This document consists the Literature Review, Technology Review and Data Review for M Zaker Humayoon Amin and Hamda Abdi's Capstone Project.

Literature Review: Machine Learning Applications in Breast Cancer Diagnosis.

Introduction:

Breast cancer (BC) diagnosis and prognosis have witnessed significant advancements through the integration of machine learning (ML) techniques. This literature review synthesizes findings from two distinct studies [1] [2] focusing on ML applications, particularly in the context of the Wisconsin Breast Cancer Dataset (WBCD).

1. Artificial Neural Networks (ANNs):

Example 1 - Early Applications:

The early 1990s marked the introduction of Artificial Neural Networks (ANNs) for BC analysis. ANNs demonstrated their potential in predicting malignancy, utilizing mammographic elements as inputs. Despite their "black box" nature, ANNs exhibited superior diagnostic performance compared to conventional expert judgment, achieving accuracies improved by 3–5%. Over the years, various ANN architectures and optimization methods have been explored. Hybrid approaches, combining ANNs with association rules (AR), evolutionary algorithms, and pruning techniques, yielded accuracies ranging from 94.36% to 99.68%.

Example 2 - Evolutionary Approaches:

Evolutionary ANN approaches, such as Pareto-different evolution (PDE), showcased improved network performance and architecture. Additionally, the introduction of metaplasticity ANN algorithms and rotation forest ANN (RF-ANN) further diversified the application of ANNs in BC diagnosis. The proposed modifications, such as genetic programming (GP) and deep belief network (DBN) architecture, demonstrated enhanced accuracy, reaching up to 99.59%. The literature emphasizes the balance between interpretability and computational burden, highlighting the continuous need for algorithm development.

2. Support Vector Machines (SVMs):

Example 1 - Basic SVM and Variants:

SVMs have been fundamental in BC classification, with basic SVM achieving accuracies of 97.2%. The introduction of least square SVM (LS-SVM) and feature selection strategies further elevated accuracies to 98.53% and 99.51%. Research has explored various SVM types, including proximal SVM (PSVM), finite Newton method for Lagrangian SVM (NSVM), and smooth SVM (SSVM). The combination of weighted-particle swarm optimization (WPSO) with SSVM reached 98.42%, demonstrating the adaptability of SVMs to different optimization approaches.

Example 2 - Two-Step SVM:

A unique approach involved integrating a two-step clustering algorithm with a probabilistic support vector machine, achieving a classification accuracy of 99.10%. This novel method addressed the identification of

clusters and exhibited efficiency in analyzing extensive datasets. The study emphasizes the significance of diverse SVM types and their strategic combination to enhance predictive performance.

3. Decision Trees (DTs):

Example 1 - Modified C4.5 Decision Tree:

The utilization of decision trees in BC diagnosis dates back to Quinlan's application of a modified C4.5 decision tree. Further innovations, including fuzzy decision trees (FDT) and hybrid models (CBFDT), showcased interpretability and achieved accuracies ranging from 94.56% to 99.9%. Notably, a two-step approach involving clustering and outlier detection achieved a remarkable accuracy of 99.90%.

Example 2 - J48 Decision Tree and Hybrid Models:

The application of the J48 decision tree and hybrid intelligent systems, combining fuzzy min-max ANN, CART, and random forest, demonstrated versatility and obtained accuracies of 94.36% and 98.84%, respectively. These findings emphasize the relevance of decision trees in BC classification, with advancements focusing on feature selection and optimization.

4. k-Nearest Neighbors (k-NNs):

Example 1 - Exploring k-NN Variants:

k-NN algorithms, including fuzzy k-NN and rank k-NN, exhibited sensitivity to the choice of 'k' and distance metrics. The application of Euclidean distance with k = 1 achieved an accuracy of 98.70%. This highlights the importance of parameter selection in k-NN algorithms and the potential impact on classification outcomes.

Example 2 - Distance Metrics and Ranking:

Exploration of various distance metrics, such as Cityblock, Cosine, and Correlation, in k-NN demonstrated varied classification accuracies ranging from 94.69% to 98.83%. The studies underscore the need for meticulous parameter tuning in k-NN algorithms for optimal performance.

Conclusion:

In conclusion, the integration of machine learning techniques in breast cancer diagnosis and prognosis, as evidenced by studies on the Wisconsin Breast Cancer Dataset, has yielded significant advancements. The synthesis of findings from artificial neural networks, support vector machines, decision trees, and k-nearest neighbors emphasizes the diverse strategies employed for improved accuracy. Ongoing research focuses on algorithmic enhancements and the application of novel optimization techniques to further refine the capabilities of machine learning in intelligent healthcare systems.

References:

1. Shetty, Simitha (2020) *Breast Cancer Analysis and Prognosis Using Machine Learning.* Masters thesis, Dublin, National College of Ireland

2. Yue, W., Wang, Z., Chen, H., Payne, A., & Liu, X. (2018). Machine Learning with Applications in Breast Cancer Diagnosis and Prognosis

Technology Review

Breast cancer remains a significant global health concern, necessitating innovative approaches for early detection and accurate prognosis. In recent years, machine learning has emerged as a powerful tool in the realm of healthcare, offering the potential to revolutionize the way we predict and manage breast cancer. This technology review delves into the diverse array of tools and frameworks available for predicting breast cancer using machine learning, aiming to provide a comprehensive understanding of the technological landscape. By exploring programming languages, machine learning libraries, data processing tools, visualization techniques, model deployment strategies, cloud services, and automated machine learning tools, we seek to identify the optimal combination of technologies that can significantly contribute to the advancements in breast cancer prediction. This review not only addresses the current state of the art but also aims to guide researchers, practitioners, and healthcare professionals in selecting the most suitable tools for their specific applications, ultimately advancing the pursuit of accurate and timely breast cancer diagnoses.

1. Programming Languages:

1.1 Python

- **Description:** Python is a versatile and widely adopted programming language in the machine learning community.
- **Relevance:** It offers an extensive ecosystem of libraries such as scikit-learn, TensorFlow, and Keras, making it suitable for various machine learning tasks.
- **Pros:** Strong community support, rich documentation, and ease of integration with other tools.
- **Cons:** May have a steeper learning curve for beginners.

1.2 R

- **Description:** R is a statistical programming language commonly used for data analysis.
- **Relevance:** It provides machine learning packages such as caret and randomForest, making it suitable for statistical modeling.
- **Pros:** Excellent statistical packages, strong visualization capabilities.
- **Cons:** May be less versatile than Python for general-purpose programming.

2. Machine Learning Libraries and Frameworks:

2.1 scikit-learn

- **Description:** A comprehensive machine learning library in Python.
- **Relevance:** Offers tools for data preprocessing, model selection, and evaluation, making it a go-to choice for traditional machine learning tasks.
- **Pros:** Well-documented, beginner-friendly, and extensive community support.
- **Cons:** Limited support for deep learning.

2.2 TensorFlow

- **Description:** An open-source deep learning framework developed by Google.
- **Relevance:** Suitable for building and training complex deep learning models.
- **Pros:** Scalability, flexibility, and support for deploying models in production.
- **Cons:** Learning curve for beginners in deep learning.

2.3 Keras

- **Description:** A high-level neural networks API running on top of TensorFlow.
- **Relevance:** Simplifies the process of building and experimenting with deep learning models.
- **Pros:** User-friendly interface, easy prototyping of neural networks.
- **Cons:** May offer less customization compared to direct TensorFlow usage.

3. Data Processing and Analysis Tools:

3.1 Pandas

- **Description:** A powerful data manipulation library in Python.
- **Relevance:** Useful for data preprocessing and analysis tasks.
- **Pros:** Efficient data handling, extensive functionality.
- **Cons:** May struggle with extremely large datasets.

3.2 NumPy

- **Description:** A library for numerical operations in Python.
- **Relevance:** Essential for efficient manipulation of large datasets.
- **Pros:** High-performance array operations, compatibility with other libraries.
- **Cons:** Requires some familiarity with array programming.

4. Visualization Tools:

4.1 Matplotlib

- **Description:** A popular 2D plotting library for Python.
- **Relevance:** Useful for creating static visualizations of data.
- **Pros:** Versatile and widely used in the data science community.
- **Cons:** Requires additional customization for complex interactive visualizations.

4.2 Seaborn

- **Description:** Built on top of Matplotlib, providing a high-level interface for drawing statistical graphics.
- **Relevance:** Enhances the aesthetics of Matplotlib visualizations.
- **Pros:** Easy syntax for creating informative plots.
- **Cons:** Less customizable than Matplotlib for some advanced use cases.

5. Model Deployment and Integration:

5.1 Flask

- **Description:** A web framework for Python.
- **Relevance:** Useful for deploying machine learning models as web applications.
- **Pros:** Lightweight, easy to use for small to medium-sized projects.
- **Cons:** May require additional tools for scaling.

5.2 Docker

- **Description:** Enables packaging machine learning models and their dependencies into containers.
- **Relevance:** Simplifies deployment and scalability of machine learning applications.
- **Pros:** Ensures consistency across different environments.
- **Cons:** Learning curve for users new to containerization.

5.3 FastAPI

- **Description:** A modern, fast web framework for building APIs with Python.
- **Relevance:** Facilitates building efficient and easy-to-use APIs for machine learning models.
- **Pros:** Automatic interactive documentation, asynchronous support.
- **Cons:** Relatively new, may have a smaller community compared to Flask.

6. Cloud Services:

6.1 AWS SageMaker

- **Description:** A fully managed service for building, training, and deploying machine learning models on AWS.
- **Relevance:** Simplifies the machine learning workflow and provides scalable infrastructure.
- **Pros:** Seamless integration with other AWS services.
- **Cons:** Cost considerations, may have a learning curve for beginners.

6.2 Google Cloud AI Platform

- **Description:** Allows developers to build and deploy machine learning models on Google Cloud.
- **Relevance:** Provides a range of tools for machine learning tasks in a cloud environment.
- **Pros:** Integration with other Google Cloud services.
- **Cons:** Pricing considerations and potentially complex setup.

6.3 Microsoft Azure Machine Learning

- **Description:** A cloud service for building, training, and deploying machine learning models.
- **Relevance:** Offers tools for end-to-end machine learning workflows.
- **Pros:** Integration with other Azure services, user-friendly interface.
- **Cons:** Cost considerations and potential limitations for large-scale projects.

7. Automated Machine Learning (AutoML) Tools:

7.1 Google AutoML

- **Description:** Provides pre-trained models for various tasks and allows users to train custom models with minimal effort.
- **Relevance:** Simplifies the machine learning pipeline, making it accessible for non-experts.
- **Pros:** User-friendly, requires minimal coding.
- **Cons:** Limited customization for advanced users.

7.2 H2O.ai

- **Description:** Offers AutoML functionality, allowing users to automate the machine learning pipeline.
- **Relevance:** Provides a range of machine learning algorithms with automated tuning.
- **Pros:** Wide variety of algorithms, scalability.
- **Cons:** Requires some familiarity with machine learning concepts.

7.3 DataRobot

- **Description:** An automated machine learning platform that automates the end-to-end machine learning workflow.
- **Relevance:** Streamlines model building and selection processes.
- **Pros:** User-friendly interface, comprehensive model comparison.
- **Cons:** May have limitations in terms of model interpretability.

Data Review: Breast Cancer Wisconsin (Diagnostic) Data Set

Overview:

The Breast Cancer Wisconsin (Diagnostic) Data Set is a widely-used dataset in machine learning for predicting whether a breast cancer mass is benign or malignant. The dataset comprises features computed from digitized images of fine needle aspirates (FNAs) of breast masses, specifically describing characteristics of cell nuclei present in the images.

Source and Availability:

The dataset is publicly available and can be accessed through the UCI Machine Learning Repository and the UW CS ftp server. It's a valuable resource for research and experimentation in the field of breast cancer diagnosis.

- UCI Machine Learning Repository: Breast Cancer Wisconsin (Diagnostic) Data Set

Attributes:

The dataset consists of the following attributes:

1. **ID number**
2. **Diagnosis (M = malignant, B = benign)** 3-32) **Ten real-valued features for each cell nucleus:**
    - a) Radius (mean of distances from center to points on the perimeter)
    - b) Texture (standard deviation of gray-scale values)
    - c) Perimeter
    - d) Area
    - e) Smoothness (local variation in radius lengths)
    - f) Compactness (perimeter^2 / area - 1.0)
    - g) Concavity (severity of concave portions of the contour)
    - h) Concave points (number of concave portions of the contour)
    - i) Symmetry
    - j) Fractal dimension ("coastline approximation" - 1)

Data Characteristics:

- **Dimensionality:** The dataset has 32 attributes.
- **Data Types:** Features are real-valued.
- **Missing Values:** There are no missing attribute values.
- **Class Distribution:** 357 benign samples, 212 malignant samples.

Use Case:

Researchers and practitioners in the field of machine learning and healthcare utilize this dataset to develop and evaluate models for breast cancer diagnosis. The binary classification problem, predicting malignancy or benignancy based on cell nuclei features, makes it suitable for various machine learning algorithms.

Data Quality:

The dataset is well-documented with clear attribute descriptions. The absence of missing values ensures data integrity, and the provided class distribution allows for understanding the balance between benign and malignant cases.

Recommendations:

Researchers and data scientists exploring breast cancer diagnosis algorithms can benefit from utilizing this dataset. Prior to analysis, it's recommended to perform exploratory data analysis (EDA) to gain insights into feature distributions and relationships. Additionally, model evaluation metrics, such as precision, recall, and F1-score, should be considered due to the imbalanced class distribution.

Conclusion:

The Breast Cancer Wisconsin (Diagnostic) Data Set serves as a valuable benchmark for the development and assessment of machine learning models in the context of breast cancer diagnosis. Its well-defined attributes and large sample size contribute to its relevance and reliability in research and experimentation.