Capstone Project Proposal

Olena Sergeevna Kormachova

Business Goals

Project Overview and Goal What is the industry problem you are trying to solve? Why use ML/AI in solving this task? Be as specific as you can when describing how ML/AI can provide value. For example, if you're labeling images, how will this help the business?	Project: Server failure detection I am trying to solve an industry problem for a web hosting company by implementing proactive fault tolerance management In a normal day to day job, the server failure causes the web sites to go into maintenance mode until the server is fixed/replaced. This can happen any time of the day and creates inconvenience to the web sites owners. By feeding the server logs of previously failed servers and of normal ones to ML/AI algorithm, I can train a model to predict fatal errors in advance. This way, the website owner will get a notification about possible upcoming server issue and can take preventive measurements in the suitable time and day.
Business Case Why is this an important problem to solve? Make a case for building this product in terms of its impact on recurring revenue, market share, customer happiness and/or other drivers of business success.	If the web site owner hosts, for example, e-commerce website, then the server failure might result in the loss of transaction, customer or revenue. Moreover, frequent server issues might leave customer unsatisfied, which it turn can increase webhosting customers churn. That's why it is important to know in advance that there might be a possibility of a server to go down and relocate a website to another healthy server.
Application of ML/AI What precise task will you use ML/AI to accomplish? What business outcome or objective will you achieve?	I will use ML/AI to predict if the server failure is expected in N number of days. This information will help the website owners to schedule a maintenance day and time when it has less impact on their application. And webhosting company will keep their customers happy by not taking their web sites down at random times, in other words, it will increase their customer satisfaction.

	I will feed sample log files with different states of the servers to the model. Those samples will be labeled to indicate whether each state is faulty or not to conduct supervised learning of the model. When model is trained, it can be used on the sample of live system to classify its state as faulty of not. In addition, I will conduct unsupervised learning of the model by feeding the unlabeled files and let the model to find the patterns of faulty systems by itself.

Success Metrics

Success Metrics What business metrics will you apply to determine the success of your product? Good metrics are clearly defined and easily measurable. Specify how you will establish a baseline value to provide a point of comparison.	First of all, I would measure customer satisfaction rate by conducting a survey with questions related to their hosting infrastructure uptime plus the questions about their difficulties which they face when server failure occurs. After I implement the ML/AI model to production, I would conduct the same survey again to compare results. Secondly, I would measure a number of server failures per week/month/year. Hopefully, they will be close to 0 after ML/AI model implementation because infrastructure engineers will be able to fix/replace the server before it fails.

Data

Data Acquisition Where will you source your data from? What is the cost to acquire these data? Are there any personally identifying information (PII) or data sensitivity issues you will need to overcome? Will data become available on an ongoing basis, or will you acquire a large batch of data that will need to be refreshed?	The data for ML/AI model will be sourced from internal server log files therefore the cost of this data will be 0. There is no sensitive data in the server log files which we need to be worried of. For prototyping, I will use two worth of data for 1000 of servers. The logs are event driven which means the data is written to files when some thresholds are reached or specific events occur (startup, shutdown, user logging, etc.). Normally, system administrators restrict the log file size to approximately 1GB and store them for a limited number of days (7 days). So the exact size of the data depends on the events which servers produce. One server might produce 10 files of 1GB size per day, whereas the other one might have only 1 file of 50MB per week. To build a proper failure model, we will require enough historical data that allows us to capture information about events leading to failure. The life span of the servers is usually in the order of years, which means that data has to be

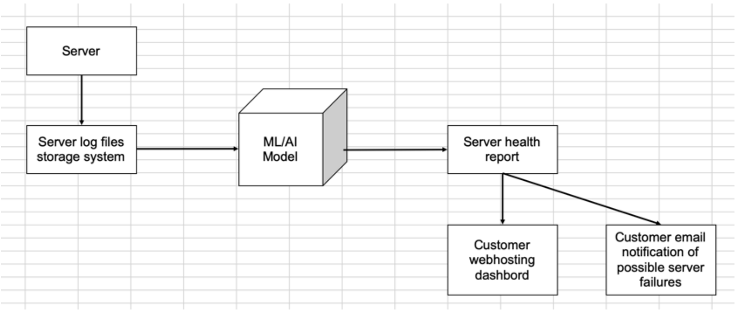
	collected for an extended period of time in order to observe the system throughout its deterioration. I am going to collect the data on a daily basis and keep refreshing the model based on its performance. Since most of the data will be “healthy”, I will identify properly working server and try to limit the number of “healthy” logs from those servers. Additionally, I will try to artificially introduce the faults to generate more positive cases for the training.
Data Source Consider the size and source of your data; what biases are built into the data and how might the data be improved?	The server log data is providing information on the server itself. But there might be other causes of the failure. That is data bias. At later stages I will add data associated with server installation, server configuration, updates/upgrades, and facility information such as power and temperature fluctuations. It may also involve data on the personnel who maintain the servers, such as how long they have been working, how much training they have, and which servers they are responsible for. Some of this topics might include sensitive information from HR department, so I will ask HR personnel to anonymize the all PII.
Choice of Data Labels What labels did you decide to add to your data? And why did you decide on these labels versus any other option?	At the starting point I will use the following labels: - Healthy – will use logs of the normal performing server - Has issues – will use historical logs of the failed server but I will take the logs from a week/month/year earlier date. The weakness of this labeling is not knowing when exactly the server is going to fail and what is the cause of the issue. Because this is the first phase of the project, I only want to train model to detect if server is having issues or not. I want to learn from my model if this is even possible to detect anomalies. Once I am satisfied with this phase, I will consider more complex labels which will show the cause of the issue, such as “HDD failure” or “Corrupted installation”.

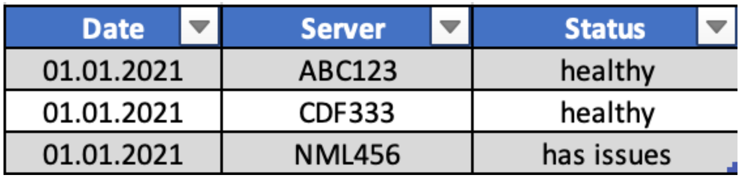
Model

Model Building How will you resource building the model that you need? Will you outsource model training and/or hosting to an external platform, or will you build the model using an in-house team, and why?	First, I will work with in-house team of data engineers do a quick prototype with automated ML tools to build a model and check if I am able to perform what I wanted and if data is good enough for the task. When we are done prototyping and have a clear understanding of the final product, I will use the services of the outsourcing company which specializes in ML/AI to build a production ready model. I will make sure that all sensitive data is anonymized before giving the access to it to external party. Once the model is ready, I will use in-house team to maintain the model and update it if necessary.
Evaluating Results Which model performance metrics are appropriate to measure the success of your model? What level of performance is required?	Our dataset is extremely imbalanced. We have a large number of negative cases (healthy server) and smaller number of positive cases (servers with issues). For the imbalance data, the confusion metric is the best metric in which false positive and false negative errors easily observed. Additionally, we could use: - precision, which will be very low because of the high number of False Positive and the ration of TP/(TP+FP) becomes low. - recall, which will be very low as well because the classifier may detect a large number of positive cases as negative and the ration of TP/(TP+FN) becomes low. - F1 score, which will be very low as well because it is based on precision and recall values. I will try to get the highest possible score which means that the model will properly identify healthy server and those with issues.

Minimum Viable Product (MVP)

Design What does your minimum viable product look like? Include sketches of your product.	For my product I do not have a particular user interface, because the data will be automatically read from the log storage to the model and provide the output to the reporting system. In the reporting system I will set up a trigger to send an email to a customer to notify him if server has any issue.

<table>
<tr>
<td></td>
<td>
I will also flag the server as “has issues” in the customer webhosting dashboard.

Server
Server log files storage system
ML/AI Model
Server health report
Customer webhosting dashbord
Customer email notification of possible server failures

Initial report should have the following information:

<table>
<tr><th>Date</th><th>Server</th><th>Status</th></tr>
<tr><td>01.01.2021</td><td>ABC123</td><td>healthy</td></tr>
<tr><td>01.01.2021</td><td>CDF333</td><td>healthy</td></tr>
<tr><td>01.01.2021</td><td>NML456</td><td>has issues</td></tr>
</table>
At the later stage I might add detailed description of the issue, when it was logged, customer information, engineer information, server specs.
</td>
</tr>
<tr>
<td><b>Use Cases</b>
What persona are you designing for? Can you describe the major epic-level use cases your product addresses? How will users access this product?</td>
<td>This product can be used by webhosting company to detect failing servers and notify the client about need of upcoming maintenance.
It can also be used by inventory team to plan ahead how many new servers/parts they need to purchase for coming month/quarter.</td>
</tr>
<tr>
<td><b>Roll-out</b>
How will this be adopted? What does the go-to-market plan look like?</td>
<td>For pre-launch I will gather the data, interview infrastructure engineers, conduct customers satisfaction survey.
For post-launch I will monitor the model performance and keep improving it if necessary, conduct post-production customer satisfaction survey, search for new data to be added to improve the model.</td>
</tr>
</table>

Post-MVP-Deployment

Designing for Longevity	Not all servers might have sufficient log files available (lost,

How might you improve your product in the long-term? How might real-world data be different from the training data? How will your product learn from new data? How might you employ A/B testing to improve your product?	corrupted, etc.). Plus there are might be new patterns of failure which were not present in the training set. I will keep updating the model to make sure it learns from the new cases. Additionally, as mentioned in the data source section, I will add server related and personnel data. We can train our new model with new data and use A/B testing by pushing 80% of the data to old model and 20% to new one and then measure how the new model is performing. If, after few iterations and checking all the metrics, the new model is performing better than the old one, we can implement the new model.
Monitor Bias How do you plan to monitor or mitigate unwanted bias in your model?	I will keep human in the loop to analyze the model and keep improving its performance. I would also work closely with data scientist and infrastructure engineers to find a root cause of previous failures to make sure I train the model with the contributors to the collapse of the system.