Contents

1. Introduction:

For various business ranging from a family owned local bakery to large corporations like Walmart, Apple, and Target; Data storage is widely used and in high demand in today's interconnected business ecosystem. Take Walmart as an example. Each day potentially several petabytes of data need to be stored due to the millions of transaction records, updates of inventory stock, information of new customers and etc. Merchants are always required to choose the best storage solution to avoid unnecessary overhead costs and provide more comfort services for customers. Their solution is storage management systems such as the cloud.

Currently, cloud storage seems to be the best option for lots of companies. It is affordable compared to traditional disk storage since it doesn't need to be operated by the business itself and pays for the ownership, maintenance, and operation of servers. Cloud storage is a cloud computing system which is mainly focused on data storage and management. Merchants can simply rent these servers provided by companies like Amazon, Microsoft and use a lot of traditional facilities for free. Most importantly, they can quickly access data stored on the cloud at any time, anywhere, with many different types of devices.

As the amount of information has grown exponentially in recent years, cloud storage technologies have been rapidly developing and are widely used in various fields. Don Tait, the senior blockchain analyst at UK blockchain investment firm FUNATOZ, predicts that the cloud storage market could exceed $131.7 billion by 2020, with a compound annual growth rate of 28%.

Our project is sponsored by Daniel Boros who is a staff software engineer in the IBM Spectrum Protect project. IBM Spectrum Protect is a generalized monolithic server with cloud capabilities, which is designed to simplify protection for large amounts of data hosted in physical, virtual, software-defined, and cloud environments for all customers. Also, Spectrum Protect simplifies backup administration, improves efficiencies in the backup process, and enable scalability to an entire enterprise of inputs. Our project is committed to improving secure enterprise level storage for IBM by providing a good solution for classifying data into the appropriate storage tier.

In this document, we will specify the problem our sponsor is currently encountering in data classification as one of the storage options. Then we will discuss the solution to address this problem with functional, non-functional, and

environmental requirements. Also, it is followed by potential risks and challenges we may face throughout this problem. Finally, we will show our future plan and conclude this document summarizing our requirements specification.

2. Problem Statement:

Our sponsor's business mainly focused on data classification as various storage options. When data is uploaded to one of Spectrum Protect's storage options, it needs to be categorized into a storage tier. These storage tiers are varying degrees of hot or cold. Hotter storage allows for quick access to data that is frequently used. Colder storage takes a longer time to get the data that is rarely accessed. Also, the high-performance hot storage comes at a higher price than the cold storage. To avoid unnecessary overhead costs for IBM, the data should be correctly assigned to the hot storage and cold storage.

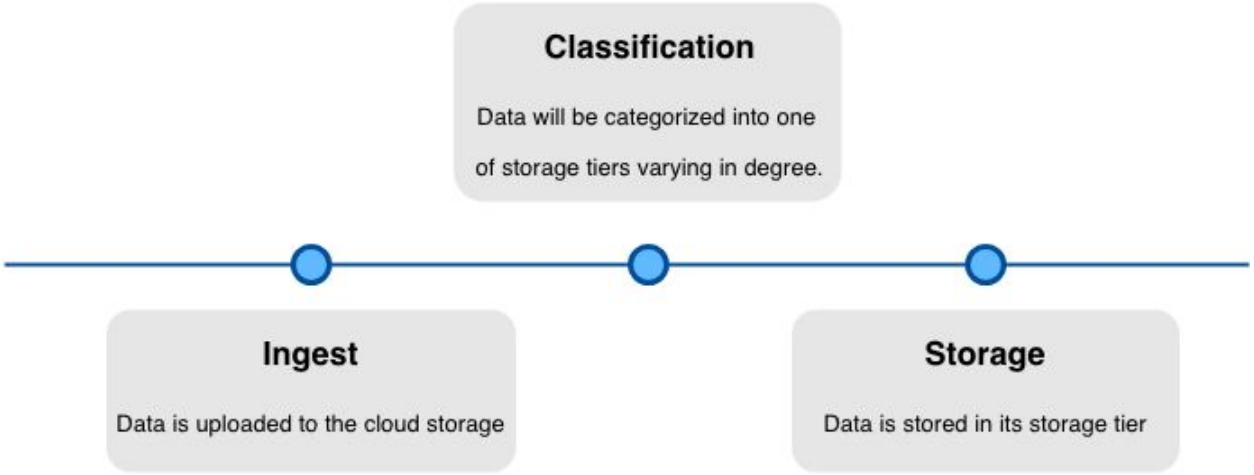


Figure 1. Project Workflow

However, the current problem is our backup administrators at IBM has to manually configure policies to allow for data to demote from hot to cold storage. To give a clear idea of the problems, we have assembled a list of issues that need to be addressed:

- Demotion policies manually configured by backup administrators to demote from hot to cold storage when labor could be used more efficiently
- Data miscategorized and stored in hot storage increases overhead costs
- Data miscategorized into cold storage takes significantly longer to access and causes a worse user experience for clients
- Dealing with miscategorization of large data (petabytes) will lead to a massive increase in costs

3. Solution Vision:

We have been tasked with designing a product that preemptively categorizes data uploaded to Spectrum Protect into its correct storage tier to avoid unnecessary overhead costs. We envision an end-to-end application which will ingest file metadata and use a machine learning framework to create a model which then can be used to classify data into storage tiers. Our solution will be able to:

- Monitor incoming Spectrum Protect metadata streams.
- Read extracted features from metadata, potentially petabytes at a time
- Use a machine learning module capable of classifying incoming data as its correct storage tier
- Gather data as it runs to continually test the accuracy of the model
- Automatically use run-time prediction accuracy results to continually improve learning module

Our application uses metadata extracted from users' data which is collected from IBM Spectrum Protect server. We will use these metadata as our training set in a machine learning framework to create and train a module that is capable of classifying incoming data as its correct storage tier. Our application will finally address our sponsor's problem by automatically and preemptively categorizing the data into its correct storage tier using machine learning module instead of manually configuring policies to demote data from hot to cold storage, which avoids the unnecessarily overhead costs and saves work hours of backup administrators for IBM. Additionally, our application should be able to classify the data that is sitting in the miscategorized storage tier.

If we are able to create a high-accuracy product and it does improve data management for IBM, it can be integrated into the next generation of Spectrum Protect products that serve for all the users around the world.

4. Project Requirements:

This section outlines the requirements of our project. The outline will progress from higher domain level requirements to lower level functional and non-functional requirements. We obtained these requirements in an iterative process. We have had weekly meetings with our sponsor where we have discussed new requirements for our project and refined requirements we already had. We obtained requirements in the following ways: direct assignment from our sponsor, branching off of given requirements, and acquisition through research.

4.1 Domain Level Requirements:

Our domain level requirements outline the bigger picture requirements from our user's perspective. These requirements will be used to create lower level functional and non-functional requirements. These requirements will cover both the expectations of the software and the expectation of the code base as our sponsor will not only be one of the primary users of the software, he will be one of the primary developers improving and maintaining the code base of the software.

1. **The system needs to be secure.**
   Security is of the highest priority as the intended product Spectrum Protect has security as one of the main attractions to the service.
2. **Model creation attempts need to be well documented**
   This capstone is focused on research as much as it is on creating an end product software. Our sponsor is interested in possibly expanding on the groundwork we lay, not repeating it. To best ensure our work is not repeated we need to document our attempts and their results.
3. **The code base should be maintainable and easy to understand**
   Our sponsor is going to be one of the developers maintaining and improving the software.
4. **The system should be easy to run on IBM's clusters**
   A common use case is deploying the product onto an arbitrary cluster IBM has access to for testing or model creation. We should ensure the deployment process is streamlined.
5. **The system needs to have easily alterable features for model creation**
   Our sponsor and users will want to try new forms of model creation and therefore will change the features used for creation often.

6. **Users should be able to easily tell the characteristics of a created model**
   Because there will be many attempts to create models comparing them should be made easier by providing the characteristics to the user.
7. **The system should create an accurate model**
   The core of the project is developing a software that can create a model and researching to find which features create an accurate model.
8. **It should be easy to re-use models**
   It should be easy to redeploy and test a trained model.

4.2 Functional Requirements:

In this section, we will present the functional requirements related to our product. We will begin by presenting a high-level functional requirement and then presenting its lower level requirements. Some of these will be necessary because of our non-functional and environmental requirements.

1. **The system must ingest and process user metadata**
   a. The system must be able to handle large amounts of data passed to it at once
   b. The system will use the analytics engine Apache Spark to process and query data
   c. The system will use Apache Spark to distribute work across any nodes made available to it

2. **The system must produce a model**
   a. The system will use processed meta-data to create a model using Machine Learning
   b. The system will be able to use Supervised Learning and Unsupervised Learning as its learning types
   c. The system will have multiple (at least 3) learning algorithms ready to use for model creation.
      i. Any work necessary for these learning algorithms use will be done so a user can easily select the option from a predetermined list before model creation

d. The system will use TensorFlow to train a model and produce it for use
    i. TensorFlow limits us to the following list of machine learning techniques: neural networks, K-means clustering, Random Forests, Support Vector Machines, Gaussian Mixture Model clustering, and Linear/logistic regression
e. The system will allow for interchangeable features to allow for easier new model attempts
    i. The system will be able to allow for different feature extraction without the need to edit the code base

3. The system must test the model's accuracy

a. The system must test a models accuracy during training
b. The system must test a models accuracy during deployment and improve the model
c. The system should be able to redeploy a model and test the model with a new set

4. The system must inform the user of the characteristics of the model

a. The system should use a GUI to inform the user the characteristics of a model
    i. The system should use a GUI to inform the user of model accuracy
    ii. The system should use a GUI to inform the user of model training time
    iii. The system should use a GUI to inform the user of the training set used to create a model
    iv. The system should use a GUI to inform the user of the testing set used to determine the accuracy of the model
    v. The system should use a GUI to inform the user of the features used to create a model
    vi. The system should use a GUI to inform the user of the learning curve for the model
    vii. The system should use a GUI to inform the user of the receiver operating characteristic for the model
b. The system will use matplotlib for data visualization and will be embedded in a pyqt container window

4.3 Non-Functional Requirements:

This section will cover Non-Functional or Performance requirements. These requirements were gathered from meeting with the client and extracted from higher level domain requirements. The requirements will be presented with a higher level first with any lower level requirements after.

1. **Scalability:**
    a. The system needs to take advantage of the hardware placed on it so that our software is not a bottleneck when training new models
        i. The system shall distribute work across any available nodes during processing of metadata

2. **Security:**
    a. The system will be used with the Spectrum Protect product which markets its security heavily. Therefore, our product cannot compromise the security Spectrum Protect offers.
        i. The system shall have all processing and training be limited to local communication
        ii. Anything saved to disk that could expose IBM's consumer's metadata needs to be encrypted

3. **Performance:**
    a. The system shall create a model with at least 80 percent accuracy

4. **Usability:**
    a. Users should have to interact directly with the code a minimal amount, at least 90 percent of the interactions with the system needs to be outside of the code base when creating a new model.
        i. The system shall have a user interface available to the user for new model production

5. **Reliability:**

a. Our system is being used with large amounts of data and any crashes could cause hours of compute time to be worthless, our system should be as reliable as possible to avoid this.
    i. The system shall be able to handle petabytes of metadata at a time.

4.4. Environmental Constraints:

This section covers environmental constraints. These are constraints outside of our control limited directly by the sponsor or the contents of the project.

1. **Technologies used must be free or open source.**
   The purpose of this project is to create a prototype showing the feasibility of prescreening and categorizing data using a machine learning model. Because the purpose is to show the feasibility of this approach investment costs need to be minimal so we must complete the solutions using only freely available software.

2. **The system needs to utilize machine learning for its solution**
   We must show the feasibility in a machine learning approach for our problem. Because of this, we must use machine learning in our solution. This is not to be confused with us being limited to just Supervised learning as the terms are sometimes used interchangeably we may use any of the major types of machine learning: Supervised Learning, Unsupervised learning, Semisupervised learning, and Reinforcement Learning.

3. **Must have documentation for all model creation attempts**
   We must have documentation outlining all of our model creation attempts. This documentation needs to include all the characteristics of the created model and all major factors that lead to its creation. To document these attempts we will save them in Jupyter Notebooks for

easy replication and organization. This is so IBM can build off of our attempts and research instead of repeating them.

5. Potential Risks:

Our Chosen Machine Learning Type Viability:

Although we have chosen supervised learning as our machine learning method, we acknowledge this might not be the best option. Thanks to our sponsor, we will be able to use a training set to create a model. However, our sponsor has also asked we create a model with about 80% accuracy or more. At this point in time we cannot determine if we will be able to get to an 80% mark with our chosen approach and have taken action to make sure that if we do not attain 80% accuracy with a supervised learning model, we will be able to explore other routes. The technology we have chosen will allow us to explore supervised, semi-supervised, and unsupervised learning in the case supervised learning does not meet our expectations.

Hitting Accuracy Requirement With Any Available Approach:

We also need to be aware of the bigger risk of us not even being able to hit the 80% accuracy mark with any of the machine learning methods. Just like supervised, we cannot be sure that unsupervised or semi-supervised will also provide a model with the desired accuracy. This means that our end product might not meet our sponsor's needs and there might not be anything we can do about it. To reduce the chance of this happening we will be extensively documenting every step of our testing and design process as to make sure we are pursuing every possible option in the most efficient way possible.

False Positive Accuracy:

Another risk is the possibility of us thinking we have a model that will categorize data with 80% accuracy and will be deployed but have far less accuracy than we think. This could even cost our sponsor more than they are already losing. If we are miscategorizing data into hot storage, our sponsor's overhead costs would rise significantly. If we miscategorize data into cold storage, our sponsor's clients could lose money or even their own customers and

at the same time, our sponsor might lose a customer. It also means that the recovery time for the data at a specific location is increased meaning data that needed to be accessed right away now won't be available for an even longer time.

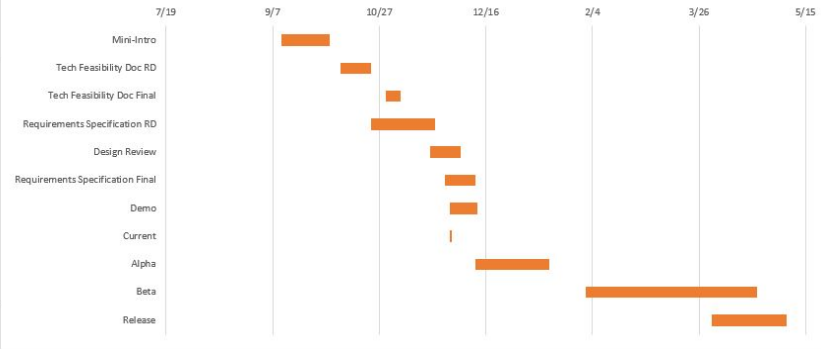
6. Project Plan:

As of now, we have completed the technological feasibility document, have done the design review, and are currently working on the requirements specification document as shown on the chart. Our plan for the next couple weeks is to get the pipeline up and running. This will basically be downloading, installing, and setting up each technology and making sure they work with each other. We are planning on meeting with our sponsor in person for the first time on November 29th. We will be receiving a relatively small data sample to work with for our demo. Although we won't have the data until the 29th, we will be setting up the pipeline before then so when we are in possession of the data, we can start on the demo as soon as possible.

Our plan for next semester consists of three different stages. Our first stage is the alpha. This will consist of the majority of planning and initial testing. Our second stage is the beta phase. The majority of the beta will be mostly trial, error, and documenting. By the end of this phase, we should be getting fairly close to our desired end product. This is also where we might have to explore other machine learning methods if supervised does not meet our sponsor's standards. The final stage is the release. This is where we will be touching up on our end product and hopefully getting very close to the 80% accuracy mark for the models we produce. By the end of this stage, we will, in theory, have completed our end product and have presented our working project to our sponsor.

Schedule

Smart Cloud Shield			DURATION (days)
ROUGH START DATE	ROUGH END DATE	DESCRIPTION	
9/11/18	10/2/18	Mini-Intro	23
10/9/18	10/23/18	Tech Feasibility Doc RD	14
10/30/18	11/6/18	Tech Feasibility Doc Final	7
10/23/18	11/20/18	Requirements Specification RD	30
11/20/18	11/20/18	Design Review	14
11/27/18	12/4/18	Requirements Specification Final	14
11/29/18	12/11/18	Demo	13
11/29/18	12/12/18	Current	1
12/11/18	2/1/19	Alpha	35
2/1/19	4/1/19	Beta	80
4/1/19	5/12/19	Release	35



7. Conclusion:

To conclude, data uploaded to IBM's Spectrum Protect product is often miscategorized into the wrong storage tier. This requires manual intervention by the system administrators to correct miscategorizations after the files have been in the wrong tier for too long. The miscategoriztion of data can lead to an increase in costs to not only our sponsor and clients but can also be a huge inconvenience. The solution we have devised is as follows:

1) Apache Spark will be ingesting the metadata and extract the features from the metadata. We will then use the features in the machine learning framework Tensorflow to create a model using supervised learning which can be used to classify data correctly.
2) The model will then be exported to be used to correctly classify the data in storage and then classify any new data uploaded.

Through this document we have set in stone the different levels of requirements required for our project to be successful, identify risks we might encounter while moving forward, and figuring out a solid plan for the rest of this semester as well as next. We have also made sure that the pipeline of technologies we have chosen meets all of our requirements in hopes of having the most successful end product we can possibly have. Through identifying risks and planning for the future we have made sure that we can mitigate these risks to the best of our ability so that we don't encounter any speed bumps next semester.

With all things considered, we believe we are on the right path to success. With a plan in place and set tasks to accomplish we should have a desirable end product by the end of next semester.