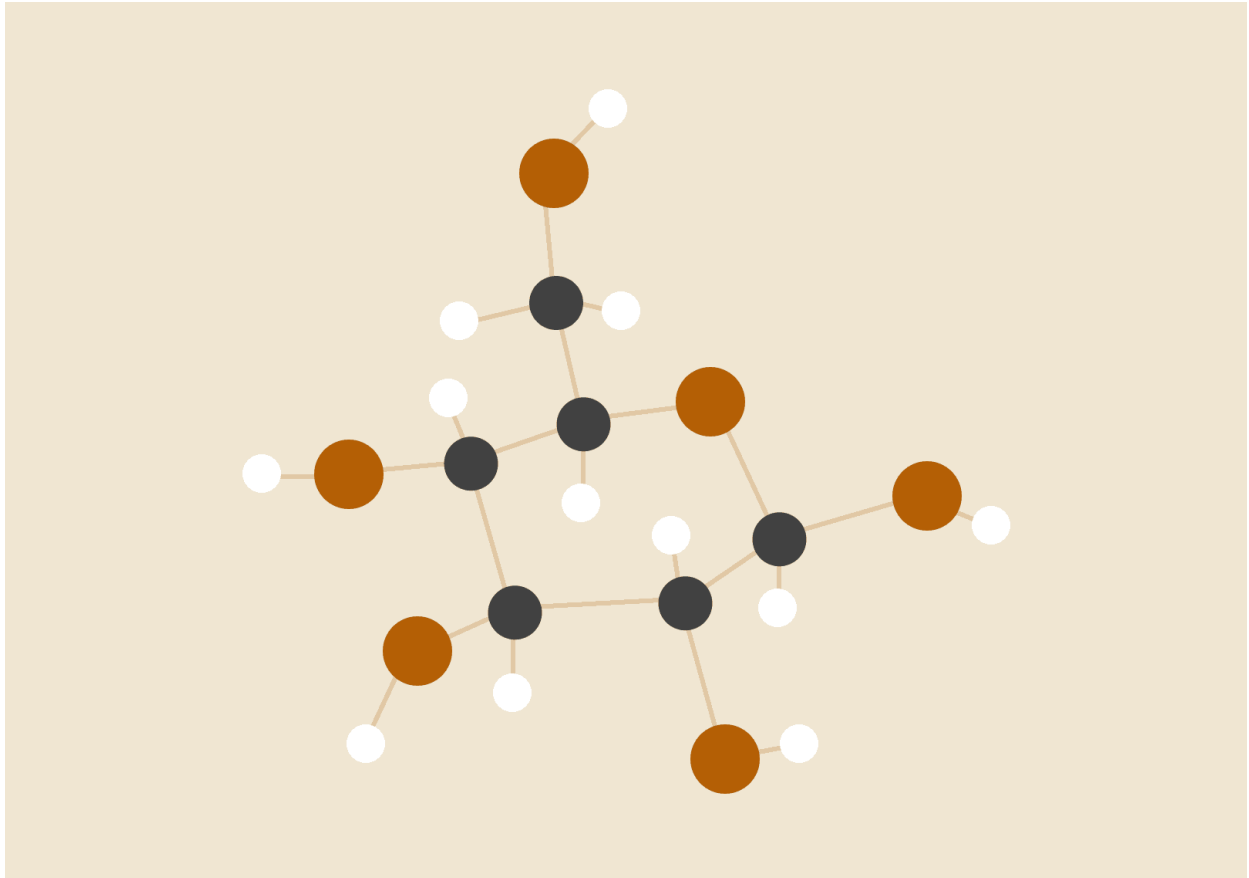


Summary of AI Projects



Athiyo Chakma

2022118

B.Tech (CSE)

Indraprastha Institute of Information Technology Delhi

Project 1: Classical Search Algorithms & Game Tree Reasoning

Topics: A*, UCS, IDA*, IDS, Bidirectional BFS, Bidirectional A*, Minimax, Alpha-Beta Pruning

Skills: Python, Graph Search, Heuristics, Game Theory, Performance Analysis

- **Implemented classical AI search algorithms** including A*, UCS, Iterative Deepening A*, IDS, Bidirectional BFS, and Bidirectional A* using custom test cases and a real-world map of IIT Delhi.
- **Compared search strategies** on time, memory, and path optimality using real data:
 - A* and UCS found shortest paths with lower memory.
 - IDS was slower but more memory-efficient.
 - Bidirectional search excelled in execution time.
- Modeled **game trees using Minimax and Alpha-Beta Pruning** with best- and worst-case arrangements and justified the theoretical complexity $O(b^{\frac{d}{2}})$.
- Developed scatter plots and tables to illustrate empirical performance across algorithms.

Project 2: Knowledge Representation & Logical Reasoning

Topics: FOL, PL, Satisfiability, Graph Coloring, PyDatalog, Rule-Based Inference

Skills: Logic Programming, Predicate Design, PyDatalog, Forward/Backward Chaining

- **Formalized traffic systems and colored graph constraints** using First-Order Logic, including predicates like $\text{Edge}(n1, n2)$, $\text{Color}(n, c)$, $\text{Distance}(n1, n2, d)$, etc.
- Represented complex logical relationships using **Propositional & First-Order Logic** and resolved them using **resolution refutation**.

- Implemented **brute-force and PyDatalog-based logical inference** for Delhi bus transit routes using GTFS data. Inferred direct connectivity using:
 - Procedural (brute-force)
 - Declarative (PyDatalog logic rules)
- Developed a **constrained planner** for optimal routes (via stop + 1 interchange only) using **Forward/Backward Chaining** and **PDDL**.
- Extended planning to fare-constrained paths using advanced rules and optimized reasoning with **pruning techniques**.

Project 3: Bayesian Networks & Hidden Markov Models (HMMs)

Topics: Sampling, Bayesian Inference, CPTs, Roomba Tracking, Viterbi Algorithm

Skills: bnlearn, HMMs, Viterbi, Probabilistic Reasoning, Conditional Independence

- Built a **Bayesian Network classifier** for fare prediction using GTFS data (features: stop IDs, distance, zones, route type).
 - Constructed, pruned, and optimized three network versions.
 - Visualized all models and compared their accuracy and training runtime.
- Explored **sampling techniques**: Direct, Rejection, and Gibbs Sampling. Applied them to real-world travel behavior datasets for empirical probability estimation.
- Investigated **adversarial ML scenarios** using Bayesian updates, modeling $P(\text{misclassification} \mid \text{adversarial/backdoor})$ and applying the “explaining away” effect.
- Developed a **Hidden Markov Model to track a Roomba’s true path** using noisy observations.
 - Implemented the **Viterbi algorithm** to recover the most probable

sequence.

- Compared tracking accuracy across three seed environments and movement policies.

Project 4: Decision Trees for Real Estate Price Prediction

Topics: Decision Trees, EDA, Pruning, Imbalanced Data, Feature Importance

Skills: Scikit-learn, Tree Visualization, SMOTE, ADASYN, Random Forests, Regression Metrics

- Trained a **Decision Tree Regressor** on real estate data to predict property prices.
 - Conducted **EDA, scaling, encoding, binning**, and handled **class imbalance** using oversampling/undersampling.
- Evaluated model via **MSE, MAE, R²**, achieving ~99.3% accuracy on both training and test sets.
- Visualized tree structure, extracted and analyzed **feature importances** (e.g., carpet area, price/sqft).
- Applied **hyperparameter tuning** (max_depth, min_samples_split, etc.) and **cost-complexity pruning**.
- Performed **cross-validation and residual analysis** to detect overfitting.
- Compared with **Random Forest Regressor** and explored **ADASYN vs SMOTE** for imbalance correction.

Key Skills Demonstrated Across Projects:

- **AI Algorithms:** A*, UCS, IDA*, IDS, Minimax, Alpha-Beta, Viterbi
- **Reasoning & Planning:** FOL, PL, PyDatalog, Rule-Based Systems, PDDL

- **Probabilistic Models:** Bayesian Nets, HMMs, Joint Distributions, Inference
- **ML Techniques:** Decision Trees, Random Forests, Feature Engineering, Imbalance Handling
- **Evaluation & Visualization:** Cross-validation, Residuals, Feature Importance, Plotly, Matplotlib