**Ovarian Cancer Detection using**

**Machine** **Learning**

**Introduction**

Ovarian cancer is one of the most challenging cancers to detect early due to its subtle symptoms. This project aims to create an effective machine learning system for predicting the likelihood of ovarian cancer using clinical and biochemical data. A web application was also developed to make predictions user-friendly and accessible.

**Dataset Overview**

The dataset used for this project contains 349 records with 51 attributes, comprising clinical and biochemical markers, demographic data, and a target variable (TYPE) indicating the presence or type of ovarian cancer.

**Key Features:**

* **Clinical markers**: AFP, CA125, HE4.
* **Biochemical markers**: ALB, ALT, AST.
* **Demographics**: Age, Menopause.
* **Target variable**: TYPE (representing cancer categories).

**Target Variable (TYPE):**

The TYPE column is a binary target variable with two possible values:

* **0**: Indicates the individual does not have ovarian cancer (negative case).
* **1**: Indicates the individual has ovarian cancer (positive case).

This binary classification setup allows the model to predict one of two outcomes: "Safe" (0) or "High possibility of cancer" (1). It simplifies the problem into a binary classification task, facilitating the use of well-optimized algorithms.

**Preprocessing:**

* **Cleaning**: Removed whitespace, converted string-based numbers, and handled non-numeric entries.
* **Imputation**: Missing values were replaced with the column mean using SimpleImputer.
* **Feature Selection**: Top 10 features were identified using the ANOVA F-test (SelectKBest).

**Selected Features:**

Age, ALB, CA125, HE4, LYM#, LYM%, Menopause, NEU, PCT, PLT.

**Methodology**

**Machine Learning Workflow:**

1. **Data Splitting**: The dataset was divided into training (80%) and testing (20%) sets.
2. **Model Selection**: A **Random Forest Classifier** was used for its robustness and interpretability.
3. **Evaluation Metrics**:
   * **Accuracy**: Percentage of correct predictions.
   * **Classification Report**: Precision, recall, and F1-score.
   * **Confusion Matrix**: Breakdown of true positives, false positives, etc.

**Web Application**

To make predictions accessible, a web application was developed using Flask:

* Features: A form for entering selected feature values.
* Prediction: Outputs a diagnosis ("High possibility of cancer" or "Safe") along with the confidence score.
* Integration: The trained model was serialized using pickle and integrated into the application.

**User Workflow:**

1. Input clinical and demographic data via the web interface.
2. Receive predictions and probability scores in real-time.

**Results**

* **Model Performance**:
  + **Accuracy**: Achieved 92% accuracy on the test set.
  + **Confusion Matrix**: Highlighted effective differentiation between positive and negative cases.
  + **Important Features**: CA125 and HE4 showed the highest predictive power.
* **Web Application**: Successfully deployed a functional application capable of real-time predictions.

**Challenges and Future Scope**

**Challenges:**

* Handling missing and irregular data entries.
* Ensuring interpretability of the model for medical practitioners.

**Future Scope:**

* Expand the dataset for better generalization.
* Incorporate advanced algorithms like XGBoost or deep learning.
* Add visualization tools for explaining model predictions to end users.

**Conclusion**

This project demonstrates the power of machine learning in healthcare, specifically for early detection of ovarian cancer. By combining robust models with an accessible web interface, the solution has the potential to assist healthcare professionals and improve patient outcomes.

**References**

1. **Dataset**: [<https://www.kaggle.com/datasets/saurabhshahane/predict-ovarian-cancer/code>]
2. **Tools and Libraries**:

* Scikit-learn: Pedregosa et al. (2011). Scikit-learn: Machine Learning in Python. JMLR 12, pp. 2825–2830.
* Flask: Flask Documentation, <https://flask.palletsprojects.com>.
* Pandas: McKinney, W. (2010). *Data Structures for Statistical Computing in Python*. Proceedings of the 9th Python in Science Conference, pp. 51-56.

1. **Model**:

* Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5–32.

