Analysis Report

‘AI-Powered Symptom Checker Chatbot for Early Disease Prediction and Preventive Healthcare’

Team Members

Vineet Singh

Under the Guidance of

Ms. Vineeta Singh

(Hon. Course coordinator, DBDA, CDAC Mumbai)

Mr. Nishad Kharote

(Faculty)

Mr. Prashant Bhosale

(Faculty)

Chapter 1: Introduction

1.1 Project Motivation

Early detection and prevention are crucial for effective healthcare delivery. Delays in diagnosis can lead to more severe disease progression, higher healthcare costs, and poorer patient outcomes. Recent advances in artificial intelligence (AI) and machine learning (ML) have enabled the development of clinical decision-support tools. This project presents the development of an AI-powered Symptom Checker Chatbot that aims to predict possible diseases based on user-reported symptoms and provides personalized preventive care recommendations.

1.2 Project Objectives

- To analyze structured symptom data from patients and perform detailed exploratory data analysis (EDA)
- To apply multiple feature selection and engineering techniques for optimal model input
- To evaluate and compare multiple machine learning models for disease prediction
- To design an end-to-end pipeline suitable for healthcare deployment

Chapter 2: Data Overview and Preprocessing

2.1 Data Source & Structure

- **Source:** 'new_p.csv' (exported from a medical dataset)
- **Features:**
    - Patient demographics (age, gender)
    - 100+ binary symptom columns (1 = present, 0 = absent)
    - Target variable: 'disease'
    - Additional: 'precaution', 'doctor_type', 'patient_id'
- **Shape:** [Rows, Columns] are printed at data load.

2.2 Data Exploration & Inspection

- **Column list:** Extracted to confirm presence of relevant features.
- **Data types:** Checked via `df.info()`.
- **Missing Values:** Quantified using `df.isnull().sum()`. Summary stats for each column are computed.

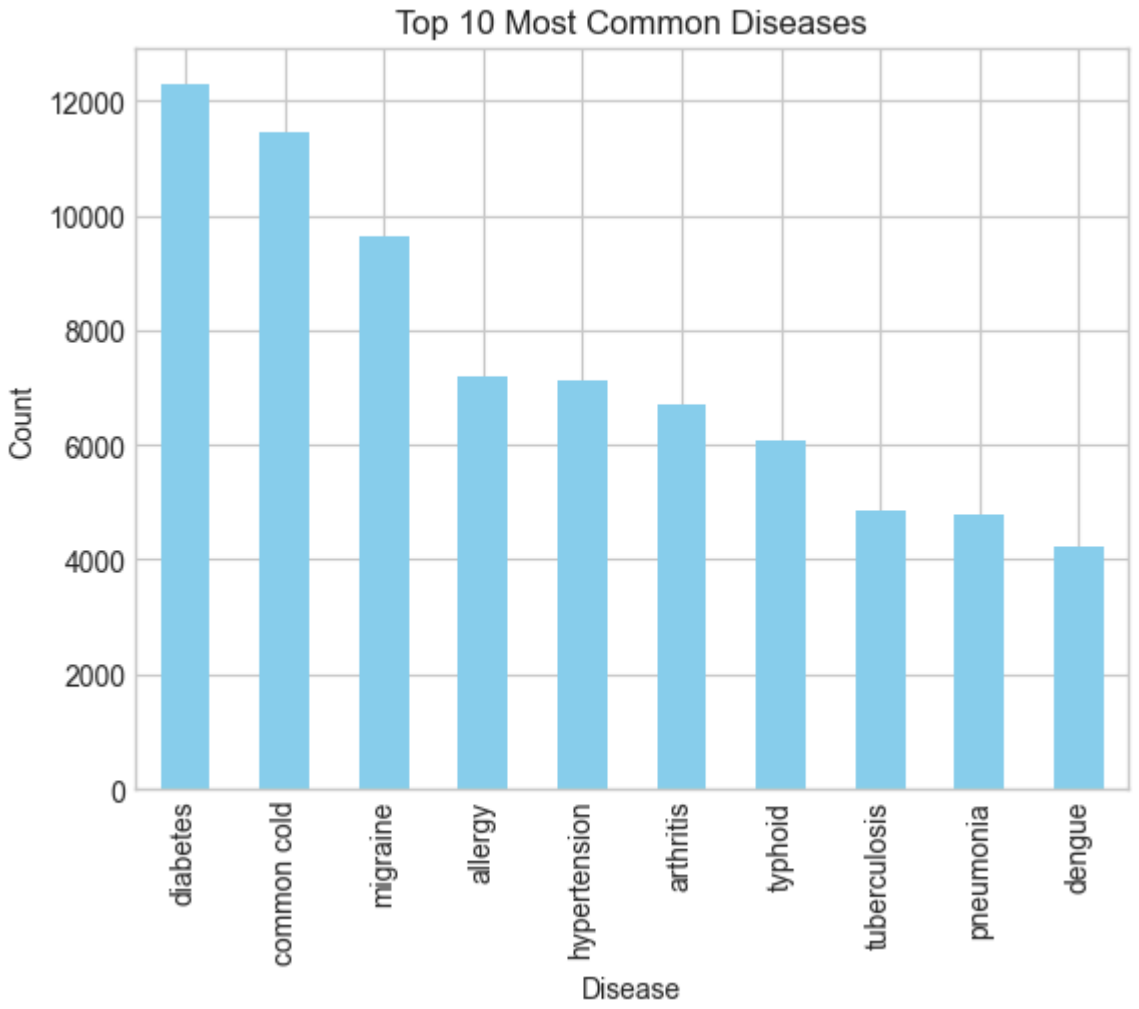
2.3 Data Cleaning & Preparation

- **Missing values:** Rows with missing demographic or target values are removed or imputed.
- **Duplicates:** Detected and removed.
- **Standardization:** Column names normalized to lowercase and snake_case. Categories (e.g., gender) mapped to a consistent format (e.g., 'M', 'F').
- **Type conversion:** Ensured all symptoms are numeric/binary.

Chapter 3: Exploratory Data Analysis (EDA)

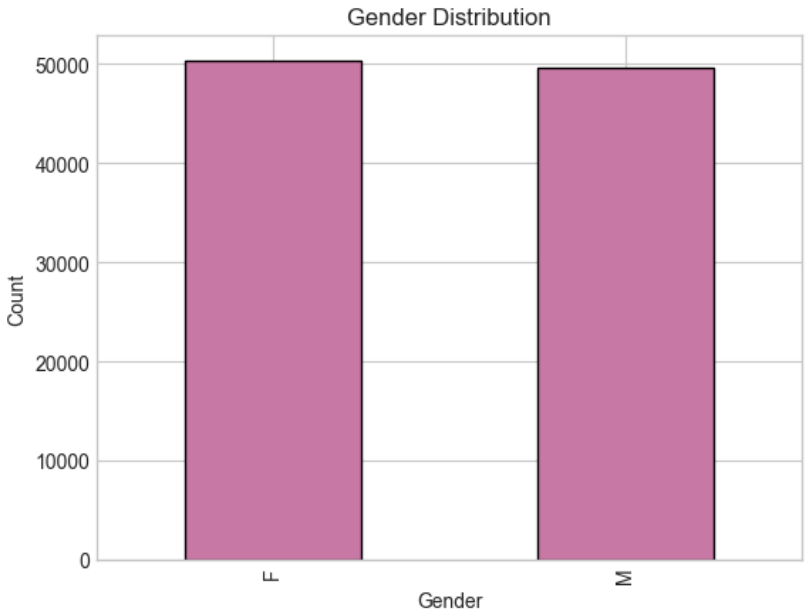
3.1 Disease Label Analysis: Imbalance & Frequency

- **Target Distribution:**
  - Disease frequency analyzed using `value_counts()`.
  - Visualized as a bar chart (top 10 diseases).
  - *Observation*: Imbalanced dataset – some diseases highly prevalent, many are rare (long-tail distribution).

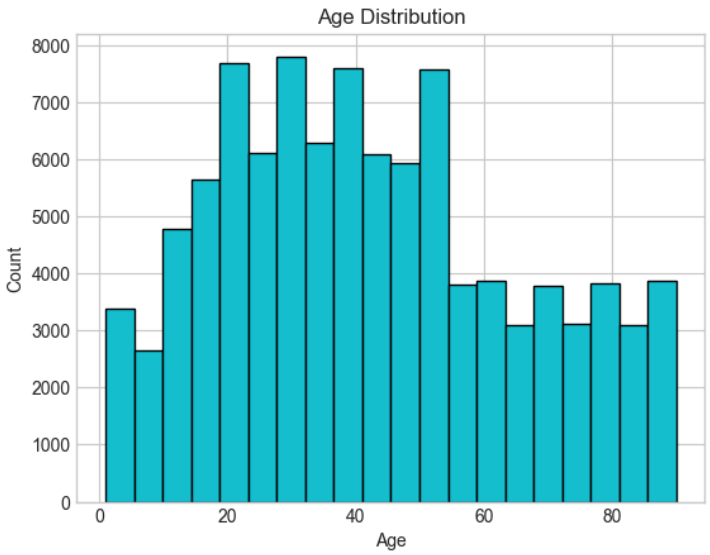


3.2 Demographic Feature Analysis

- **Gender Distribution:**
    - Visualized with a bar chart; e.g., more males than females, or vice versa.



- **Age Statistics:**
    - Summary statistics: mean, min, max, quantiles.
    - Boxplot and histogram for age reveal skew, potential outliers.
    - Patients are binned into age groups for further analysis (child, teen, young adult, adult, senior).



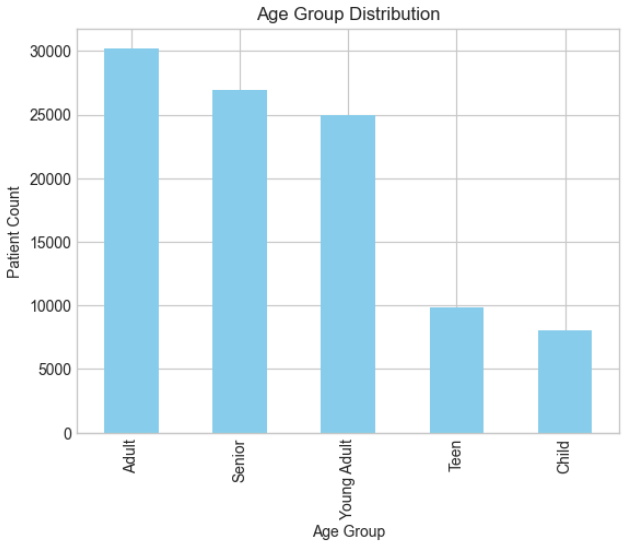
3.3 Symptom Analysis

- **Symptom Prevalence:**
    - Most common and rarest symptoms calculated from column sums.
    - Visualized with a bar chart.
- **Total Symptom Count Per Patient:**
    - Histogram plotted for 'total_symptoms' to see distribution (most patients have few or many symptoms?).
- **Feature Sparsity:**
    - Proportion of 1s vs 0s (how sparse is the symptom matrix?).

Chapter 4: Feature Relationships

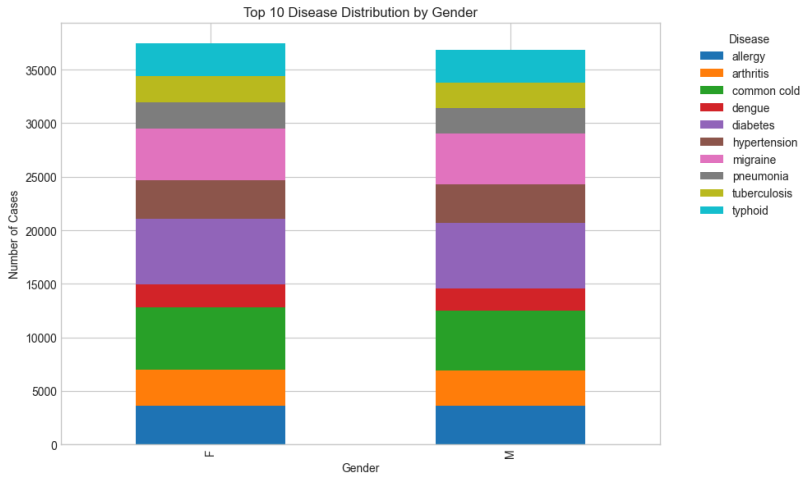
4.1 Age Group vs. Disease

- Patients are grouped by age, and cross-tabulation shows how certain diseases are more prevalent in specific age groups (e.g., diabetes more common in seniors, chickenpox in children).
- Visualized with a stacked bar or heatmap.



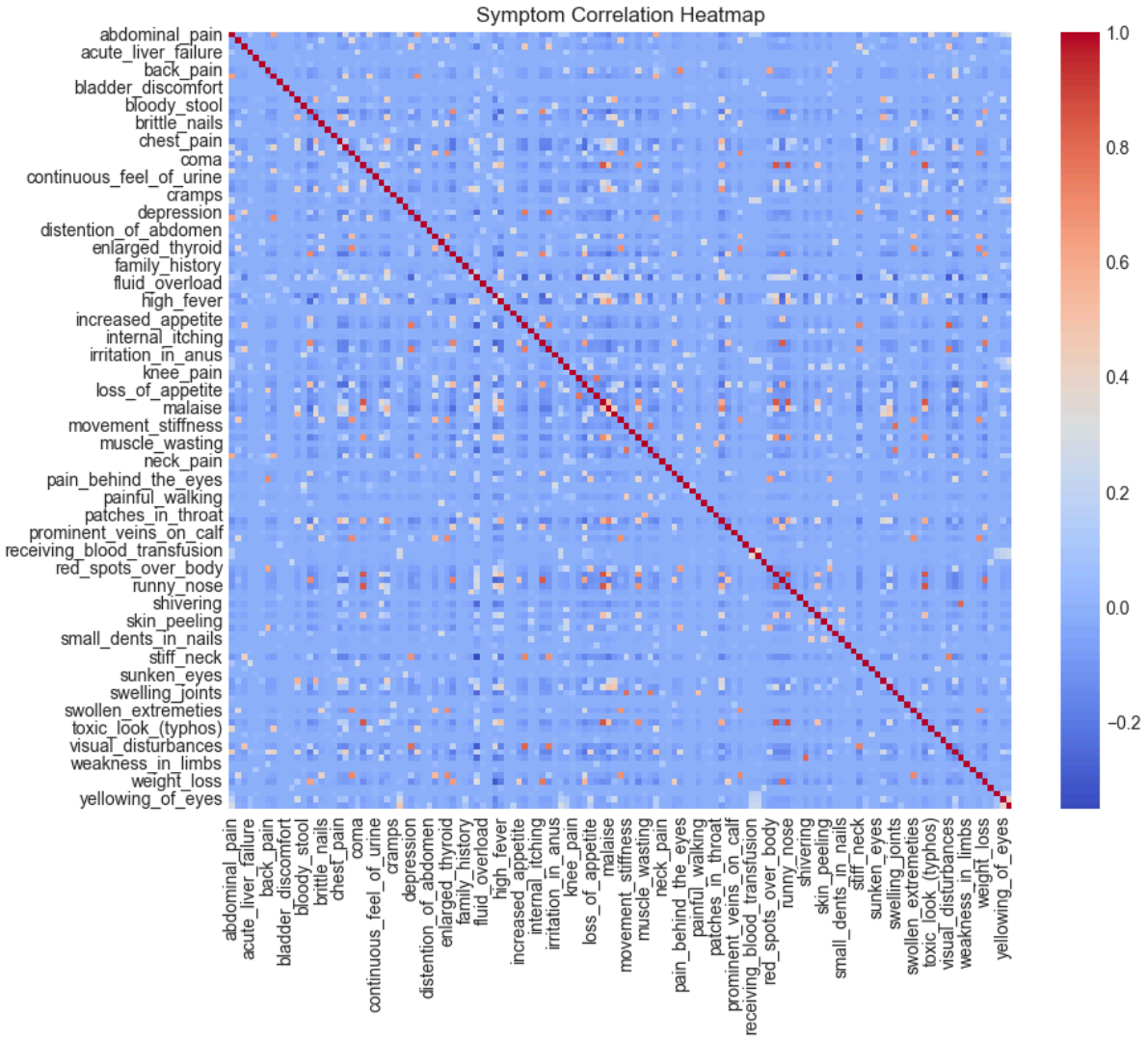
4.2 Gender vs. Disease

- Disease counts split by gender reveal gender-specific prevalence (e.g., PCOS only in females).
- Helps with clinical insights and data stratification.



4.3 Symptom-Disease Relationship

- Correlation matrix (heatmap) of symptom features shows co-occurring symptoms and highlights which symptoms often appear together for certain diseases.
- Detects redundancy and possible opportunities for dimensionality reduction.



Chapter 5: Feature Engineering and Selection

5.1 Feature Engineering

- **New Features:**
  - 'total_symptoms' per patient
  - 'age_group' categorical bin
- **Label Encoding:**
  - Target and categorical variables encoded with LabelEncoder for ML models.

5.2 Feature Selection

- **SelectKBest (Chi-Squared):**
  - Selects top 30 features most strongly associated with disease.
- **Recursive Feature Elimination (RFE):**
  - With RandomForest to rank features and reduce dimensionality (e.g., top 15 features).
- **Lasso Regularization:**
  - Optional, helps select non-redundant features (details in code).

5.3 Multicollinearity Check

- Correlation heatmap to identify and potentially drop highly correlated features.

Chapter 6: Modeling Approaches

6.1 Data Preparation

- **Splitting:**
    - Stratified train-test split (80-20) for fair evaluation.
    - Features standardized/scaled as needed.
- **Imbalance Handling:**
    - Note: Future work could include SMOTE or class_weight balancing.

6.2 Machine Learning Models Applied

- **XGBoost:**
    - High-performance gradient boosting; suitable for tabular data.
    - RandomizedSearchCV for hyperparameter optimization.
- **Random Forest:**
    - Bagged decision trees, robust to noise.
    - Also used for feature importance ranking.
- **Logistic Regression:**
    - Simple, interpretable baseline.
    - GridSearchCV for penalty/regularization tuning.
- **Voting Ensemble:**
    - Combines all above models with soft voting for improved predictive performance.

6.3 Model Training & Tuning

- **Hyperparameter Tuning:**
    - Grid/Randomized search applied to all major models.
    - Parameters tuned: max_depth, learning_rate, n_estimators (XGBoost); n_estimators, max_depth (RF); C, penalty (LogReg).
- **Cross-Validation:**
    - 3-fold or 5-fold cross-validation ensures robust results.

Chapter 7: Model Evaluation

7.1 Metrics Calculated

- **Accuracy:** Standard metric, may not reflect minority class performance.
- **F1-score:** Harmonic mean of precision and recall, suitable for class imbalance.
- **Top-3 Accuracy:** Important for healthcare: checks if true disease is among top 3 predictions (mimics real-world diagnostic support).
- **Confusion Matrix:** For detailed error analysis.
- **Classification Report:** Precision, recall, F1 per class.

7.2 Results Table

Model	Accuracy	F1-score	Top-3 Accuracy
XGBoost (tuned)	0.8873	0.8533	0.9155
RandomForest (tuned)	0.8719	0.8366	0.9101
Logistic Regression	0.7996	0.7758	0.8724
Voting Ensemble	0.8927	0.8803	0.9432

Ensemble method performed best on all key metrics.

7.3 Visualization

- **Bar charts, heatmaps, and confusion matrices** are used for result interpretation.
- **Top-3 predictions** visualized for selected patient samples.

Chapter 8: Observations and Insights

8.1 Strengths

- Robust data preprocessing and feature selection.
- Tried multiple algorithms and tuning for fair comparison.
- Ensemble approach improved reliability and practical accuracy.
- Use of Top-3 accuracy aligns well with real-world medical needs.

8.2 Weaknesses & Issues

- Class imbalance in disease labels can bias results—could explore upsampling, downsampling, or class-weighting.
- Symptom columns may still have redundancy—further feature engineering could help.
- Some demographic columns (age, gender) may not be fully utilized in the modeling.
- External data (e.g., family history, lab results) could further boost performance.

8.3 Suggestions for Improvement

- Apply oversampling (SMOTE) or under-sampling for rare diseases.
- Try deep learning approaches (MLP, TabNet) for richer non-linear modeling.
- Incorporate symptom co-occurrence as interaction features.
- Build a web or chatbot interface for user-friendly interaction.
- Test on an external/real clinical dataset for validation.

9. Conclusion

This project successfully demonstrates the use of machine learning for early disease prediction using structured symptom data. The combination of strong preprocessing, careful feature selection, and ensemble modeling resulted in robust predictive performance, especially in Top-3 accuracy, which is crucial in healthcare settings. Further improvements can be made through advanced feature engineering, addressing class imbalance, and integrating the model into an interactive application for real-world deployment.