# Project 1: HPI Agent

Assigned to:

**Background:**

In US, whenever a patient visits a provider (synonymous to doctor), the patient describes a “Chief Complaint” – the primary reason for visit to the provider. The provider then write a SOAP note for that visit/encounter. SOAP consists of following 4 parts (detail in Appendix):

1. Subjective
2. Objective
3. Assessment
4. Plan

HPI refers to history of presenting illness. Upon first visit, the nurse collects a detailed history of the patient in addition to the presenting complaint. The provider may also ask several follow-up questions from a patient based on their presenting complaint to evaluate their condition and relevant history (medical, family, social, allergies etc.).

We aim to automate this process to save time on each encounter, by creating an agent that can accurately collect relevant history.

**Problem Statement:**

Given a patient’s “Chief Complaint”, ask relevant questions and make a complete, accurate and relevant HPI (History of Present Illness) and other histories to be added into SOAP note.

**Tasks:**

* Analyze different SOAP notes to get an idea of the structure and significance of each part.
* Research about guidelines regarding SOAP.
* RnD about different methods to make HPI, and how they can be effectively combined for best results
  + Use already made templates of Providers
  + Use guidelines
  + Use LLM.
* Develop and compare different methodologies individually and in a collaborative setting.
* Prepare a report that includes a detailed description of the project: literature review, data insights and preprocessing steps, model architectures, training/development methodologies, RAG techniques and evaluation results. Present the results and insights gained from the analysis. Additionally, provide recommendations for future improvements.

**Deliverables:**

* Develop a system having following properties: (for all providers or a sub-set of providers (specifically North Carolina Providers).
  1. Input: Chief Complaint
  2. Objective: Ask follow-up questions
  3. Output: HPI
* Report evaluation metrics of above system. How did you determine the accuracy/effectiveness of the system?
* A project report summarizing the entire project, including the background, prior art, problem statement, methodology, results, and conclusions. This documentation should be clear and organized for future reference and replication.
* A presentation summarizing the key findings, methodologies, and outcomes of the project.
* Properly documented source code.

**References:**

1. <https://www.ncbi.nlm.nih.gov/books/NBK482263/>
2. <https://github.com/rajpurkarlab/craft-md?tab=readme-ov-file> (<https://www.medrxiv.org/content/10.1101/2023.09.12.23295399v2>)
3. <https://arxiv.org/pdf/2406.00922v1>
4. <https://arxiv.org/abs/2312.02441>
5. Robbins Basic Pathology
6. Macleod’s Clinical Examination
7. Bates guide to physical examination and history taking
8. <https://ada.com/>
9. <https://www.drlex.ai/>
10. <https://litfl.com/>
11. <https://www.wolterskluwer.com/en/solutions/uptodate>
12. <https://www.deeplearning.ai/the-batch/amie-a-chatbot-that-outperforms-doctors-in-diagnostic-conversations/>
13. <https://arxiv.org/pdf/2403.09057v3>
14. <https://www.nature.com/articles/s41598-024-64827-6>
15. <https://www.figma.com/proto/04x7YNI5cqpipA0RHobkKO/novelHealth?node-id=2800-18701&node-type=canvas&viewport=5869%2C4621%2C0.92&t=sFlQEa7hblsq9OqT-0&scaling=scale-down&content-scaling=fixed&starting-point-node-id=2800%3A18701&show-proto-sidebar=1>

# Project 2: Super MAS

Assigned to:

**Background**:  
While multiple frameworks exist to ease creation of agents and MAS that rely on LLMs, there is a gap between potential and expertise. We aim to develop an MAS that specializes in creating agents and small MAS systems for users with no expertise in LLMs or LangGraph. It should be capable of receiving descriptions of the various agents (beliefs, goals, plans), and the associated tools, while the code and prompt generation should be handled by the super MAS.

**Problem Statement:**Develop an MAS (using LLMs and LangGraph) that is capable of developing other simple agents/MAS given a description of goals and behaviors

**Tasks:**

* Understand agents:
  + Agentic workflows
  + Multiagent systems
  + Development frameworks
* Explore and test appropriate LLMs and prompt engineering techniques for code generation/system development.
* Create a simple agent (or MAS) in LangGraph (CureBuzz, code evaluator) to understand how agents are developed.
* Identify a set of minimal, complete, necessary inputs that would suffice the automatic development of such simple agents.
* Develop a system capable of generating working agents in LangGraph provided your devised set of inputs. The system may itself be an agent (or MAS) in LangGraph, but this is not a requirement.
* Evaluate and improve your system to involve minimal human input for the automatic development of simple agents.

**Deliverables:**

* A system capable of developing simple agents and MAS in LangGraph, provided a set of requirements and other necessary user inputs.
* Simple user-friendly application to demonstrate the working of your system, targeting any simple use case.
* Devise and report evaluation metrics and criteria of the above system.
* A project report summarizing the entire project, including the background, prior art, problem statement, methodology, results, and conclusions. This documentation should be clear and organized for future reference and replication.
* A presentation summarizing the key findings, methodologies, and outcomes of the project.
* Properly documented source code.

**References:**

* <https://superagi.com/multi-agent-system/>
* <https://www.deeplearning.ai/short-courses/ai-agents-in-langgraph/>
* [https://blog.langchain.dev/langgraph-multi-agent-workflows/](https://blog.langchain.dev/langgraph-multi-agent-workflows/?utm_source=chatgpt.com)
* [https://langchain-ai.github.io/langgraph/concepts/multi\_agent/](https://langchain-ai.github.io/langgraph/concepts/multi_agent/?utm_source=chatgpt.com)
* <https://walkingtree.tech/empowering-virtual-sales-agents-with-ai-driven-automation-using-langgraph/>
* <https://blog.langchain.dev/code-execution-with-langgraph/>
* <https://langchain-ai.github.io/langgraph/tutorials/code_assistant/langgraph_code_assistant/>
* <https://wandb.ai/tensorgirl/PythonCodeGenerator/reports/Building-a-Coding-Assistant-using-LangChain-and-CodeLlama-with-QLoRA---Vmlldzo2NTg0NTE0>

# Project 3: XML classification

Assigned to:

**Background:**

In US, whenever a patient visits a provider (synonymous to doctor), the patient describes a “Chief Complaint” – the primary reason for visit to the provider. The provider then writes a SOAP note for that visit/encounter. Once the subjective (S) and objective (O) are documented, we want an AI model to suggest likely ICD10 codes to assist doctor in assessment, planning and billing. Your development strategy should utilize appropriate language models to generate text embeddings, which are then fed to any XML classification technique.

Instead of creating predictive models that predict one disease or so, we want to create a general solution for all types of complaints and visits. This requires the use of the presenting complaint - description of symptoms and relevant medical history, which is only available in clinical notes i.e. free text. Thus, the solution must incorporate some language model to create features from a text paragraph, which can the be fed into an XML classification model.

**Problem Statement:**

Develop an XML classification model that uses small language models (e.g. clinical BERT) to understand a provided HPI and complaint to assign relevant ICD10 codes (from all possible ICD10 codes)

**Tasks:**

* Research foundational models for clinical text and extreme multi-label classification tools and methods.
* Analyze and understand the notes and diagnoses data - structure, and how to extract relevant portions
* Use appropriate data pre-processing and feature engineering using NLP (embeddings) models and bid data technologies
* Experiment with different models Compare performance across models and ensure proper validation and testing to assess model robustness.
* You may consider training multiple models focused on groups of ICD10 codes (e.g. disease codes vs billing codes vs social history codes etc.)
* Prepare a report that includes a detailed description of the project: literature review, data preprocessing steps, feature engineering techniques, model architectures, training methodologies, and performance metrics. Present the results and insights gained from the analysis. Additionally, provide recommendations for future improvements.

**Deliverables:**

* An XML classification model that can predict multiple ICD10 codes provided a complaint (and HPI) text.
* Simple user-friendly application to demonstrate the working of your model, with the ability to compare predictions with actual outcomes, and analyzing the model’s results.
* Model evaluation report to provide insights into the models' performance
* Report the most important features and variables that contribute to the prediction. Incorporate XAI to enhance transparency.
* A project report summarizing the entire project, including the background, prior art, problem statement, methodology, results, and conclusions. This documentation should be clear and organized for future reference and replication.
* A presentation summarizing the key findings, methodologies, and outcomes of the project.
* Properly documented source code.

**References:**

* <http://manikvarma.org/downloads/XC/XMLRepository.html>
* <https://www.linkedin.com/pulse/ranking-foundational-models-use-healthcare-rahul-garg-md-mba--rwzsc/>

# Project 4: SmartDialer 2.0

Assigned to:

**Background:**

Our organization provides insurance billing services to healthcare providers, which involves generating claims, submitting them to insurance companies, and resolving any issues that arise. Claims that remain unsettled after 40 days require further follow-up. At this stage, we use the SMART DIALER system to contact insurance companies and resolve claim rejections, ensuring timely payments for the providers.

During these interactions, insurance companies often employ Interactive Voice Response (IVR) systems. These systems request claim-related information (e.g., claim identifiers) before transferring the call to a human representative. Each IVR system has its unique structure, varying in the type, sequence, and format of information requested. To manage this process, we rely on the following workflow:

1. Call Initiation: Human Agents calls the insurance company.
2. IVR Interaction: The IVR system prompts for claim-related details.
3. Mapping and Configuration: A human agent manually maps the IVR process, identifying:
   1. The information requested by the IVR.
   2. The order and medium (spoken or text input) required for responses.
   3. The phrases used by the IVR for each query.
4. SMART DIALER Configuration: Based on the mapping, the SMART DIALER is configured to automatically handle future interactions with the insurance.
5. Human Handoff: Once the IVR transfers the call to a human representative, the SMART DIALER notifies a human agent to join the call.
6. This approach enables the SMART DIALER to streamline IVR interactions and reduce wait times for human agents. However, the process involves significant manual intervention and faces challenges as outlined below.

The Problem:

* Dynamic IVR Systems: IVR systems frequently change their prompts, modify the order of requested information, or alter the phrasing of their queries. Each change requires manual updates to the SMART DIALER configuration, which is time-consuming and resource-intensive.
* Scalability Challenges: Expanding the number of insurance providers covered increases the workload for manual mapping and configuration. This limits the scalability of the current process and creates bottlenecks.
* Manual Intervention Dependency: Despite automation, the system relies heavily on human agents to analyze and configure IVR mappings. This dependency reduces efficiency and increases operational costs.
* Error-Prone Process: Frequent updates and human involvement increase the risk of errors, leading to delays in resolving claims or failed interactions with IVR systems.

Leveraging the power of LLMs and their decision-making capability, you have to create an MAS System that will understand the query from the IVR system, and automatically trigger the required agent to respond back to the IVR system with the inquired information, Keeping the following features in mind:

1. Dynamic Adaptability: The system must handle frequent changes in IVR prompts without requiring extensive manual reconfiguration.
2. Scalability: It should seamlessly scale to accommodate a growing number of insurance providers and workflows.
3. Minimal Human Dependency: Reduce reliance on human intervention by automating the mapping and configuration process.
4. Fault Tolerance: The system must handle errors or unforeseen IVR scenarios gracefully, minimizing disruptions to the workflow.
5. Comprehensive Logging and Reporting: Enable accurate monitoring of IVR interactions and provide actionable insights to address any gaps

**Problem Statement:**

Create an MAS System that will handle the interactions with the IVR systems of the insurances and provide them with required responses.

**Tasks:**

* Understand the current SmartDialer system and agentic workflows
* Determine a list of possible “actions” and identify appropriate “tools” that would allow an LLM to perform those actions.
* Follow best practices for prompt engineering to write an LLM agent that correctly identifies the next action based on transcribed input received
* Design your agent to seamlessly fit into the SmartDialer system
* Evaluate and compare performance of your developed agent for:
  + Existing dial plans (handled in the current system)
  + New dial plans
* Prepare a report that includes a detailed description of the project: literature review, any data preparation steps, system architecture, development methodology, and evaluation results. Present the results and insights gained from the analysis. Additionally, provide recommendations for future improvements.

**Deliverables:**

* A revamped SmartDialer system that does not rely on dial plans/manual configurations, rather uses an LLM-powered MAS to perform all the necessary actions after receiving instructions from the IVR.
* Evaluation report to provide insights into the systems' performance, compared to the existing sytem. Analyze the pros and cons of your developed solution versus the existing one.
* A project report summarizing the entire project, including the background, prior art, problem statement, methodology, results, and conclusions. This documentation should be clear and organized for future reference and replication.
* A presentation summarizing the key findings, methodologies, and outcomes of the project.
* Properly documented source code.

**References:**

* <https://superagi.com/multi-agent-system/>
* <https://www.deeplearning.ai/short-courses/ai-agents-in-langgraph/>
* [https://blog.langchain.dev/langgraph-multi-agent-workflows/](https://blog.langchain.dev/langgraph-multi-agent-workflows/?utm_source=chatgpt.com)
* [https://langchain-ai.github.io/langgraph/concepts/multi\_agent/](https://langchain-ai.github.io/langgraph/concepts/multi_agent/?utm_source=chatgpt.com)
* <https://walkingtree.tech/empowering-virtual-sales-agents-with-ai-driven-automation-using-langgraph/>

# Project 5: Federated Learning for Disease Prediction Models Using Tree Ensembles

**Assigned To**:

**Background**:

In healthcare, predictive models are commonly used to forecast the likelihood of various diseases based on patient data. These models play a critical role in supporting clinicians with early diagnosis and treatment decisions. Traditionally, disease prediction models rely on centralized repositories that collect data from multiple practices. However, concerns regarding data privacy, compliance with regulations like HIPAA, and logistical challenges in transferring large volumes of sensitive data across institutions limit the feasibility of centralized solutions.

Federated learning (FL) is an emerging paradigm that enables collaborative model training across decentralized data sources while keeping the data localized. By employing FL, different practices can contribute to training disease prediction models without sharing raw data. Our current approach uses Random Forests (RF) in a centralized setup, which provides interpretable predictions but necessitates centralized data storage.

The goal of this project is to explore and implement a federated learning framework to train disease prediction models on distributed data from multiple practices. The framework will aim to preserve the benefits of tree ensembles, such as interpretability and high performance on tabular datasets, while addressing the challenges of federated setups, such as communication efficiency, model aggregation, and heterogeneity of data distributions across practices.

**Problem Statement**:

Develop a federated learning framework that enables the training of disease prediction models using tree ensemble techniques (e.g., Random Forests or Gradient Boosted Trees) on decentralized datasets from multiple practices. The framework should ensure data privacy, efficient model updates, and robust performance across diverse data distributions.

**Tasks**:

* Understand federated learning. Investigate existing federated learning frameworks, with a focus on those supporting tree ensemble models. Understand challenges in federated learning, including communication costs, data heterogeneity, and model aggregation techniques.
* Analyze the structure and characteristics of practice-specific data, focusing on disease-related features and labels.
* Identify preprocessing steps and feature engineering techniques suitable for distributed settings.
* Design and implement a federated learning pipeline that supports tree ensemble models, utilizing tools such as FedML, Flower, or custom solutions.
* Experiment with techniques for aggregating tree ensemble models, such as weighted averaging of decision trees or federated boosting methods.
* Train disease prediction models using federated learning and compare their performance with centralized approaches.
* Validate the models' robustness across data from diverse practices and evaluate their generalizability.
* Use explainable AI techniques to interpret the predictions of tree ensembles in the federated setup. Identify key features contributing to disease predictions to provide actionable insights to clinicians.
* Prepare a comprehensive report detailing the project, including background, methodology, results, and conclusions.
* Document the codebase, making it modular and reproducible for future extensions.

**Deliverables**:

* A federated learning framework that supports training disease prediction models with tree ensembles across multiple practices.
* Comprehensive comparison of federated learning performance against centralized models, highlighting trade-offs in privacy and accuracy.
* A report summarizing the methodology, results, and insights from the project.
* Visualizations demonstrating the explainability of predictions and the impact of features.
* A user-friendly interface to simulate federated training and display results.
* Properly documented source code for all developed components.

**References**:

<https://medium.com/@gaurigst1970/a-practical-introduction-to-tree-based-federated-learning-decision-trees-and-socket-programming-in-eafd95e4945c>

**Appendix:**

|  |  |  |
| --- | --- | --- |
| **Part** | **Output** | **Sub-outputs** |
| **Subjective** | HPI  (OLDCARTS: Onset, Location, Duration, Characterization, Alleviating and Aggravating factors, Radiation, Temporal Factor, Severity) | Onset: When did the CC begin? |
| Location: Where is the CC located? |
| Duration: How long has the CC been going on for? |
| Characterization: How does the patient describe the CC? |
| Alleviating and Aggravating factors: What makes the CC better? Worse? |
| Radiation: Does the CC move or stay in one location? |
| Temporal factor: Is the CC worse (or better) at a certain time of the day? |
| Severity: Using a scale of 1 to 10, 1 being the least, 10 being the worst, how does the patient rate the CC? |
| Medical History |  |
| Surgery History |  |
| Procedure History |  |
| Family History |  |
| Social History |  |
| Medications |  |
| Allergies |  |
| [Review of Systems  No fill: NCBI Colour: acc.org link or CMS (Centre for Medicare and Medicaid Services)](file:///C:\Users\shaheer.ahmed\Downloads\Now\Doctor%20Agent\Agents%20Definition(AutoRecovered).xlsx#'ROS Guidelines'!A1) | [General: Weight loss, decreased appetite](https://www.ncbi.nlm.nih.gov/books/NBK482263/) |
| Gastrointestinal: Abdominal pain, hematochezia |
| Musculoskeletal: Toe pain, decreased right shoulder range of motion |
| [Constitutional symptoms (i.e. fever, weight loss, vital signs)](https://www.acc.org/Tools-and-Practice-Support/Practice-Solutions/Coding-and-Reimbursement/Documentation/Evaluation-and-Management/Review-of-Systems) |
| Eyes |
| Ears, nose, mouth, throat |
| Cardiovascular |
| Respiratory |
| Gastrointestinal |
| Genitourinary |
| Musculoskeletal |
| Integumentary |
| Neurological |
| Psychiatric |
| Endocrine |
| Hematologic/Lymphatic |
| Allergic/Immunologic |
| OBGYN History (In case of females) |  |
| **Objective** | Vital signs |  |
| Physical exam findings |  |
| Laboratory data |  |
| Imaging results |  |
| Other diagnostic data |  |
| Recognition and review of the documentation of other clinicians. |  |
| **Assessment** | [Problem](https://www.ncbi.nlm.nih.gov/books/NBK482263/) |  |
| Differential Diagnosis |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **Plan** | State which testing is needed and the rationale for choosing each test to resolve diagnostic ambiguities; ideally what the next step would be if positive or negative |  |
| Therapy needed (medications) |  |
| Specialist referral(s) or consults |  |
| Prescriptions |  |
| Patient education, counseling |  |