▶ **Humayra Rahman Nipa - 2121128042**

- ▶ Summary Documentation on Transformers

- ▶ **Umme Suraia Haque Setu - 2031278642**

- ▶ Summary Documentation on Transformers

References

-  A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 3rd ed. Sebastopol, CA, USA: O'Reilly Media, 2023.
-  LangChain Engineer, “LLM Part: Learn RAG From Scratch – Python AI Tutorial,” YouTube. [Online]. Available: <https://www.youtube.com/>. [Accessed: Feb. 19, 2025].
-  P. Lewis, E. Denoyer, and S. Riedel, “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” *arXiv preprint arXiv:2005.11401*, May 2020. [Online]. Available: <https://arxiv.org/abs/2005.11401>.
-  DeepLearning.AI, *Attention in Transformers: Concepts and Code in PyTorch*. [Online]. Available: <https://www.deeplearning.ai/short-courses/attention-in-transformers-concepts-and-code-in-pytorch/>. [Accessed: Mar. 11, 2025.]