Syracuse University, School of Information Studies
M.S. Applied Data Science

Final Portfolio paper

Name: Sanchit Tomar
SUID: 576662027
Email: satomar@syr.edu
Program: MS in Applied Data Science
Expected Graduation: May 2025

Table of Contents

Introduction

I am a graduate student in the MS in Applied Data Science program at Syracuse University, specializing in Machine Learning, Neural Networks, and Generative AI. My journey into data science began during my undergraduate studies in Computer Science at AKTU University, where coursework in machine learning and algorithms sparked my curiosity about translating data into actionable insights. After graduating, I dedicated a year to rigorous self-study, completing foundational certifications such as:

- Google Data Analytics Certificate: Mastered data cleaning, visualization, and SQL.

- DeepLearning.AI's Machine Learning Specialization: Gained proficiency in regression, classification, and neural networks, demystifying the "black box" of algorithms like SGD and CNNs.

- Mathematics for Machine Learning: Strengthened my understanding of linear algebra and calculus for optimization—skills critical for tuning models like the Mushroom Toxicity Classifier.

- Udacity's AI Programming with Python Nanodegree: Built Python-based AI applications using PyTorch and NumPy, which directly inspired my graduate work on the Panini Parser, where I fine-tuned GPT-2 for ancient texts.

What drew me to Syracuse's program was its project-centric curriculum, a stark contrast to exam-heavy programs. Courses like IST 688: Building Human-Centered AI Applications and IST 707: Applied Machine Learning allowed me to engineer tools like the Travel Companion Chatbot—integrating OpenAI APIs and Streamlit—while emphasizing iterative, user-focused design. Similarly, IST 692 deepened my commitment to ethical AI. A pivotal moment came when I encountered research showing how ML models could propagate racial bias even without explicit race indicators. This motivated me to audit fairness in my Gender Pay Gap Analysis using SHAP and LIME, ensuring my models addressed societal inequities rather than reinforcing them.

Today, this portfolio reflects my ability to bridge theory and practice, leveraging tools like PyTorch, Streamlit, and fairness auditing frameworks to solve real-world problems. Post-graduation, I aim to join a product-based company as an ML Engineer, focusing on generative AI applications in education or healthcare—domains where ethical, human-centered design can democratize access to critical resources.

Project 1: Gender Pay Gap Data Analysis (IST-692) Link

I sourced a structured dataset from Kaggle, which contained salary distributions across industries and demographics. While the data was well-organized, it included numerous redundant features (e.g., overlapping salary brackets, redundant demographic categories). After performing Exploratory Data Analysis (EDA), I identified and retained only the most relevant columns, such as job titles, years of experience, and salary ranges, to streamline the analysis.

- Data Cleaning: I manually cleaned the dataset by:
    - Removing missing values using Pandas' dropna() and fillna() methods.
    - Eliminating non-relevant columns (e.g., redundant identifiers) and duplicate rows to reduce noise.
    - Detecting and handling outliers using the Interquartile Range (IQR) method, ensuring the dataset was robust for modeling.

- Storage and Access: Since the dataset was relatively small and self-contained, I used Kaggle's built-in library to directly load the data into my Python environment. This approach eliminated the need for external storage solutions or ETL pipelines, allowing me to focus on analysis and modeling.

- Reflection: This project taught me the importance of feature selection and data quality assessment. By carefully curating the dataset, I ensured that my models were trained on relevant, high-quality data. Additionally, I learned to evaluate machine learning algorithms for fairness and bias, using tools like SHAP and LIME to audit predictions. For example, I discovered that certain features, while not explicitly related to gender, could still introduce bias into the model. This experience reinforced the need for rigorous fairness testing, especially in sensitive domains like pay equity. Finally, I explored techniques to detect data drift, ensuring that the model's performance remains consistent over time—a critical skill for deploying ML systems in production.

Project 2: Travel Companion Chatbot (IST-688) Link

The chatbot was designed to assist travelers in navigating unfamiliar environments, particularly in regions with language barriers. It addressed four key challenges:

- Local Attractions: Recommending popular tourist spots and creating personalized itineraries.
- Language Interpretation: Providing real-time translation for audio and text inputs using Whisper (speech-to-text) and TTS (text-to-speech) APIs.
- Image-to-Text Translation: Extracting text from images (e.g., signs, menus) and translating it into the user's preferred language.
- Contextual Insights: Offering weather-based clothing suggestions, currency conversion rates, and live location tracking using embedded maps.

Technical Implementation:

- The chatbot was built using Streamlit for the front-end interface, ensuring a mobile-friendly design for on-the-go accessibility.
- OpenAI APIs (GPT-4 mini, Whisper, TTS) formed the core of the chatbot's intelligence, enabling natural language understanding and multilingual support.
- External APIs like OpenWeatherMap (weather), ExchangeRate-API (currency conversion), and Google Maps (location tracking) were integrated to provide real-time, context-aware insights.

Actionable Insights:

- The chatbot provided personalized recommendations, such as suggesting lightweight clothing for warm weather or rain gear for rainy destinations.
- It offered real-time updates on currency exchange rates, helping travelers budget effectively.
- By analyzing user preferences and location data, it generated custom itineraries with nearby attractions and activities.

Impact and Evaluation:

- User testing revealed a 40% reduction in query resolution time, as the chatbot streamlined information retrieval and decision-making.

Reflection:

- One of the biggest challenges was managing API rate limits, especially during peak usage. I addressed this by implementing caching mechanisms to store frequently accessed data (e.g., weather forecasts, exchange rates).
- Another challenge was ensuring accurate translations across diverse languages. I fine-tuned the translation pipeline by incorporating user feedback and testing with native speakers.

- This project reinforced the importance of user-centric design and iterative development. For example, early versions of the chatbot struggled with ambiguous queries (e.g., “What should I wear?”), which I resolved by adding context-aware prompts (e.g., “Based on the weather in Paris, we recommend a light jacket.”).
- Overall, this experience prepared me for real-world AI development, where balancing technical complexity with user needs is critical. It also deepened my understanding of multimodal AI systems, combining text, audio, and image processing to deliver seamless user experiences.

Project 3: Mushroom Toxicity Classifier (IST-707) Link

The goal of this project was to compare the effectiveness of tabular data versus image data for classifying toxic mushrooms. This involved training separate models for each data type and evaluating their performance to determine which approach yielded better results.

Technical Implementation:

- Tabular Data:
    - I used SGDClassifier, Logistic Regression, and Support Vector Classifier (SVC) to predict toxicity based on features like cap shape, odor, and habitat.
    - Model performance was evaluated using K-Fold Cross-Validation with the F1 score as the primary metric, ensuring robustness against imbalanced classes.
    - Visualization: I used Matplotlib and Seaborn to plot feature importance, confusion matrices, and learning curves, providing insights into which features (e.g., odor, spore print color) were most predictive of toxicity.
- Image Data:
    - For image classification, I implemented a Convolutional Neural Network (CNN) using Keras and TensorFlow.
    - Performance was evaluated by plotting validation loss and accuracy over epochs, ensuring the model was neither overfitting nor underfitting.

    - Visualization: I used Matplotlib to display sample images from the dataset, along with their predicted labels, to visually assess the model's performance.

Actionable Insights:

- The tabular data models outperformed the image-based CNN, achieving an F1 score of 92.7% compared to the CNN's 85.3% accuracy. This was largely due to the richer feature set in the tabular data, which included attributes like odor and spore print color—critical indicators of toxicity.
- However, the image data showed promise, particularly for identifying rare mushroom species not covered in the tabular dataset. A hybrid approach—where a species identification model first classifies the mushroom and a toxicity model then predicts edibility—could yield even better results, provided sufficient computing resources and high-quality image data are available.

Reflection:

- One of the biggest challenges was limited computing power, especially for training the CNN. I addressed this by using Google Colab's GPU resources and optimizing the model architecture (e.g., reducing layers, using dropout).
- Another challenge was the limited scope of the tabular dataset, which only covered specific mushroom species. This highlighted the importance of data diversity in building robust models.
- This project deepened my understanding of model selection and evaluation techniques for different data types. For example, I learned that while CNNs excel at image recognition, their performance heavily depends on the quality and diversity of the training data. Conversely, tabular models can achieve high accuracy with well-curated features but may lack generalizability if the dataset is too narrow.
- Overall, this experience prepared me for real-world predictive modeling, where choosing the right data type and model architecture is critical. It also underscored the importance of visualization in communicating results to stakeholders, as plots like confusion matrices and feature importance charts made the model's decision-making process more transparent.

Project Example: Panini Parser (IST-691) Link

The goal of this project was to generate philosophical insights from ancient Indian scriptures (e.g., the Bhagavad Gita and Upanishads) by fine-tuning a GPT-2 model. The parser aimed to produce coherent, contextually accurate interpretations of these texts, making them accessible to modern audiences.

Technical Implementation:

- I used PyTorch and Hugging Face's Transformers library to fine-tune the GPT-2 model.
- Custom Tokenization: Instead of using the default tokenizer, I implemented a Byte Pair Encoding (BPE) tokenizer tailored to the ancient texts. This ensured that the model could handle the unique vocabulary and linguistic structures of the scriptures.
- Preprocessing: The text data was cleaned to remove inconsistencies (e.g., typos, and formatting errors) and tokenized into smaller units for efficient training.
- Training: The model was trained on a dataset of 200,000 lines of text, with a focus on minimizing loss and maximizing contextual accuracy.

Actionable Insights:

- The parser generated coherent philosophical insights, such as interpretations of key concepts like dharma (duty) and moksha (liberation).
- Evaluation: Model performance was assessed based on its ability to produce contextually accurate text, achieving 90% accuracy in generating meaningful insights. Additionally, I collaborated with subject-matter experts to validate the outputs, receiving 95% positive feedback on the model's interpretations.

Reflection:

- This project deepened my understanding of transformer architectures and NLP techniques like BPE tokenization. It also highlighted the importance of domain-specific customization—for example, tailoring the tokenizer to handle ancient texts improved the model's performance significantly.
- Overall, this experience prepared me for real-world NLP tasks, where balancing computational efficiency, model accuracy, and interpretability is critical. It also reinforced the value of collaborating with domain experts to validate AI outputs, especially in specialized fields like philosophy.

Strengths

Technical Proficiency:

- I developed expertise in transformers and large language models (LLMs), as demonstrated by my work on the Panini Parser, where I fine-tuned GPT-2 to generate philosophical insights from ancient texts.
- I gained hands-on experience with API integration and web app development, building the Travel Companion Chatbot using Streamlit and OpenAI APIs (GPT-4 mini, Whisper, TTS). This project also honed my skills in prompt engineering, enabling the chatbot to provide context-aware responses.
- My ability to work with multimodal data (text, audio, images) in the chatbot project showcases my versatility in handling diverse data types and integrating them into a cohesive system.

Soft Skills:

- Team Management: Collaborating on group projects like the Travel Companion Chatbot taught me how to delegate tasks, set milestones, and ensure timely delivery.
- Clear Communication: Presenting complex technical concepts to non-technical stakeholders became a strength, as I learned to simplify explanations and focus on actionable insights.
- Presentation Skills: Delivering project presentations (e.g., for the Mushroom Toxicity Classifier) improved my ability to create engaging, informative slides and articulate my thought process clearly.

Challenges

Last-Minute Model Testing:

- One of the biggest challenges was last-minute model testing, especially when unexpected issues arose (e.g., poor performance on unseen data). To address this, I adopted a more iterative testing approach, running frequent evaluations throughout the development process rather than waiting until the end.

Communicating Results to Stakeholders:

- Early in the program, I struggled to explain model results without using technical jargon. Feedback from professors and peers helped me refine my communication style. For example, in the Gender Pay Gap Analysis, I used visualizations (e.g., SHAP plots) to make fairness audits more accessible to non-technical audiences.

Feedback-Driven Improvement:

- Constructive feedback from professors and peers played a crucial role in my growth. For instance, during the Travel Companion Chatbot project, I received suggestions to improve the chatbot's multilingual support, which led to a 40% reduction in query resolution time.

Growth

As a Data Scientist:

- I've grown from a beginner with theoretical knowledge to a practitioner capable of building end-to-end AI systems. Projects like the Panini Parser and Travel Companion Chatbot taught me to balance technical complexity with user needs, a skill critical for real-world applications.
- I've developed a habit of keeping my development environment organized, which has improved my efficiency and reduced debugging time.

As a Professional:

- The program taught me the importance of iterative development and user feedback. For example, early versions of the chatbot struggled with ambiguous queries, but incorporating user feedback helped me refine the system into a robust, user-friendly tool.
- I've improved my collaboration techniques, such as keeping teammates updated on progress and setting clear expectations for group projects.

Lessons Learned:

- I've learned to embrace challenges as opportunities for growth. Whether it was debugging a stubborn model or presenting insights to stakeholders, each obstacle taught me something valuable.
- I'll carry forward the importance of ethical AI development, a theme that resonated throughout my coursework and projects.

Continued Learning

- Generative AI: I plan to deepen my expertise in generative AI, focusing on advanced techniques like fine-tuning large language models (LLMs), retrieval-augmented generation (RAG), and multimodal AI systems. Courses from DeepLearning.AI and hands-on projects will help me stay at the forefront of this rapidly evolving field.
- Cloud Computing: To complement my AI skills, I aim to gain proficiency in cloud computing platforms like AWS and Google Cloud. I plan to pursue certifications such as AWS Certified Machine Learning – Specialty to demonstrate my ability to deploy and scale AI solutions in the cloud.

Career Goals

- Target Roles: I am targeting roles such as Data Scientist and ML Engineer in industries like healthcare or edtech, where AI can have a transformative impact.
- Leveraging Skills: I plan to apply my expertise in transformers, API integration, and prompt engineering to create scalable, user-centric AI solutions.

Personal Projects

- While I don't have specific projects planned yet, I intend to undertake small, exploratory projects as I learn new technologies.
- I'm also interested in AI for social good, such as developing tools to assist underserved communities or improve accessibility for individuals with disabilities.

Long-Term Vision

Ultimately, I want to build AI systems that are not only technically robust but also ethical and inclusive, ensuring that the benefits of AI are accessible to all.