UNIVERSITY OF LONDON
INTERNATIONAL PROGRAMMES

BSc Computer Science and Related Subjects

CM3070 PROJECT
FINAL PROJECT REPORT

Deep Learning on Wisconsin Breast Cancer Dataset

Author: Daarmi Sai Lanka
Student Number: 210220142
Date of Submission: 25th March 2024
Supervisor: Tarapong Sreenuch

Contents

CHAPTER 1: INTRODUCTION

Project Template: CM3015 Machine Learning and Neural Network

- Idea 1: Deep Learning on a Public Dataset

Background Information:

Breast cancer, which affects millions of people globally, is one of the most common and potentially fatal types of cancer. The timely identification of breast cancer is essential for enhancing the results of treatment and preserving lives. Machine learning algorithms have become effective tools in medical diagnosis, particularly breast cancer detection, in recent years. Machine learning can help healthcare practitioners accurately identify and categorize cases of breast cancer by utilizing the massive amounts of available data and advanced algorithms. These algorithms can provide insightful analysis and forecasts by analyzing a variety of medical data sets, including genetic information, patient histories, and mammography pictures. Machine learning algorithms show promise in transforming early detection and diagnosis of breast cancer, ultimately improving patient care and outcomes. These algorithms can learn from patterns and make data-driven judgments.

Breast cancer is essentially a range of development patterns that disproportionately affect women. People are better equipped to deal with this difficult health issue when they are aware of the distinctions between benign and malignant tumors as well as the value of early identification.

Motivation:

Despite the developments in machine learning over the years and the availability of numerous data, algorithms to detect potential signs and warn individuals have not really improved much. The battle against breast cancer is propelled not only by medical necessity but also by a potent combination of life-improving incentives. Its fundamental goal is to improve patient outcomes, including life expectancy and quality of life. This means developing therapies that are effective and have few adverse effects, understanding risk factors better to support preventative initiatives, and working relentlessly to achieve earlier detection through improved screening and diagnosing techniques. However, in this report we will focus more on what deep learning can aid in which is the identification of breast cancer types; Malignant or Benign.

Project Aim:

Two main aims in this Deep Learning investigation of the Wisconsin Breast Cancer dataset are Feature Importance Analysis and Comparison with Conventional Machine Learning Techniques.

Firstly, by identifying key characteristics, we can learn important basic details about the biology of breast cancer. We want to determine which features in the dataset are most crucial for the classification of cancer. This can be considered essential as it will aid readers/future developers to clearly identify which features are more essential than others.

Secondly, contrasting Deep Learning's performance on this dataset with that of conventional Machine Learning models offers important background information for the selected methodology. It enables us to evaluate whether Deep Learning's complexity and processing power are required for this activity. If more conventional, simpler models can produce outcomes that are on par with or even better than Deep Learning, then perhaps Deep Learning isn't the most sophisticated option. This knowledge can help develop new strategies and possibly lower the computing costs involved in setting up and using complicated models.

We will train and assess several conventional machine learning models to do a benchmark comparison. Models that were initially learned from scratch included Random Forests and Support Vector Machines. After they were trained, we assessed their performance using F1-score, accuracy, precision, and recall metrics. We may ascertain whether the extra complexity of Deep Learning was necessary for this assignment by contrasting these outcomes with the model's performance. This comparison gave the selected method important context and enabled us to determine whether simpler models may produce comparable or superior outcomes.

Deliverable:
Two essential deliverables—a thorough report and a succinct video—can be used to explain my deep learning experiment on the Wisconsin Breast Cancer Dataset.

The report will function as a comprehensive record that delineates the complete project trajectory. It can explore the history and rationale behind the selected deep-learning methodology, describe the model architecture and training procedure, and present the outcomes attained. A comparison with conventional machine learning models, a description of the selected models and their performance measures, and an analysis of whether the complexity of deep learning was warranted in this instance will all be included in the report.

A quick five-minute film is required for my project. This film offers an eye-catching synopsis of your project. It presents the report's important themes in a more engaging manner. It includes components like short explanations of the feature importance analysis and the classic machine learning model comparison, animations that illustrate the inner workings of the deep learning model, and simple data visualizations.

Word Count: 744

CHAPTER 2: LITERATURE REVIEW

1. Overview of Artificial Intelligence in Breast Cancer Medical Imaging [1]

The study offers a thorough analysis of the use of artificial intelligence (AI) in breast cancer diagnosis, with a particular emphasis on the application of AI in imaging modalities such as PET, MRI, ultrasound, mammography, and radiomics/radio genomics. It also addresses the possible advantages and drawbacks of AI in the early identification and treatment of breast cancer.

Although upon reading the title, the report may not seem relevant to the topic I am conducting my report on; Deep Learning and Machine Learning are foundational tools for developing Artificial Intelligence and this is talked about in this article.

Advantages:
At section 3.1, the paper presents the fundamental ideas behind deep learning (DL), machine learning (ML), and artificial intelligence (AI).

The study examines several deep learning (DL) advantages regarding medical imaging and breast cancer diagnosis. One of the main benefits emphasized is that DL, which draws inspiration from the biological nervous system, is more suited for human learning and does not rely on specially designed characteristics. This makes it possible for DL to use "end-to-end" uninterpretable neural network analytical decision procedures to make more factual decisions or predictions. Furthermore, the research highlights how deep learning has advanced in identifying and assimilating large amounts of data and information due to the simultaneous rapid expansion of computer technology and data storage capacity.

This feature increases its relevance to every facet of cancer research and care, including the automatic and precise identification of cancer from radiological images or stained tumor slides, which eliminates the need for radiologists and pathologists to perform redundant labor. The research also notes the successful application of DL techniques, including convolutional neural networks (CNN), in the identification of breast cancer in mammograms, the categorization of breast cancer subtypes, and the prediction of breast tumor forms in mammograms. These uses show how DL can be used to increase diagnostic precision and offer insightful information about the detection and management of breast cancer.

Disadvantages:
The paper addresses several DL drawbacks regarding medical imaging and breast cancer diagnosis. The lack of defined procedures and evaluation standards in radio genomics is one of the drawbacks mentioned. This results in differences in imaging scans and necessitates the use of sizable sample datasets and strong computing resources for validation. The paper also notes that deep learning algorithms in clinical practice may be limited due to high false positive and biopsy rates. The text also highlights the limited

repeatable consistency and the possibility that expert-defined lesions may not completely and properly reflect the genuine lesion area and extent, particularly in cases of small volumes or cryptic features.

This suggests that employing DL-based segmentation techniques to precisely define tumor boundaries and features will be difficult. Furthermore, the research notes that developing a consistent DL algorithm across people, devices, and modalities is a challenging issue, and that DL-based AI solutions for breast cancer diagnosis require many high-quality breast examination photos as a training dataset. These drawbacks highlight the necessity of additional study and advancement to overcome the difficulties DL presents in the diagnosis and imaging of breast cancer.

2. Machine Learning Algorithms For Breast Cancer Prediction And Diagnosis [2]

With a particular focus on the Breast Cancer Wisconsin Diagnostic dataset, this study investigates the use of machine learning algorithms for breast cancer diagnosis and prediction. After comparing five methods, Support Vector Machine (SVM) showed the best accuracy. The study highlights the value of SVM in the diagnosis and prognosis of breast cancer.

Advantages:
The advantages of applying machine learning algorithms to the diagnosis and prognosis of breast cancer are covered in the article. It highlights how crucial it is for cancer predictions to be as accurate as possible to alter treatment plans and increase patient survival. The study emphasizes how machine learning approaches, which have been shown to be effective, can make a substantial contribution to the early detection and prediction of breast cancer. Support Vector Machine (SVM), which achieves the maximum accuracy of 97.2%, is shown to be the most effective technique when compared to five other classifiers: Random Forest, Logistic Regression, Decision tree (C4.5), K-Nearest Neighbors (KNN), and Support Vector Machine.

The paper also highlights how data mining algorithms are used in the healthcare sector, specifically in the areas of disease diagnosis and prediction, drug cost reduction, and real-time decision-making to save lives. The essay also emphasizes how important it is to use machine learning algorithms to lower healthcare expenditures and enhance patient care quality.

The study applied various machine learning algorithms, including Support Vector Machine (SVM), Random Forest, Logistic Regression, Decision tree (C4.5), and K-Nearest Neighbours (KNN), on the Breast Cancer Wisconsin Diagnostic dataset. Among these algorithms, Support Vector Machine achieved the highest accuracy of 97.2%, outperforming other classifiers. Additionally, the Support Vector Machine showed better precision, sensitivity, and F-Measure compared to other classifiers, indicating its effectiveness in predicting breast cancer. The paper highlights the benefits of using machine learning algorithms for breast cancer diagnosis and prediction, including high

accuracy in prediction, potential for early diagnosis, and improvements in treatment outcomes and patient survivability.

3. Breast Cancer–Detection System Using PCA, Multilayer Perceptron, Transfer Learning, and Support Vector Machine [3]

The creation of a breast cancer detection system with PCA, Multilayer Perceptron, Transfer Learning, and Support Vector Machine is covered in this study. The suggested approach, which prioritizes data preprocessing, cross-validation, and classifier comparison, performed more accurately than alternative methods. The system's objective information analysis is intended to help physicians diagnose breast cancer.

Advantages:

Effective feature extraction from high-dimensional data is made possible by the combination of PCA and MLP, which may increase the accuracy of breast cancer prediction. The scalability of the suggested method is demonstrated by its great performance as the number of instances increases. When tested with the BCCD dataset, the suggested technique has a high accuracy rate of 86.97%, according to 10-fold cross-validation findings. When applied to several datasets related to breast cancer, comparative analysis indicates that the suggested strategy performs better than previous methods.

Disadvantages:

The limits and disadvantages unique to this approach are not thoroughly discussed in the text It does, however, note certain basic restrictions, such as the fact that a trained model cannot change itself when new data is input and that training times grow with data quantity. Since the research did not give a comparison with other models, it is unclear how well this strategy works in comparison to other prediction models or methodologies already in use.

4. Exploratory Data Analysis (EDA) [4]

This paper covers how to create frequency or contingency tables, summarizes data, and shows you how to run statistical tests. Non-graphical EDA techniques like tabulating categorical data and examining the properties of quantitative data are also included in the document. It also offers examples of EDA procedures based on various aims and information on testing a dataset's distribution.

Advantages:

Using EDA, researchers can find patterns, trends, and correlations between variables to obtain a basic understanding of the dataset. It aids in the formulation of theories for additional investigation. Also, EDA aids in identifying anomalous observations or extreme values that can adversely affect statistical models. It finds data gaps that must be filled in before moving on to more analysis.

Based on their relationship to the target variable or their ability to explain variation in the dataset, EDA helps choose pertinent variables for modeling. Through graphs, charts,

tables, and other visual aids, EDA facilitates the efficient dissemination of findings to stakeholders to support well-informed decision-making.

Disadvantages:
Since EDA depends more on human judgment than rigorous statistical guidelines, the interpretation of visualizations and summary data may be subjective. EDA gives broad insights into the features of the data, but without further inferential analysis, it cannot dive deeply into causal linkages or draw firm conclusions about underlying phenomena.

If done carefully, comprehensive exploratory data analysis can take a lot of time, especially when working with huge datasets with many variables. Unconscious biases may affect how visualization results are interpreted or the selection of data aspects that researchers thoroughly examine throughout the process.

Word Count: 1341

CHAPTER 3: PROJECT DESIGN

Domain and Users

This project is in the machine learning category and focuses on the use of deep learning methods in the identification of medical conditions. A branch of machine learning called "deep learning" is motivated by the composition and operations of the human brain. Artificial neural networks are employed to acquire sophisticated patterns from data, rendering it appropriate for applications such as medical diagnosis and image identification.

There are two primary groups of prospective users for this project. One is researchers and the others are medical practitioners. Researchers will be able to better understand the potential of deep learning approached for Breast Cancer diagnosis with the aid of machine learning. They will be able to gain many insights from this project, gaining knowledge about the processes involved in model creation, training, and assessment.

Medical Practitioners can recognize deep learning's potential as an additional diagnostic tool for breast cancer. They will be able to recognize the benefits and drawbacks of using Machine Learning and Deep Learning in their diagnosis. Currently, it is doubtful that this initiative will result in a clinical tool; however, without further research and more documentation for practitioners to read up on, progress will be drastically slow.

Overall Structure

The overall structure of the project was built on Jupyter Notebooks; importing multiple necessary libraries that are required to run the program and building ML models. Exploratory Data Analysis is conducted where the publicly available data is imported into the notebook. It is then explored and cleaned to guarantee data quality and stop models from picking up biases or mistakes. The project is written entirely in Python as it is versatile, allowing for multiple libraries to be used and the code being easily readable; even for amateurs. For a better understanding of the workflow, analysis and evaluation of the baseline models and the process of developing the final model will be carried out throughout the project.

Techniques and Methodologies – Exploratory Data Analysis (EDA)

EDA will be our road map to comprehending the data landscape prior to delving into intricate models. We'll use a variety of approaches with EDA to find trends, spot possible problems, and obtain insightful knowledge. This includes using boxplots and histograms to visualize data distributions, scatter plots and correlation matrices to analyze feature relationships, and looking for missing values or outliers. Through this preliminary investigation, we may customize our modeling strategy. Decision trees or Support Vector Machines, for example, might be given priority if the data shows non-linear correlations. EDA can also highlight problems with data quality that need to be fixed, like outliers that could distort the model or missing values that need to be imputed. To put it simply, EDA serves as an essential first step that ensures the data is ready for optimal learning and lays the foundation for wise model selection.

Techniques and Methodologies - Machine Learning
I will be using multiple models in supervised learning to accomplish our project's goals. We will use perceptron, the basic building component of artificial neural networks. This straightforward yet efficient model will offer a classification baseline and shed light on the linear separability of the data. To further explore and identify any non-linearities, Logistic Regression (LR) and Random Forests (RF) will thereafter be presented. Compared to perceptron, LR and RF provide better accuracy and resilience by mixing many decision trees. Lastly, we'll use a Support Vector Machine (SVM). Even non-linearly separable data points can be successfully classified using SVMs, which are particularly good at handling high-dimensional data. This layering method begins with a Perceptron to facilitate interpretation, moves on to LR and RF to capture intricacies, and ends with an SVM to handle non-linear correlations. By utilizing the advantages of each method, we will be able to determine which model works best for our particular purpose thanks to this thorough investigation.

Work Plan – Gantt Chart
Look at appendix A

Testing and Evaluation
Our deep learning model's efficacy must be evaluated using a strict testing and evaluation approach. K-Fold Cross Validation will be employed to guarantee a reliable and broadly applicable performance assessment. Using this method, the data is divided into several folds. A subset of the folds is used for model training, while the remaining folds are used for testing. Compared to a single train-test split, this procedure is repeated for each fold, yielding a more accurate assessment of the model's performance on unknown data. We will also use a range of performance measurements in line with the project's objectives. Accuracy, precision, recall, and F1-score will be utilized in classification tasks to assess the model's capacity to recognize real positives and negatives and steer clear of incorrect classifications. Through the examination of these metrics throughout the K-Fold Cross Validation procedure, we can evaluate how well the model performs in reaching the intended result, which may be a precise diagnosis of cancer or a trustworthy categorization within a certain dataset.

Word Count: 815

CHAPTER 4: IMPLEMENTATION

Defining the problem and assembling the dataset
The dataset is in the form of one singular CSV file. This is beneficial to us as it makes it simpler to assemble the dataset. Only slight checking and tidying up is required. This has

been done by checking for any null values and deleting the extra column that shows up upon importing the CSV into the data-frame; the code and technique can be seen in appendix B1.

Measure of Success

The main measures of success we will be using are the accuracy score and a confusion matrix. When evaluating a model's performance, accuracy and a confusion matrix are a useful place to start, especially for straightforward classification tasks with well-balanced datasets; which is the case for the Wisconsin Breast Cancer dataset we are using. While the confusion matrix shows potential biases towards classes by visually breaking down predictions by class, accuracy provides a summary of overall performance. This combo can be useful for making preliminary assessments and spotting serious mistakes. From the confusion matrix we can acquire the precision, recall and f1-scores of each method as well.

- Accuracy: The simplest metric, which is the ratio of the number of accurate forecasts to the total number of predictions made by the model.
- Precision: Calculates the percentage of optimistic forecasts that came true. It indicates how well the model detects true positive cases, excluding false negative cases.
- Recall: Calculates the percentage of real positive cases that the model accurately detected. It provides you with an indication of how well the model captures and does not miss any pertinent positive cases.
- F1 – Score: a single metric that combines the insights of precision and recall through a harmonic mean. It provides a fair picture of the model's performance, particularly for imbalanced datasets, by considering both false positives and false negatives.

Data Preparation

We start off by loading the dataset into a dataframe (Put in appendix for the code) and carrying out a check for any null values and removing them. Machine learning models suffer greatly from null values in a few different ways. They firstly reflect missing data that the model is unable to learn from. Predictions may result from this, particularly if the missing data isn't random. Second, nulls are handled inconsistently by different algorithms, which complicates comparisons and interpretations.

Feature Engineering and Exploratory Data Analysis

First, we print out a heatmap:

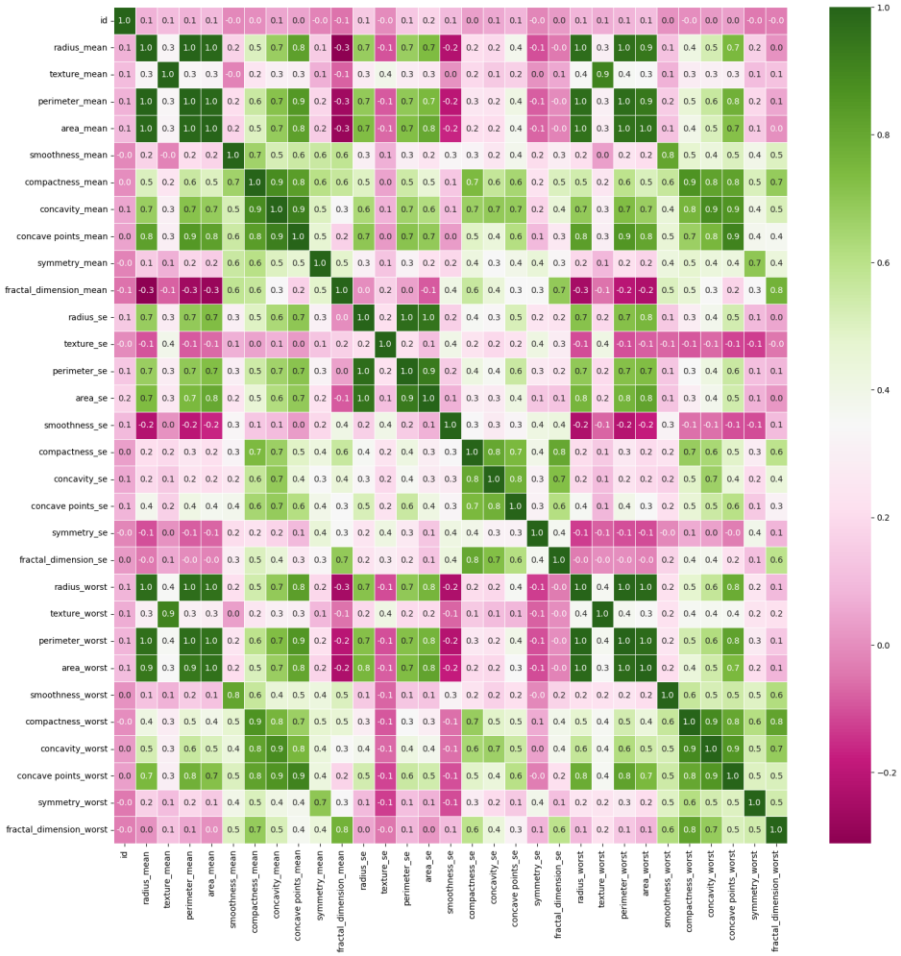


Figure 4.1: Initial Heatmap

Next, we look at the correlation between columns. From here we determine what is needed and what is not moving forward. This is decided by taking the correlation values into account and whether the column will be required elsewhere later in the project. The columns that have been removed are as follows:

- Id
- Perimeter_mean
- Area_mean
- Concave points_mean
- Concave points_se
- Radius_worst
- Perimeter_worst
- Area_worst

New Heatmap:

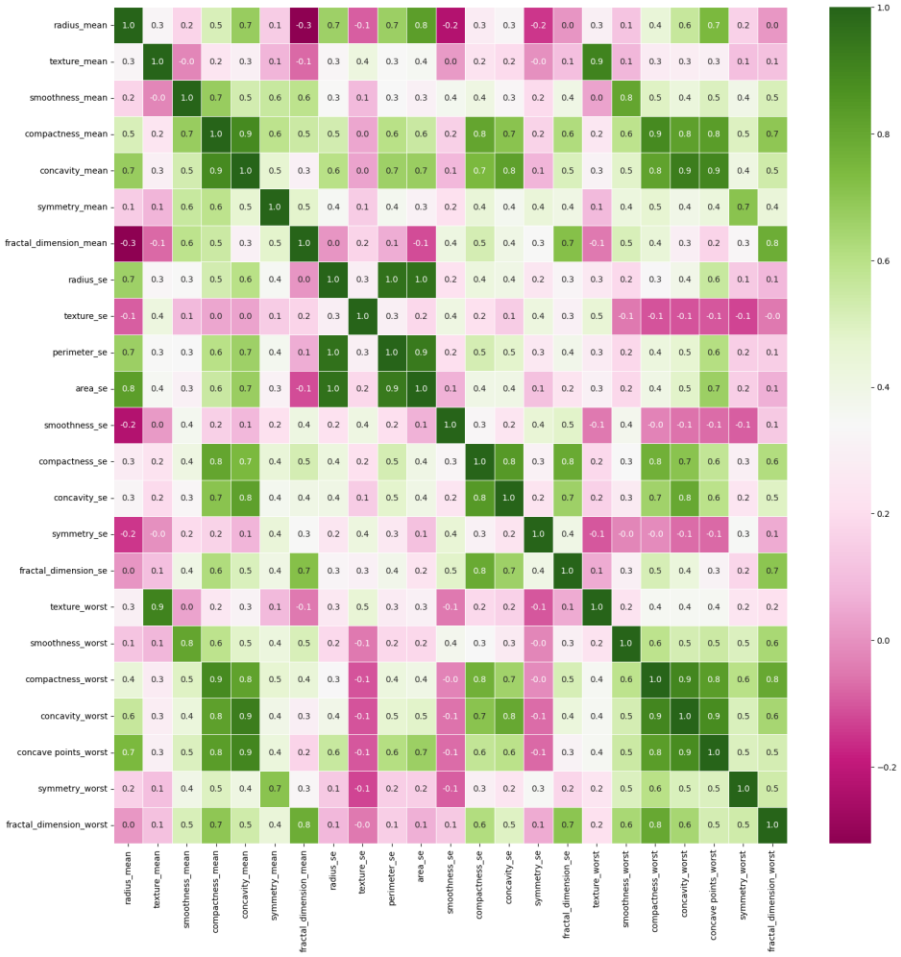


Figure 4.2: Updated Heatmap

Next, we look at the skewness values of our remaining columns (provide figure below or reference to appendix). As it can be seen below, the original skewness values for some are quite high. We want the skewness values to be between –1 to 1 as skewness values in the range of -1 to 1 indicate a distribution that is somewhat like a bell curve with symmetry (provide an appendix to an evenly distributed histogram). Since many machine learning models perform best when features have a normal distribution, this is perfect for them, especially for distance-based techniques. To achieve the desired skewness values, we carry out transformations; either square root or logarithmic transformation on columns with skewness values over 1. The results can be seen in appendix B2 and B3

Implementing Machine Learning Models and Deep Learning

We implemented 4 different models in our project. This is to showcase the difference between using different kinds of models and to investigate which one is more efficient and provides a better accuracy result. Deep learning is implemented here in the form of using the perceptron model. We use cross fold validation to split the data into the training and test sets.

Model 1: Perceptron

Perceptron is held in a higher regard when it comes to binary classification. This is because it implements neural networks to carry out its classification, assigning importance to each individual input and then weighing it together at the end to provide the outputs of its classification. Iteratively going over training data, it modifies weights according to how expected and actual outputs differ from one another. While inaccurate forecasts cause changes to move the output closer to the intended class, accurate predictions leave the weights constant. In our use of perceptron, we carried out a 7-fold cross validation to split the dataset and then ran the perceptron model on it. We got the following results

- Average Accuracy: 0.9298834258183851
- Precision: 0.9009433962264151
- Recall: 0.9095238095238095
- F1-Score: 0.9052132701421802

More details available at appendix: B4

Model 2: Logistic Regression (LR)

When it comes to binary classification problems, LR is known to excel; it calculates the likelihood that an event will fall into one of two categories, usually 0 or 1. It uses a linear model but converts the output into a probability between 0 and 1 using a sigmoid function. This makes it simple to comprehend the model's predictions because the output corresponds to the probability that anything will fall into the positive class. Additionally, because the sigmoid function limits the output to the desired range (0 for negative, 1 for positive), logistic regression is excellent at class separation. In our use of logistic regression, we carried out a 5-fold cross validation and these are the acquired results

- Average Accuracy: 0.952585002328831
- Precision: 0.9198113207547169
- Recall: 0.9512195121951219
- F1-Score: 0.9352517985611509

More details available at appendix: B5

Model 3: Random Forest

When it comes to binary classification jobs, random forest is king. Instead of taking a chance on a single model, it creates a "forest" of decision trees, each of which is trained using a different subset of the characteristics and data. Due to its diversity, the forest can record complex correlations between features and the binary outcome (0 or 1), preventing overfitting. Each tree casts a vote for a class when making predictions, with the majority taking the lead. When compared to a single decision tree, an ensemble approach produces outcomes that are more reliable and accurate. In our use of random forest, we carried out a 5-fold cross validation and these are the acquired results

- Average Accuracy: 0.9613569321533924
- Precision: 0.9339622641509434
- Recall: 0.9658536585365853
- F1-Score: 0.949640287769784

More details available at appendix: B6

Model 4: Support Vector Machines (SVM)

In binary classification, SVMs are the winners. They are skilled in locating the ideal separation line (hyperplane) in the feature space of your data that separates the two classes. This isn't just any line, though; SVMs identify the line that maximizes the margin—that is, the distance between the line and the most important data points—for each class. The decision boundary of the model can be clearly seen with this geometric method. Furthermore, SVMs perform exceptionally well with high-dimensional data and can even use kernel functions to discover non-linear correlations between features. SVMs are an effective toolkit for solving binary classification issues because of these advantages. In our use of random forest, we carried out a 7-fold cross validation and these are the acquired results

- Average Accuracy: 0.9702112100486084
- Precision: 0.9528301886792453
- Recall: 0.9665071770334929
- F1-Score: 0.9596199524940617

More details available at appendix: B7

Word Count: 1219

CHAPTER 5: EVALUATION

After the four different machine learning models have been trained and applied to our dataset, this part compares and evaluates their performance. We will examine the average accuracy, precision, recall and f1-score of each model. We can learn a great deal about the advantages and disadvantages of each strategy by analyzing these metrics, which will help us choose the best model for our classification problem.

Comparison Table:
(Metrics are in percentages, rounded up to 2 decimal places)

Algorithm	Average Accuracy	Precision	Recall	F1 - Score
Perceptron	92.99	90.10	90.95	90.52
Logistic Regression	95.26	91.98	95.12	93.53
Random Forest	96.14	93.96	96.59	94.96
Support Vector Machines	97.02	95.28	96.65	95.96

Comparisons
From the values of the table above, quite a few evaluations can be made

1) The Support Vector Machine algorithm was the best algorithm in all aspects, and this can be easily seen in the table above where it has the highest percentage values for comparison metrics. This can be due to a plethora of reasons such as:
    a. Being better to handle data of all kinds
    b. Being less prone to overfitting
    c. Providing a distinct division between classes
2) Generally, we can classify the algorithms from worst to best in the following order: Perceptron, Logistic Regression, Random Forest, Support Vector Machines. However, if we are looking specifically at recall, Random Forest and Support Vector Machines are almost identical. This means random forest is almost as good as support vector machines in identifying positive cases.

Real World Consideration

Some may argue that accuracy scores that you aim to achieve should be as close to 1.0 (100%) as possible. This is more in relation to the context of machine learning algorithms in the medical field. This is due to a concept called the zero-tolerance policy [5], which states that the accuracy achieved must be 1.0 so that no individual is misdiagnosed. (put reference to citation here)

However, this should not be the case as you do not want to risk over-fitting the model to an extent such that the results you get back cannot be trusted. Over-fitting will cause the model to work perfectly on the initial dataset but will become useless and untrustworthy to any new data that we would like to classify, and as such the trained model will not be able to be used in real word scenarios moving forward

Possible Additional Comparison Metric:

For the case of medical diagnosis, we always aim to be as accurate as possible and to be more precise, we can focus more on the number of false negatives that occur; the lower number of false negatives the machine learning model produces, the more trustworthy it is for real world applications. Looking based on false negative numbers, for our project we can determine that the Random Forest and Support Vector Machine algorithms are the best, as they have the least amount with just only 7. (Refer to appendix B4 – B7 for more details)

Word Count: 490

CHAPTER 6: CONCLUSION

The goal of this study was to investigate several machine learning models for tasks involving binary classification. We investigated the inner workings of many models, including Random Forests, Support Vector Machines (SVMs), Perceptron's, and Logistic Regression, to determine which is better suited for our dataset.

We evaluated these models using assessment criteria such as accuracy, precision, recall, and F1-score to compare their performance. Specifically for our dataset with high dimensionality, non-linear correlations between features, or overfitting problems, we found that SVMs emerged as strong contenders. They are an effective option because of their capacity to identify a distinct decision boundary and use kernel functions to achieve non-linearity. Generally, perceptron is the most ideal for a binary classification, but it was not the case here and a reason for this occurring may be since our dataset may be "too simple" for a perceptron machine learning model to be used on. Perceptron may also require more training (which can be considered for further work) and as such was not the ideal algorithm in this case.

But in the end, the situation and the properties of the data will determine which model is "best". While Random Forests are excellent at handling imbalanced datasets and intricate relationships, Logistic Regression provides interpretability. In addition to model selection, methods for preventing overfitting and the significance of high-quality data were emphasized. To guarantee data quality and stop models from learning noise in the training set, zero-tolerance methods for data labeling and early training stop were investigated.

To sum up, this study offered deep learning on a dataset specifically named "Wisconsin Breast Cancer Dataset" for binary classification. When addressing real-world classification jobs, we may make informed decisions, use the best tool for the job, and get the best outcomes by being aware of the advantages and disadvantages of various models.

Further Work:
A strong basis for comprehending and contrasting machine learning models for binary classification has been established by this effort. More research could be done to push the performance envelope. We might examine each model's hyperparameter setting in further detail, which might lead to notable advancements. Multiple model ensemble approaches are also promising, and more research into their use may produce even better outcomes.

On the data side, feature engineering methods such as generating new useful features or correcting imbalances in classes could be investigated. When working with image data, data augmentation methods that inflate the training set artificially can be rather effective. More advanced cross-validation methods can be used to reinforce evaluation even more, particularly for imbalanced datasets. Examining the model's mistakes can reveal important information about its flaws and direct future development. Lastly, putting the selected model into a working context opens the door to real-world application.

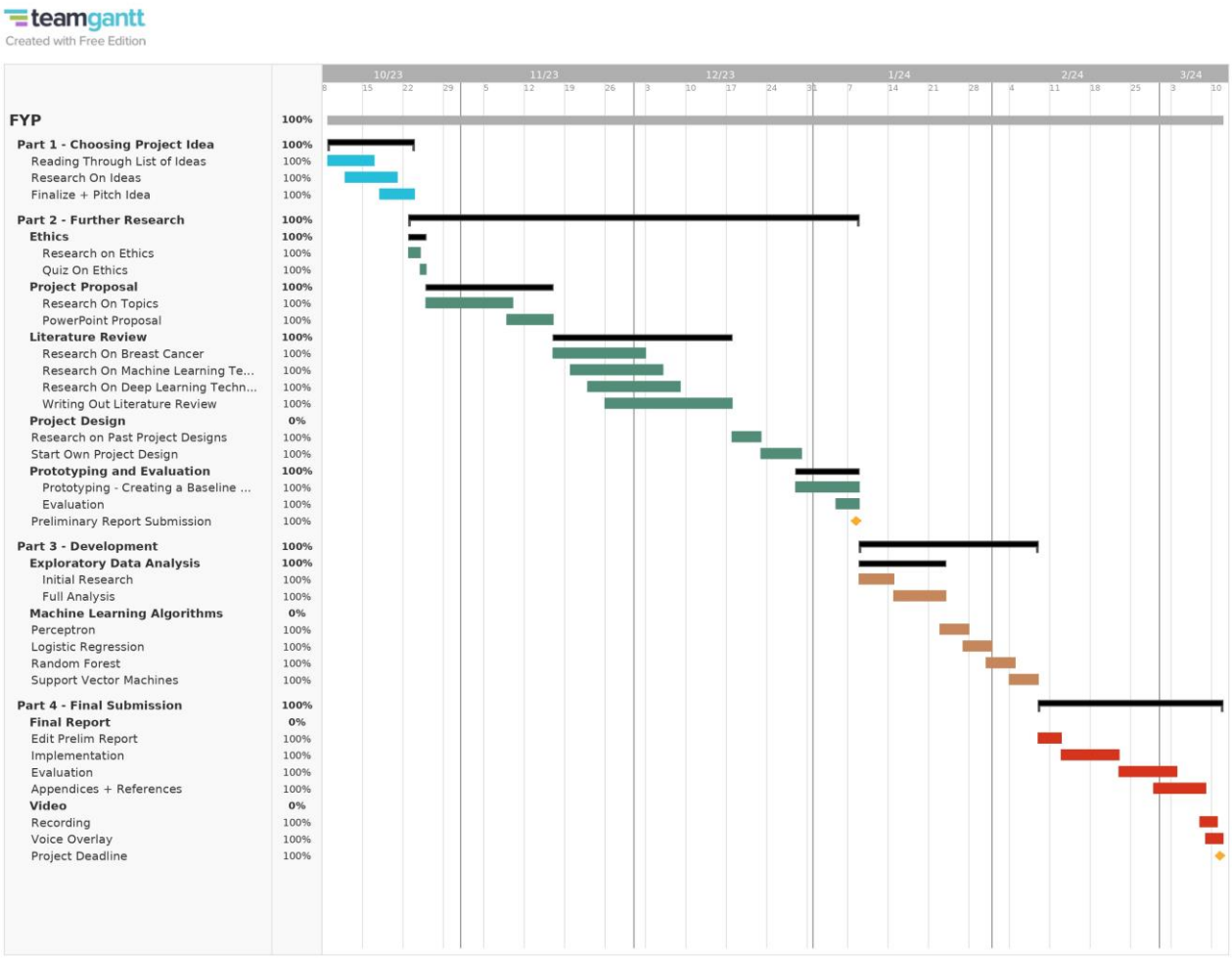
In addition to these fundamental topics, investigate explainability strategies for intricate models and produce data visualizations to gain a deeper comprehension of the data and the model's behavior. We may improve the project, obtain even better categorization results, and develop a deeper comprehension of the issue at hand by exploring these possibilities.

Word Count: 504

Overall Word Count: 5086

CHAPTER 7: APPENDICES

Appendix A – Gantt Chart



Appendix B – Codes

B1 – Cleaning Dataset

Clean Up Dataset

First we start off by checking for any Null values in any the loaded in Dataframe

```
In [3]: df.isnull()
```

Out[3]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	texture_worst	perimeter_worst	area_worst	smoothness_worst	compactness_worst	concavity_worst	concave points_worst	symmetry_worst	fractal_dimension_worst	Unnamed: 32
0	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	True
1	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	True
2	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	True
3	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	True
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	True
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
564	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	True
565	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	True
566	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	True
567	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	True
568	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	True

569 rows × 33 columns

From the above, we can see 'True' being returned for every value in 'Unamed: 32'. As it is of no use to us moving forward, we can drop it from our table.

```
In [4]: df.drop('Unnamed: 32', inplace = True, axis = 1)
        df
```

Out[4]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	radius_worst	texture_worst	perimeter_worst	area_worst	smoothness_worst	compactness_worst	concavity_worst	concave points_worst	symmetry_worst	fractal_dimension_worst
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	...	25.380	17.33	184.60	2019.0	0.16220	0.66560	0.7119	0.2654	0.4601	0.11890
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	...	24.990	23.41	158.80	1956.0	0.12380	0.18660	0.2416	0.1860	0.2750	0.08902
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	...	23.570	25.53	152.50	1709.0	0.14440	0.42450	0.4504	0.2430	0.3613	0.08758
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	...	14.910	26.50	98.87	567.7	0.20980	0.86630	0.6869	0.2575	0.6638	0.17300
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430	...	22.540	16.67	152.20	1575.0	0.13740	0.20500	0.4000	0.1625	0.2364	0.07678
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
564	926424	M	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	...	25.450	26.40	166.10	2027.0	0.14100	0.21130	0.4107	0.2216	0.2060	0.07115
565	926682	M	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	...	23.690	38.25	155.00	1731.0	0.11660	0.19220	0.3215	0.1628	0.2572	0.06637
566	926954	M	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302	...	18.980	34.12	126.70	1124.0	0.11390	0.30940	0.3403	0.1418	0.2218	0.07820
567	927241	M	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200	...	25.740	39.42	184.60	1821.0	0.16500	0.86810	0.9387	0.2650	0.4087	0.12400
568	92751	B	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000	...	9.456	30.37	59.16	268.6	0.08996	0.06444	0.0000	0.0000	0.2871	0.07039

569 rows × 32 columns

B2 – Skewness Before

```
In [7]: df.skew(axis = 0, skipna = True, numeric_only = True)
```

```
Out[7]: radius_mean               0.942380
        texture_mean              0.650450
        smoothness_mean           0.456324
        compactness_mean          1.190123
        concavity_mean            1.401180
        symmetry_mean             0.725609
        fractal_dimension_mean    1.304489
        radius_se                 3.088612
        texture_se                1.646444
        perimeter_se              3.443615
        area_se                   5.447186
        smoothness_se             2.314450
        compactness_se            1.902221
        concavity_se              5.110463
        symmetry_se               2.195133
        fractal_dimension_se      3.923969
        texture_worst             0.498321
        smoothness_worst          0.415426
        compactness_worst         1.473555
        concavity_worst           1.150237
        concave points_worst      0.492616
        symmetry_worst            1.433928
        fractal_dimension_worst   1.662579
        dtype: float64
```

B3 – Skewness After

```
In [32]: df.skew(axis = 0, skipna = True, numeric_only = True)
```

```
Out[32]: radius_mean               0.942380
         texture_mean              0.650450
         smoothness_mean           0.456324
         compactness_mean          0.564793
         concavity_mean            0.360016
         symmetry_mean             0.725609
         fractal_dimension_mean    0.853573
         radius_se                 0.572974
         texture_se                0.741415
         perimeter_se              0.637943
         area_se                   0.797609
         smoothness_se             0.404364
         compactness_se           -0.004041
         concavity_se              0.937892
         symmetry_se               0.689416
         fractal_dimension_se      0.530075
         texture_worst             0.498321
         smoothness_worst          0.415426
         compactness_worst         0.604870
         concavity_worst           0.027867
         concave points_worst      0.492616
         symmetry_worst            0.890223
         fractal_dimension_worst   0.799345
         dtype: float64
```

B4 – Perceptron

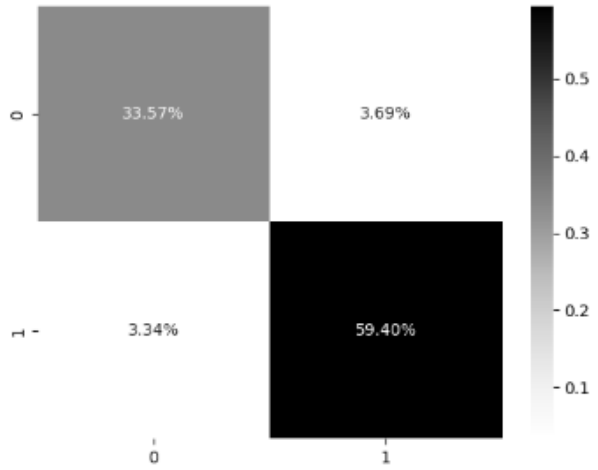
Perceptron

Accuracy + Confusion Matrix

In [63]:
```
from sklearn.linear_model import Perceptron
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_predict
from sklearn.metrics import confusion_matrix

X = df_2.iloc[:,1:]
y = df_2.iloc[:,-24]

k = 7
kf = KFold(n_splits = k, random_state = None)
model = Perceptron()

result = cross_val_score(model , X, y, cv = kf)
y_train_pred = cross_val_predict(model, X, y, cv = kf)
cf_matrix = confusion_matrix(y, y_train_pred)

print("Avg accuracy: {}".format(result.mean()))
print(cf_matrix)

labels = ['True Neg','False Pos','False Neg','True Pos']
labels = np.asarray(labels).reshape(2,2)
sns.heatmap(cf_matrix/np.sum(cf_matrix), annot = True, fmt = '.2%', cmap = 'Greys')
```

```
Avg accuracy: 0.9298834258183851
[[191  21]
 [ 19 338]]
```

Out[63]: <Axes: >



From the above, we can use the values acquired from the confusion matrix to calculate the F1 Score, Recall and Precision as well.

In [64]:
```
# True Positive
TP = 191
# True Negative
TN = 338
# False Negative
FP = 21
# False Positive
FN = 19

# Precision
precision = TP / (TP + FP)
print ("Precision = {0}".format(precision))

# Recall
recall = TP / (TP + FN)
print ("Recall = {0}".format(recall))

# F-1 Score
f1_score = 2 / ((1/recall) + (1/precision))
print ("F1-Score = {0}".format(f1_score))
```

```
Precision = 0.9009433962264151
Recall = 0.9095238095238095
F1-Score = 0.9052132701421802
```

B5 – Logistic Regression

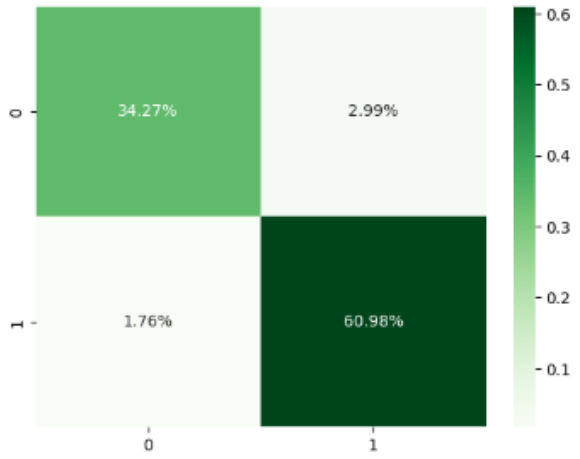
Logistic Regression

Accuracy + Confusion Matrix

In [65]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_predict
from sklearn.metrics import confusion_matrix

X = df_2.iloc[:,1:]
y = df_2.iloc[:,-24]

k = 5
kf = KFold(n_splits = k, random_state = None)
model = LogisticRegression(solver = 'liblinear')

result = cross_val_score(model , X, y, cv = kf)
y_train_pred = cross_val_predict(model, X, y, cv = kf)
cf_matrix = confusion_matrix(y, y_train_pred)

print("Avg accuracy: {}".format(result.mean()))
print(cf_matrix)

labels = ['True Neg','False Pos','False Neg','True Pos']
labels = np.asarray(labels).reshape(2,2)
sns.heatmap(cf_matrix/np.sum(cf_matrix), annot = True, fmt = '.2%', cmap = 'Greens')
```

```
Avg accuracy: 0.952585002328831
[[195  17]
 [ 10 347]]
```

Out[65]:
```
<Axes: >
```



From the above, we can use the values acquired from the confusion matrix to calculate the F1 Score, Recall and Precision as well.

In [66]:
```
# True Positive
TP = 195
# True Negative
TN = 347
# False Negative
FP = 17
# False Positive
FN = 10

# Precision
precision = TP / (TP + FP)
print ("Precision = {0}".format(precision))

# Recall
recall = TP / (TP + FN)
print ("Recall = {0}".format(recall))

# F-1 Score
f1_score = 2 / ((1/recall) + (1/precision))
print ("F1-Score = {0}".format(f1_score))
```

```
Precision = 0.9198113207547169
Recall = 0.9512195121951219
F1-Score = 0.9352517985611509
```

B6 – Random Forest

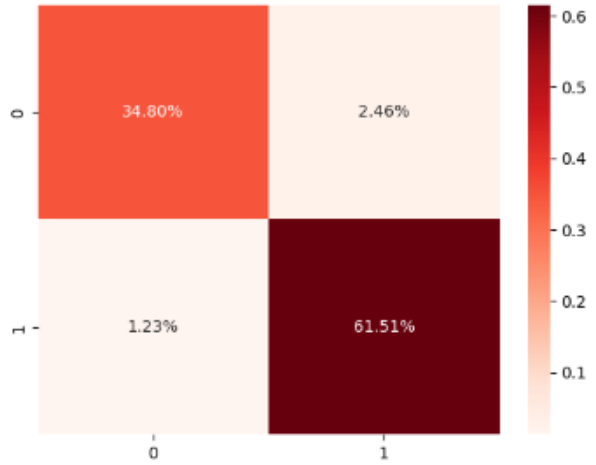
Random Forest

Accuracy + Confusion Matrix

In [67]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_predict
from sklearn.metrics import confusion_matrix

X = df_2.iloc[:,1:]
y = df_2.iloc[:,-24]

k = 5
kf = KFold(n_splits = k, random_state = None)
model = RandomForestClassifier()

result = cross_val_score(model , X, y, cv = kf)
y_train_pred = cross_val_predict(model, X, y, cv = kf)
cf_matrix = confusion_matrix(y, y_train_pred)

print("Avg accuracy: {}".format(result.mean()))
print(cf_matrix)

labels = ['True Neg','False Pos','False Neg','True Pos']
labels = np.asarray(labels).reshape(2,2)
sns.heatmap(cf_matrix/np.sum(cf_matrix), annot = True, fmt = '.2%', cmap = 'Reds')
```

```
Avg accuracy: 0.9613569321533924
[[198  14]
 [  7 350]]
```

Out[67]:
```
<Axes: >
```



From the above, we can use the values acquired from the confusion matrix to calculate the F1 Score, Recall and Precision as well.

In [71]:
```
# True Positive
TP = 198
# True Negative
TN = 350
# False Negative
FP = 14
# False Positive
FN = 7

# Precision
precision = TP / (TP + FP)
print ("Precision = {0}".format(precision))

# Recall
recall = TP / (TP + FN)
print ("Recall = {0}".format(recall))

# F-1 Score
f1_score = 2 / ((1/recall) + (1/precision))
print ("F1-Score = {0}".format(f1_score))
```

```
Precision = 0.9339622641509434
Recall = 0.9658536585365853
F1-Score = 0.949640287769784
```

B7 – Support Vector Machines

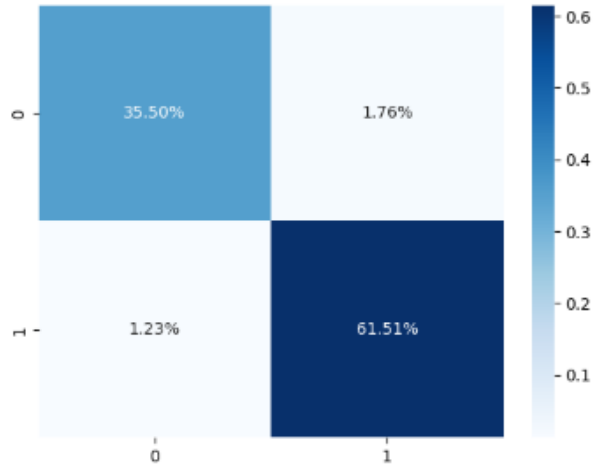
Support Vector Machines (SVM)

Accuracy + Confusion Matrix

In [69]:
```
from sklearn import svm
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_predict
from sklearn.metrics import confusion_matrix

X = df_2.iloc[:,1:]
y = df_2.iloc[:,-24]

k = 7
kf = KFold(n_splits = k, random_state = None)
model = svm.SVC(kernel = 'linear')

result = cross_val_score(model, X, y, cv = kf)

y_train_pred = cross_val_predict(model, X, y, cv = kf)
cf_matrix = confusion_matrix(y, y_train_pred)

print("Avg accuracy: {}".format(result.mean()))
print(cf_matrix)

labels = ['True Neg','False Pos','False Neg','True Pos']
labels = np.asarray(labels).reshape(2,2)
sns.heatmap(cf_matrix/np.sum(cf_matrix), annot = True, fmt = '.2%', cmap = 'Blues')
```

```
Avg accuracy: 0.9702112100486084
[[202  10]
 [  7 350]]
```

Out[69]:
```
<Axes: >
```



From the above, we can use the values acquired from the confusion matrix to calculate the F1 Score, Recall and Precision as well.

In [70]:
```
# True Positive
TP = 202
# True Negative
TN = 350
# False Negative
FP = 10
# False Positive
FN = 7

# Precision
precision = TP / (TP + FP)
print ("Precision = {0}".format(precision))

# Recall
recall = TP / (TP + FN)
print ("Recall = {0}".format(recall))

# F-1 Score
f1_score = 2 / ((1/recall) + (1/precision))
print ("F1-Score = {0}".format(f1_score))
```

```
Precision = 0.9528301886792453
Recall = 0.9665071770334929
F1-Score = 0.9596199524940617
```

CHAPTER 8: REFERENCES

[1] Zheng, D., He, X., & Jing, J. (2023, January 4). *Overview of artificial intelligence in Breast Cancer Medical Imaging*. MDPI. https://www.mdpi.com/2077-0383/12/2/419

[2] Naji , M. A., Filali , S. E., Aarika, K., Habib , B. E., Abdelouhahid, R. A., & Debauche, O. (2021, September 8). *Machine learning algorithms for breast cancer prediction and diagnosis*. Procedia Computer Science. https://www.sciencedirect.com/science/article/pii/S1877050921014629

[3] H. -J. Chiu, T. -H. S. Li and P. -H. Kuo, "Breast Cancer–Detection System Using PCA, Multilayer Perceptron, Transfer Learning, and Support Vector Machine," in IEEE Access, vol. 8, pp. 204309-204324, 2020, doi: 10.1109/ACCESS.2020.3036912. keywords: {Training;Support vector machines;Multilayer perceptrons;Breast cancer;Data models;Data mining;Principal component analysis;Multilayer perceptron network;principal component analysis;support vector machine;transfer learning}, https://ieeexplore.ieee.org/document/9253659

[4] Komorowski, M., Marshall, D. C., Salciccioli, J. D., & Crutain, Y. (1970, January 1). *Exploratory Data Analysis*. SpringerLink. https://link.springer.com/chapter/10.1007/978-3-319-43742-2_15#citeas

[5] Mitani, Tomohiro & Doi, Shunsuke & Yokota, Shinichiroh & Imai, Takeshi & Ohe, Kazuhiko. (2020). Highly accurate and explainable detection of specimen mix-up using a machine learning model. Clinical Chemistry and Laboratory Medicine (CCLM). 58. 375-383. 10.1515/cclm-2019-0534. https://www.researchgate.net/publication/339095631_Highly_accurate_and_explainable_detection_of_specimen_mix-up_using_a_machine_learning_model