June 21, 2024

C964: Computer Science Capstone

Blake Higdon

ID: 011015090

Task 2 parts A, B, C and D

# Part A: Letter of Transmittal

21 June 2024

Ed Hopkins

Regan Medical Center

742 Evergreen Terrace, Springfield, Illinois

Dear Dr. Hopkins,

In the modern American landscape, many American are unable to afford healthcare for themselves. As a director at a large medical center, it is also part of your mission to continuously improve the medical center’s practices, including practices for diagnosis. It is imperative that low-cost alternatives are introduced to provide financially sensitive patients with options. Without low-cost alternatives, lives will be lost that could have otherwise been saved if early low-cost diagnostic options were available.

Thankfully, a solution for this problem comes in the form of prediction applications for patient and professional use, in the form of an application that predicts if cancer is present in an individual based on their input information. Because this process is automated, if patients choose to not interact with healthcare professionals at all throughout the process, this application can be offered to them at extremely low costs. This also allows the patient to be able to afford the things that would really matter to them, cancer treatment.

Also, while the application will be targeting financially sensitive individuals, it works for all individuals which provides the potential for the staffed healthcare professionals to spend little to no time on the diagnostic process as there will already be a ‘jumping off point’ provided to the healthcare professional in the form of the prediction. The direct effect of this is allowing healthcare professionals to spend way more time on treatment and thus, ideally, save more lives.

This application also provides the benefit of improving patient satisfaction rates as the application allows them to forego the hassle of undergoing a formal diagnostic assessment while still providing the patients with accurate and instantaneous results. From there the patient has a reference to use when talking with a healthcare professional about their results if they even deem it appropriate to speak to someone about them.

While there are costs associated with getting this application up and running, both financially and human time costs, they are extremely minimal in comparison to the time staff will save and the profits the medical center will make over time with this low-cost application.

Since this application is storing and dealing with patient information it is important to go ahead and note that there are no ethical or legal compliance issues with the handling of data in this application. The source data was sourced from real patients who consented to having their data being used. Also, the patients’ name or any sort of identification are not present in the data set, providing complete anonymity for the patients whose data is present in the initial data set. Going forward, the project team tasked with this application will ensure continuous compliance by keeping the data in a protected environment and ensuring all data that is to be added to the data set follows all ethical standards and laws.

As someone with a formal education in both business and computer science, as well as professional experience in those fields, I understand the application from both a technical and business standpoint. It is because of this background that I know without a shadow of a doubt that the successful execution of this application will provide the medical center with a continuous source of revenue, provide individuals with a low-cost alternative to more a formal cancer diagnosis, save the healthcare professionals’ time which allows them to focus on more important tasks, and ultimately, save the lives of patients who otherwise wouldn’t have been able to afford a diagnosis in addition to patients more interested in a real time and accurate prediction.

Sincerely,

Blake Higdon

Software Engineer

# Part B: Project Proposal Plan

## Project Summary

Unfortunately, in modern day America, many individuals are priced out of healthcare. Given the cancer is one of the terminal illnesses on most people’s minds, especially as they age, it is important to provide healthcare solutions to all individuals, not just the middle and upper class of Americans. A model such as this one that can understand a patient’s risk factors to predict if cancer is present can be aidful to healthcare providers and individuals. For hospitals and other such healthcare providers such as specialists, this model can aid them in extremely early diagnosis which would then lead to early intervention. For the patients/individuals, ultimately their lives could be saved as a direct result of this model providing them with a low-cost solution to check for cancer.

As a healthcare provider, Regan Medical Center has a responsibility to its patients to be continuously improving its practice. One way to accomplish this, while also providing more healthcare accessibility to more financially sensitive individuals is to integrate this cancer prediction model into the currently existing diagnostics system. The ultimate goal of this application is to save human lives through assisting medial professional in decision making, improving early cancer detection, and allowing for early treatment.

As far as deliverables are concerned, there are four main project deliverables.

1. The Model Itself
   1. This binary model outputs “cancer” or “no cancer” to the patient. It will be trained using the patient input data of age, gender, BMI, smoking status, genetic risk, physical activity, alcohol intake, and cancer history.
2. Interactive Element
   1. Through the consol after running the program, the patients and healthcare professionals can input the required user data to receive instant predictions about if the patient has cancer or not. At a later time, if the hospital wishes, this element can be web hosted.
3. User Guide
   1. This is the clear and detailed documentation which tells the end users how to use the application from start to finish.
4. Visualizations
   1. Visualizations will be provided in the form of confusion matrix heatmaps, ROC curves, and classification report charts which are able to show the model’s metrics and allow for some interpretation.

This application will benefit the hospital by significantly enhancing the diagnostic capabilities of the hospital. More detailed benefits of this are improved early detection, improved efficiency, and improved patient satisfaction.

By providing a low-cost self-service option to patients that can use the application whenever they want, real time and accurate cancer predictions are provided which will then increase patient trust in their healthcare professionals and the hospital as a whole.

This ‘at will’ prediction model gives healthcare professionals the ability to use the patient’s prediction to drastically improve the time it takes to get officially diagnosed, resulting in early treatment and ideally saving lives.

Also, because we have these timely predictions as a result of the automated model, the time and effort usually required by healthcare professionals to do this is eliminated, allowing them to spend more time on other tasks such as treatment.

## Data Summary

The raw data set was taken directly from kaggle.com, and the data was unedited from its original form on Kaggle.

In the design phase, relevant data sources will be identified. For this application, the Kaggle data set was identified, reviewed, and then included as the data set in the csv file. In development, if any data were missing, it would have to be handled; however, one of the reasons this data set was selected was due to it having complete entries. In addition to this, we must also standardize and normalize the data to have consistency. Also in the development phase, the data must be split into training, test, and validation sets so that the model can be properly evaluated in terms of its performance. In the maintenance phase of the application, the data must be constantly monitored to ensure the quality of the data is meeting standards. We can also include validation checks within the application to ensure that any data anomalies are reported as soon as they are found. For the “other” phases, data security can be handled by ensuring that it is stored securely with the proper access controls and encryption enabled on it to protect patient information.

The data set adequately meets the needs of this project as all predictors are well known risk factors for all types of cancer. As far as anomalies as concerned, we have ensured that none are included in the initial data set; however, if the hospital wished to add patient data to this data set to fine tune the model, this is how the most prominent anomalies would be handled:

* Outliers
  + Identified outliers will be dealt with using z-score and IQR to prevent them from drastically skewing the model.
* Missing Data/Incomplete Data
  + Users are not allowed to submit data that is not accepted by the model; however, if the hospital were to manually edit the data set with incomplete data, that data would be excluded from the model to ensure the model maintains data integrity.

With this data set, there are no ethical or legal concerns present. All patient data is anonymous, which protects the privacy of the patients and eliminates privacy concerns. Also, patients consent to having their data be included in the data set so there is no ethical issue of using patient data behind their back. This data is HIPAA compliant and for continual legal compliance, the data must stay HIPAA compliant. The model will also be routinely checked for any potential biases present in the data as well as disparate impact to specific demographic groups.

## Implementation

While there are several industry-standard methodologies, CRISP-DM would be the best methodology for this application. CRISP-DM (Cross-Industry Standard Process for Data Mining) is an industry standard used for machine learning applications such as this. CRISP-DM lays out planning and execution methods which consist of the following:

1. Business Understanding
   * This consists of understanding the project objectives from a business perspective as well as defining the business objectives.
2. Data Understanding
   * Data Understanding is the process of collecting our initial data set and familiarizing ourselves with it. If any issues exist within the initial data set, they must be identified.
3. Data Preparation
   * This is the process of cleaning and formatting our data set as well as handling identified outliers and/or missing values. Variables are also transformed at this step if needed.
4. Modeling
   * In this step modeling techniques are selected and applied to the data set.
5. Evaluation
   * In this step the model must be evaluated to ensure that it meets the business objectives that were defined in step 1. The model must also be confirmed to be reliable in this step.
6. Deployment
   * Finally, in the deployment step the models will be deployed into the live production environment. After a successful deployment, the model must be monitored and maintained (Olavsrud, n.d.).

The implementation plan for the model would begin with defining the overall objective(s) of the project alongside identifying stakeholders. Early implementation conversations should focus on establishing all requirements for the project in tandem with KPIs. A thorough understanding of the goals of the project allows for better alignment with the overall business need(s).

After the first implementation step has successfully been completed, the source data must then be collected and prepared. Relevant sources like hospital databases and/or online databases such as Kaggle should be looked over for any potential data that should be included in the initial data set. After all source data has been identified it should then be carefully looked over and cleaned of any missing values or outliers by either removing them from the data set or implementing solutions to handle those issues. Once the set is properly cleaned, the data set needs to be split into test sets, training sets, and validation sets. Splitting the data this way allows for the proper facilitation of machine learning models. Then for binary sets such as this, the logistic regression algorithm can then be trained and fine-tuned using the training set which was just created. After successful training overall model performance can be shown utilizing the validation set. Metrics such as model accuracy, precision, f1-score, recall, and ROC Area Under Curve are all useful metrics that should be generated.

Then, the model should be fine-tuned, and the best performing model version should be selected to move forward with. Then the model that is selected must be evaluated to make sure that it is meeting the business goals/objectives. From there, the model will be deployed into the live production environment and any integrations that are needed because of this will all be handled.

After a successful deployment, the model must be monitored with routine maintenance to ensure that the model is consistent with the KPIs and business objectives of it. Optionally, new data can be introduced during model maintenance to enhance model accuracy and relevancy.

Additionally, once the model has been used by users for a set amount of time, user feedback can then be gathered and integrated with continuous improvement strategies for the model. Doing so will again keep the model the best it can possibly be in supporting healthcare providers and making as accurate predictions as possible.

## Timeline

|  |  |  |  |
| --- | --- | --- | --- |
| Milestone or deliverable | Duration  (hours or days) | Projected start date | Anticipated end date |
| Objective Setting & Identifying Stakeholders | 3 business days | July 1st | July 3rd |
| Define Project Requirements and KPIs | 2 business days | July 5th | July 8th |
| Collect Initial Data | 5 business days | July 9th | July 15th |
| Clean Data | 4 business days | July 16th | July 19th |
| Split Data | 3 business days | July 22nd | July 24th |
| Train & Fine-Tune Logistic Regression Model | 12 business days | July 25th | August 9th |
| Evaluate Logistic Regression Model | 5 business days | August 12th | August 16th |
| Ensure Logistic Regression Model Meets Business Requirements | 2 business days | August 19th | August 20th |
| Deploy Into Production Environment | 5 business days | August 21st | August 27th |
| Monitoring & Maintenance | Ongoing | August 28th | Continuous |
| Collect User Feedback | Ongoing | August 28th | Continuous |
| Continuous Improvement Strategies (Includes adding in new data) | Ongoing | August 28th | Continuous |

## Evaluation Plan

Throughout the development lifecycle verification methods will be employed after each major step to ensure beyond a shadow of a doubt that the step has been completed before moving onto the next step. As we begin with the business understanding step, as shown in the above CRISP-DM model, we will verify successful completion of the step by conducting a review with all stakeholders. During the review all project objectives, the scope of the project, and any assigned roles and responsibilities should be verified to ensure the successful completion of this step.

Moving onto the Data Understanding step, all relevant stakeholders must meet again. When these stakeholders meet, to verify the successful completion of this step, the relevant stakeholders much conduct exploratory data analysis. This is to ensure that any potential issues the data might have, such as outliers, are identified. Once this has been completed and the stakeholders have verified the data to be of quality, this step is completed.

To verify the completion of the data preparation step, the development team, or the team tasked with handling the data should clean the data by developing scripts of manually cleaning the data set. The data should also be successfully split in this step. To verify the successful completion of this step, the team will conduct peer reviews to verify any scripts that were created, ensuring successful cleaning of the data, and the splitting of the data.

For the modeling step, the team will ensure successful completion of this step by using cross-validation techniques to verify the model’s stability and reliability of metrics on the validation set. Also, while this doesn’t apply to this project, in projects where models other than logistic regression are being considered, all models being considered should be trained and validated. Once the development team, or team tasked with this step has completed the above requirements then the modeling step is complete.

To verify the completion of the evaluation step, the team tasked with this step must ensure model performance metrics by comparing them to the model goals/objectives which were defined in the first step. Also, the team must test for disparate impact by conducting bias testing. Once the team is ready to move on from this step, the team must show their evaluation to stakeholders and gain their approval before moving onto the deployment step.

For the last step in the CRISP-DM steps, deployment verification can be done through integration testing by the team. The team must verify that the deployed model has no integration issues and is able to produce expected results based on the user input.

While the monitoring, maintenance, user feedback & continual improvement post-deployment steps are not a part of CRISP-DM, these continuous post-deployment steps can be continuously verified with monitoring tools, a company wide commitment to continuous improvement, and fully functioning automated tools which are able to capture feedback.

Upon completion of the project and a successful start to the continuous steps, the project can be verified using a variety of steps, however multiple steps will be used in combination with each other to verify the project completion. To verify the project user acceptance testing, external validation, and an ethical and legal review will be conducted to verify successful project completion.

For user acceptance testing, the project team will work with healthcare providers and individuals to validate that the deployed model is meeting expectations. Test cases will be proved to users for them to use and get predictions. Once users have exhausted the test cases provided to them, their feedback will be gathered on the topics of usability, accuracy, and prediction relevance. Based on the user feedback changes can be made. To further verify using external validation, the team will find an external data set with the same variables as the model’s data set. Then the model’s performance on the dataset will be tested and compared with the performance of the original test set. Doing this properly ensures that our model generalizes well to new, unseen data, which is important as most if not all user data will be unseen. Lastly, internal and external legal and ethical experts will review the way the model handles and process data to ensure it is compliant with the law and any relevant ethical guidelines.

Only once all three verification methods have been successfully completed will the model’s completion be considered a success.

## Resources and Costs

To successfully design, create, and implement this application, initial investments must be made to successfully launch the application. The hardware and software costs are the following:

|  |  |
| --- | --- |
| ITEM | COST IN US DOLLARS |
| Workstations for Development | $4,000 |
| Licenses for Software (e.g. PyCharm license) | $500/year |
| Software Libraries | Open Source |
| Visualization Tools | Open Source |
| Antivirus and Security Software (e.g, MacAfee) | $200/year |
| Cloud Computing Resources (e.g. AWS EC2) | $2,000/year |
| Miscellaneous Allowance | $500 |
| TOTAL HARDWARE & SOFTWARE COSTS | $7,200 |

Once all software and hardware items are purchased and set up properly, it is now time for the human resources to begin work. Once the project teams begin work the expected costs and times are the following:

|  |  |  |  |
| --- | --- | --- | --- |
| POSITION | RATE | ESTIMATED HOURS | TOTAL |
| Project Manager | $75/hour | 125 | $9,375 |
| ML Engineer | $80/hour | 300 | $24,000 |
| Software Engineer | $70/hour | 250 | $17,500 |
| Data Engineer | $80/hour | 250 | $20,000 |
| QA Tester | $50/hour | 150 | $7,500 |
| Data Scientist/Analyst | $60 | 200 | $12,000 |
| TOTAL |  | 1,275 | $90,375 |

In addition to these costs, it is also important to know that there will be some costs associated with the environment in which the application is hosted as well as deployment and maintenance costs. The estimated costs associated with these are the following:

|  |  |
| --- | --- |
| ITEM | COST IN US DOLLARS |
| Deployment Services | $3,000/year |
| Hosting Services | $1,000/year |
| Maintenance | $3,000/year |
| Monitoring Services | $1,000/year |
| Data Backup Costs | $1,000/year |
| TOTAL | $9,000 |

In summary, total estimated costs associated with the project are $106,575. Of that amount, it is estimated that $9,000 of it will be annually recurring.

# Part C: Application

Application submitted in a zip folder along with this word doc.

# Part D: Post-implementation Report

Create a post-implementation as outlined below. Provide sufficient detail so that a reader knowledgeable in computer science but unfamiliar with your project can understand what you have accomplished. Using examples and visualizations (including screenshots) beyond the three required is recommended (but not required). **Write everything in the past tense.**

## Solution Summary

As previously mentioned, the rising costs of healthcare price many Americans out of service they should have access too, which creates a unique opportunity for your business. Americans priced out of the outrageous cost of healthcare still have small amounts of disposable income but because there is not a low-cost alternative available to them, you lose business. To have services available to all at your hospital I have created an automated solution for these individuals which you can provide at very low cost to them. This solution uses machine learning to predict if cancer is present in a patient, which is one of the illnesses most people are concerned with as they age. This solution provides the widest net due to the illness it tests for, and it also has solved the people’s issue of not having a low-cost healthcare solution and your problem of not having a low-cost alternative for those individuals who cannot afford healthcare.

The application provides a solution to the question “do I have cancer” by using machine learning. More specifically, this application has been trained with a large dataset of real patients along side their medical information and diagnosis. With this data set I then used the python library sklearn to create a linear regression model with the data set and using this model the application predicts if a patient has cancer by running the user submitted information against the model to make a prediction by using the library function ‘.predict()’. By making this prediction with a high degree of accuracy, we provide the people and the hospital with a solution that solves the issues they were having.

## Data Summary

The raw dataset comes from the website Kaggle. The direct link can be found on the reference page. This data set was chosen over others as it has been preprocessed and cleaned to ensure that users of the data set can focus on the most critical aspects of cancer prediction. Eliminating a lot of that ‘background noise’ is crucial in ensuring our model is as accurate as possible.

Throughout the development life cycle the data has always been contained in a .csv file that lives within the PyCharm project folder. When the program is run it starts by creating a reader to read through the csv file and save it to rawCancerData. Once the raw data is loaded in, it is then self-containing in the program, meaning we do not keep referencing the source data in the csv file over and over again after out initial load of it. Once the data is loaded into the running instance of the program it is split up into x and y values representing the medical information and the diagnosis before it is trained and used to create a linear regression model that we then can make our predictions with given user submitted information.

## Machine Learning

For what our ML method is, that would be logistic regression. This method is primarily used for binary classification, such as our prediction model where the answer is either Yes or No, making it a perfect fit for this data set.

This method was developed by following a few steps, data collection, preprocessing, splitting, training, evaluation, then prediction. For our collection of data, we read the data set along with all its features. After that, we preprocessed the data by further cleaning it, making sure that there are no missing values, and standardizing features to ensure that our model will be as accurate as possible while also performing very well. We then split the data to evaluate its performance on unseen data, such as user input data. After that, we finally were able to train the model by applying the logistic regression which was discussed earlier to the training data set. This logistic regression model can find the best fit by estimating the coefficients that minimize the distance between predicted and actual labels. Next, evaluating the model required us to evaluate the trained model on the test set using a variety of metrics which will be shown in the visualizations section. Finally, after the above steps were developed, the model can then be used to make predictions.

Designing it this way ensures that our application is simple and straightforward, as logistic regression is a relatively simple algorithm and it can be easily implemented and interpreted. Also, in addition to being simple, logistic regression has great performance on binary problems, such as this one where the answer is either Yes or No. In addition to this, logistic regression models can scale extremely well, meaning that if the hospital has their own data that they wanted to include into the csv file, they could do so without harming the performance of the model.

## Validation

For our logistic regression method, it can be validated using cross-validation. Cross-validation is the practice of splitting our dataset up into multiple smaller subsets. Then we simply train the model on some subsets and evaluate it on the remaining subsets. We would cross-validate the method multiple time and then average out the results to find a more accurate estimate of the application’s performance.

To develop this validation method, we first partition our dataset, which would be the data found in our csv file, into *K* folds. For our *K* value, it is common practice to pick a number from 5 to 10, in this case we select 10. Each fold is the validation set once while the rest of the folds make up the training set, repeating the process *K* times. After each iteration performance metrics are calculated and saved to be averaged out at the end. Finally, after the validation method is finished iterating, the individual performance metrics from each fold are averaged out to show the final evaluation of the model’s performance.

By selecting cross-validation we ensure that we end up with a more reliable estimate of the model’s performance compared to if only one train-test split was done. This is partially because doing this ensures that every single data point within our data set is used both to test and to train the model, which also maximizes the use of the data set.

To implement cross-validation in the future we would need to do the following:

1. Implement the cross-validation
   1. Integrate the code into the project
2. Run the model
   1. Execute the cross-validation code on the data set
3. Analyze Results
   1. Collect the results and analyze the cross-validation scores to assess the overall performance of the model.
4. Hyperparameter tuning (optional)
   1. Based on the results of the cross-testing we might want to tune the hyperparameters of the regression model.

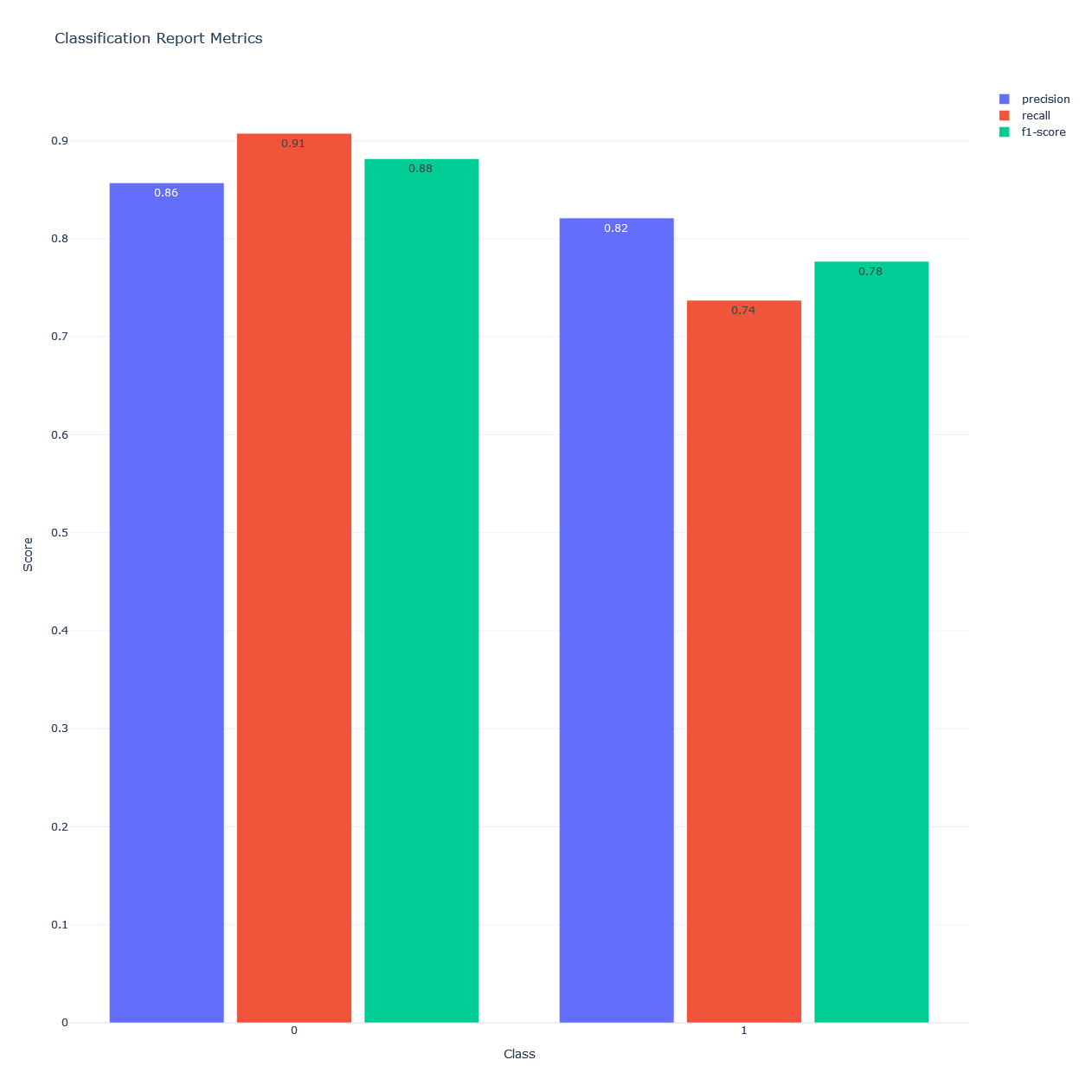
## Visualizations

## 

Above is the first visualization which is a confusion matrix heatmap. For binary problems such as this, confusion matrixes are a 2x2 table consisting of four major regions which are:

1. True Positives: The number of instances where the model can correctly predict that a patient has cancer. This is shown in the bottom right box.
2. True Negatives: The number of instances where the model correctly predicts the negative class which in this case is no cancer. This is shown in the top left box.
3. False Positives: The number of instances where the application incorrectly showed that a patient with no cancer had cancer. This is shown in the top left box.
4. False Negatives: The number of instances where the model incorrectly predicted that a patient with cancer had no cancer. This is shown in the bottom left box.

From a confusion matrix alone, we can find the accuracy, which is the overall correctness of the model, precision, which shows how many of the instances predicted as cancer are actually cancer, recall, which measures the ability of the model to correctly identify actual positive cases, and f1-score, which is the mean of precision and recall.



While the confusion matrix might not directly show it, this bar chart showing the classification report metrics for the application does show the statistics defined above for each of our classes, with 0 being the no cancer group and 1 being the cancer group.

To further describe the statistics using formulas:

**Precision** = True Positives / (True Positives + False Positives)

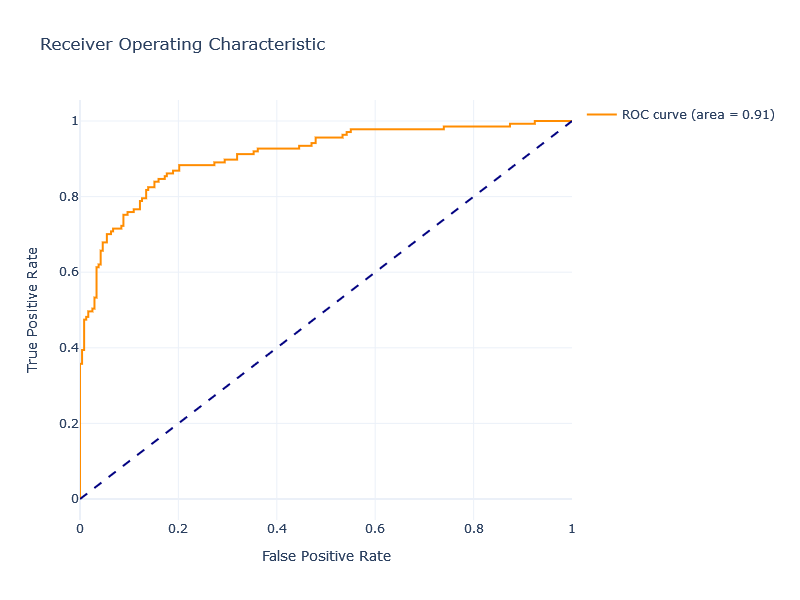
**Recall =** True Positives / (True Positives + False Negatives)

**F1-Score =** (precision x recall) / (precision + recall)

To further interpret the bar chart, for our 0 class (no cancer), we see an f1-score 0.88 or 88%, which is a high f1-score. Because of the high f1-score we are then able to confirm that the application performs well for patients who do not have cancer. Also, because the f1-score is high that means our model is both precise and has a good recall score meaning that we have low amounts of false positives *and* low amounts of false negatives.

For the 1 class (cancer), the model has a f1-score of 0.78. This indicates that while the model doesn’t perform as well when compared to the 0 class, it still performs quite well. The model is still able to correctly identify many actual cancer cases.

Given the f1-scores, patients who use this application can be confident in the model’s prediction and seek further guidance from the hospital if they are predicted to have cancer or just have any questions or concerns. From the perspective of the hospital, doctors and other medical professionals working with patients who bring a prediction back to them do not have to doubt the model and can operate as if the prediction is fact until scans are able to be done to 100% confirm or deny the presence of cancer in the patient.



For the third visual, we have the visual of the application’s receiver operating characteristic (ROC). The ROC is a special graph that we can use to further evaluate performance of binary classification models such as this one. The point of the graph is the plot the true positive rate against the false positive rate. Keep in mind that the true positive rate is also know as recall, which was discussed above, and the false positive rate is equal to [False Positives / (True Positives + True Negatives)], it measures the proportion of actual negatives that were labeled as positives. The ROC curve you see in orange depicts the trade off between the true positive rate and the false positive rate at various thresholds, the closer the curve is to the left boarder the better the model is.

The area under the curve ranges from 0 to 1 and in this graph, we see that it is 0.91. A perfect area under curve would be 1.0 which means that the model is perfect and makes no errors whereas anything under 0.50 would mean that the model is basically just randomly guessing. Because our model has an area under curve of 0.91, we know that our model has excellent performance. This means that there is a 91% chance of our model being able to correctly distinguish between a randomly chosen positive and a randomly chosen negative instance.

## User Guide

Include an enumerated (steps 1, 2, 3, etc.) guide to execute and use your application.

* Include instructions for downloading and installing any necessary software or libraries.
* Provide an example of how the client should use the application.

**Background knowledge/Before you start:**

* The user should have general working knowledge of a computer.
* The user must be on a Windows system, version 10 or 11
* The user should have working internet in order to download the tools and resources that are required if they do not already have them installed.
* The user should have access to the folder containing the application.

**Steps To Operate the Application:**

1. The user must ensure that the latest version of python is installed on their machine. If the user doesn’t know or is unsure, please do the following:
   1. Go to <https://www.python.org/downloads/> (JetBrains, n.d.)
   2. Navigate to the middle of the screen where it says “Python releases by version number”
   3. Click the “download” button next to the latest release of python
   4. Follow the directions of the downloader to download and install python on your system.
2. Once the user is confident that Python is on their machine, we will now work on setting up the IDE. If the user has PyCharm 2023.3.4 or a more recent version on their machine, skip to step 3, otherwise follow the instructions below.
   1. Navigate to <https://www.jetbrains.com/pycharm/download/?section=windows> (JetBrains, n.d.)
   2. Click the download button and while the software is downloading set up your JetBrains account by navigating to <https://account.jetbrains.com/login> and following the instructions under the “Not registered yet” text (JetBrains, n.d.).
   3. Once your account is registered successfully, navigate to <https://www.jetbrains.com/store/?section=personal&billing=monthly> and purchase a 1 month subscription to PyCharm (or any other subscription plan that includes access to PyCharm) (JetBrains, n.d.).
   4. Once that is complete, go back to your download of PyCharm, follow the instructions for installing it and login when prompted to do so.
3. Once the user has python and pycharm all set up on their machine, ensure that you have a python interpreter set up. If you are unsure or know you do not have one set up navigate to <https://www.jetbrains.com/help/pycharm/configuring-python-interpreter.html#add-existing-interpreter> and follow the instructions for setting up the interpreter (JetBrains, n.d.).
4. After the above steps have been completed, and the user makes sure they are logged into PyCharm, we will now click the “file” button and navigate to the “open” button within that dropdown.
   1. **A screenshot of a computer

      Description automatically generated**
5. Once the user clicked on the “open” button and the file explorer has been opened, navigate to where the application folder has been saved. If the user did not change the name, the name of the folder would be: CAPSTONE
6. Once the user opens the application into PyCharm, navigate to main.py which can be found in the CAPSTONE folder.
   1. A screenshot of a computer

      Description automatically generated
7. In main.py, make sure that all the imports and libraries are on your machine, if you are unsure or know they are not on your machine, hover your mouse over the name of the library/import and follow the prompts from PyCharm on how to get it on your machine.
8. After making it to main.py, we need to click the green run arrow icon in the top right corner.
   1. ****
9. After the run button has been clicked, you will notice that the console on the bottom of the IDE has opened and started to ask the user questions. Answer the questions one by one until you finish all the questions which will then give you your prediction.
10. (OPTIONAL) Depending on the result, the user may want to contact a healthcare professional for any comments or concerns they may have.

# Reference Page

JetBrains. (n.d.). PyCharm: The Python IDE for professional developers. Retrieved June 22, 2024, from <https://www.jetbrains.com/pycharm/>

JetBrains. (n.d.). Essential tools for software developers and teams. Retrieved June 22, 2024, from <https://www.jetbrains.com/>

Olavsrud, T. (n.d.). CRISP-DM: A Standard Methodology to Ensure a Good Outcome. Retrieved June 21, 2024, from <https://www.datascience-pm.com/crisp-dm-2/>