**SOFTWARE REQUIREMENT SPECIFICATION**

**MEDICINE RECOMMENDATION SYSTEM**

04-12-2024

PREPARED BY

|  |  |  |
| --- | --- | --- |
| Specialization | SAP ID | Name |
| MCA-AIML | 500120083 | Jayant Dev |
| MCA-AIML | 500120087 | Vinay Kumar Pandey |
| MCA-AIML | 500117998 | Devansh Maheshwari |
| MCA-AIML | 500123144 | Sunil Singh |



School Of Computer Science

UNIVERSITY OF PETROLEUM & ENERGY STUDIES,

DEHRADUN- 248007. Uttarakhand

Table of Contents

|  |  |  |
| --- | --- | --- |
| **Topic** | | **Page No** |
| Table of Content | |  |
| Revision History | |  |
| 1 | Introduction |  |
|  | 1.1 Purpose of the Project |  |
|  | 1.2 Target Beneficiary |  |
|  | 1.3 Project Scope |  |
|  | 1.4 References |  |
| 2 | Project Description |  |
|  | 2.1 Reference Algorithm |  |
|  | 2.2 Data/ Data structure |  |
|  | 2.3 SWOT Analysis |  |
|  | 2.4 Project Features |  |
|  | 2.5 User Classes and Characteristics |  |
|  | 2.6 Design and Implementation Constraints |  |
|  | 2.7 Design diagrams |  |
|  | 2.8 Assumption and Dependencies |  |
| 3 | System Requirements |  |
|  | 3.1 User Interface |  |
|  | 3.2 Software Interface |  |
|  | 3.3 Database Interface |  |
|  | 3.4 Protocols |  |
| 4 | Non-functional Requirements |  |
|  | 4.1 Performance requirements |  |
|  | 4.2 Security requirements |  |
|  | 4.3 Software Quality Attributes |  |
| 5 | Other Requirements |  |
| Appendix A: Glossary | |  |
| Appendix B: Analysis Model | |  |
| Appendix C: Issues List | |  |

**1. INTRODUCTION**

**1.1 Purpose of the Project**

The **Classification Model Evaluation System** provides an automated framework to evaluate the performance of various machine learning models on a dataset. This system addresses the need for an efficient and consistent approach to compare models, focusing on metrics like accuracy and confusion matrices to help users identify the best-suited algorithm for their data.

The system evaluates five popular machine learning algorithms:

1. Support Vector Classifier (SVC)
2. Random Forest Classifier
3. Gradient Boosting Classifier
4. K-Nearest Neighbors (KNN)
5. Multinomial Naive Bayes (MNB)

The motivation behind this project is to simplify model selection processes for researchers, data scientists, and ML practitioners, ensuring informed decisions are based on objective comparisons.

**1.2 Target Beneficiary**

The beneficiaries of this system include:

* **Data Scientists and ML Practitioners:** To quickly compare and evaluate multiple models for various datasets.
* **Academicians and Researchers:** For teaching and exploring machine learning concepts and performance evaluation.
* **Organizations:** Streamline the selection of machine learning models in real-world applications.

**1.3 Project Scope**

The system evaluates and compares machine learning models based on standard performance metrics, providing users with actionable insights for model selection.

* **Benefits:**
  + Consistent model evaluation.
  + Automated generation of performance metrics.
  + User-friendly outputs for decision-making.
* **Objectives:**
  + Load and preprocess datasets.
  + Train five pre-defined models using default or customizable hyperparameters.
  + Compare models using key metrics like accuracy and confusion matrices.
  + Provide clear outputs for interpretation.
* **Deliverables:**
  + Python-based CLI tool for evaluating classification models.
  + Detailed performance reports (accuracy, confusion matrix).

**1.4 References**

* **Documentation:** Scikit-learn library documentation for model and evaluation techniques.
* **Datasets:** UCI Machine Learning Repository, synthetic datasets from sklearn.datasets.
* **Tools:** Python libraries: Scikit-learn, NumPy, Pandas, Matplotlib (optional for visualizations).

**2. PROJECT DESCRIPTION**

**2.1 Reference Algorithm**

The system leverages the following machine learning algorithms:

1. **Support Vector Classifier (SVC):** Uses a linear kernel for creating hyperplanes to separate classes.
2. **Random Forest Classifier:** An ensemble model using multiple decision trees for robust predictions.
3. **Gradient Boosting Classifier:** Sequentially builds models, reducing errors from previous iterations.
4. **K-Nearest Neighbors (KNN):** Classifies based on the nearest neighbors in the dataset.
5. **Multinomial Naive Bayes (MNB):** Probabilistic model suited for text and multi-class data.

Each algorithm is evaluated using structured datasets, with performance metrics highlighting their suitability for various classification problems.

**2.2 Characteristics of Data**

* **Primary Data Source:** Synthetic datasets (sklearn.datasets) or user-provided CSV files.
* **Data Preprocessing:** Includes handling missing values, scaling features, and encoding categorical variables.
* **Sampling Techniques:** Stratified sampling to maintain class balance in training and testing datasets.
* **Statistical Methods:** Data normalization, feature scaling, and exploratory data analysis (EDA).

**2.3 SWOT Analysis**

* **Strengths:** Automates model evaluation, supports popular classifiers, and generates interpretable metrics.
* **Weaknesses:** Limited to default hyperparameters without explicit tuning unless specified by the user.
* **Opportunities:** Can expand to include additional models, hyperparameter tuning, and visualization.
* **Threats:** Dependency on dataset quality; potential issues with scalability for very large datasets.

**2.4 Project Features**

* **Core Features:**
  + Load datasets from CSV files or generate synthetic data.
  + Train multiple machine learning models automatically.
  + Generate performance metrics (accuracy, confusion matrix).
  + Compare models and recommend the best-performing one.

**2.5 User Classes and Characteristics**

* **Beginners:** Users with minimal ML experience; benefit from easy-to-interpret outputs.
* **Data Scientists:** Require detailed metrics for decision-making.
* **Researchers:** Use the tool to analyze model performance on academic datasets.

**2.6 Design and Implementation Constraints**

* **Hardware Requirements:** Modern systems with at least 8GB RAM for efficient model training and testing.
* **Software Requirements:** Python 3.6+ with libraries like Scikit-learn, Pandas, and NumPy.
* **Security:** Focused on local execution; data is not transmitted externally.

**2.7 Design Diagrams**

* **Activity Diagram:** Demonstrates workflow from data loading to model evaluation.
* **Class Diagram:** Describes system components like DatasetHandler, ModelTrainer, and ResultEvaluator.
* **Sequence Diagram:** Illustrates the interaction between system modules during evaluation.

**2.8 Assumptions and Dependencies**

* Assumes data is properly formatted (e.g., no missing labels).
* Relies on Python libraries for machine learning and data manipulation.

**3. SYSTEM REQUIREMENTS**

**3.1 Functional Requirements**

1. **Data Input:**
   * Load datasets from CSV files or generate synthetic data.
   * Split datasets into training and testing subsets.
2. **Model Training:**
   * Train the following classifiers: SVC, RandomForest, GradientBoosting, KNeighbors, MultinomialNB.
3. **Model Evaluation:**
   * Compute accuracy for each model.
   * Generate confusion matrices.
4. **Results Output:**
   * Print model accuracy and confusion matrices.
   * Optionally save results to a log file.

**3.2 Non-Functional Requirements**

* **Performance:** Train models on datasets with up to 100,000 samples efficiently.
* **Scalability:** Support future enhancements (e.g., additional models, metrics).
* **Portability:** Compatible with Python 3.6+ environments.
* **Usability:** Clear command-line interface with minimal configuration.

**4. NON-FUNCTIONAL REQUIREMENTS**

1. **Availability:**
   * The system should be executable whenever the required libraries are installed.
2. **Maintainability:**
   * Modular code structure for easy updates.
3. **Security:**
   * Ensures no tampering with local data during processing.
4. **Reliability:**
   * Consistent results for repeated evaluations with the same dataset.
5. **Usability:**
   * Simple setup; users only need to pass a dataset and run the script.

**5. OTHER REQUIREMENTS**

1. **Data Preprocessing:**
   * Handles missing values and scales features appropriately.
2. **Model Hyperparameters:**
   * Default settings used unless overridden by the user.
3. **Output Storage:**
   * Save results to a log file for review.
4. **Dependencies:**
   * Python libraries: Scikit-learn, NumPy, Pandas.

**Appendices**

**Appendix A: Glossary**

* **Accuracy:** The ratio of correct predictions to total predictions.
* **Confusion Matrix:** Summarizes classification performance (e.g., true positives, false negatives).
* **Hyperparameter:** Model-specific settings like the number of trees in Random Forest.

**Appendix B: Analysis Model**

* Workflow includes:
  + Dataset preparation.
  + Model training and evaluation.
  + Result aggregation and comparison.

**Appendix C: Issues List**

1. Dataset size may slow evaluation for very large datasets.
2. Hyperparameter tuning not included by default.
3. Results interpretation for complex models like Random Forest may require additional tools.