The data science department of a leading US-based entertainment/media corporation with revenue over $10B selected Amazing's Jelastic MapProduce (JMP) over DataB for its production predictive analytics solution.

Winner	Amazing	**Offering / Deal Type**	Predictive Modeling
Industry	Media	**Region**	US
Competitors	Amazing, DataB		
Date of Interview	June 2018	**Date of Decision**	October 2017
Persona	Technical	**Title**	Senior Data Scientist
Purchasing Role	Recommender	**Level**	Individual Contributor
Company Revenue	$10B	**Anonymous**	Yes

Context

- **This is a sample winloss interview. It was uncommissioned and it has been anonymized. The names of the vendors have all been changed. The interview is abridged, and it illustrates the conversation nature of our interviewing approach.**
- **This company evaluated Amazing JMP and DataB for its production machine learning solution**
- **The company selected Amazing's JMP over DataB for its production needs, because DataB creates a lock-in problem**
  "DataB actually make the data Speed and Scalability side of things really quick and really pleasant. If you just want to look through like huge datasets, just fire it up and do it and it's so easy. It's so great. But then it comes to building something for production, that's where getting locked in becomes a much bigger issue in my mind. That's where we're doing production in JMP, but then like surface stuff and DataB."

Favorable to Amazing Jelastic MapProduce (JMP)

- **Amazing JMP provides flexibility with a development platform that supports other tools and third-party libraries**
  "They've given you unlimited flexibility to do what you want. Then I could go in and I can bring in extra tools and I can augment what they're doing, but that's work, that's extra layers of complexity."
- **Amazing JMP has no "lock-in" because your apps run on Sparky, whereas DataB apps are limited to running on DataB**
  "I think with minimal changes, all of my code (on Amazing JMP) could run on a Sparky cluster that I

configure myself. Everything that runs on writing something for JMP is like writing it for any Sparky cluster. (Whereas) writing for DataB you're just writing for DataB."

- **Amazing JMP offers more machine configuration types than DataB**
  "JMP I think there are maybe 20 or 30 different machine types. You can make really small machines, you can make really large machines; it's all there. With DataB, you're limited to a subset that they've determined are the best ones to use."

Unfavorable to Amazing Jelastic MapProduce (JMP)

- **Amazing JMP lacks a quality visualization tool compared with DataB**
  "(JMP) has an add-on which lets you use Jupyter Notebook, which is great because it's open source. It's half as good as DataB Speed and Scalability. Jupyter has sort of an example of (a custom display function), but it's very limited. It's not very intuitive, it's not very quick."
- **Amazing JMP is getting old; for instance, its YARN scheduling is getting to be outdated**
  "JMP product that AMS has, it's old. It was originally developed for MapReduce and either just table the recycle of framework and use it for Sparky clusters as opposed to MapReduce clusters. It's old and it depends on what's called the YARN scheduling, which is also getting a little bit outdated."

Unfavorable to DataB

- **DataB is innovative, but your apps will only run on DataB**
  "With every new feature it's more lock-in. That's really this problem of this really innovative company that's doing a lot of things to really make things easier, but with every innovation, the more you will depend on DataB, which is ultimately what we really want to avoid. Writing for DataB you're just writing for DataB."
- **This respondent was concerned that DataB may retire an API, which will force them to rewrite code**
  "DataB are coming up with the new stuff and they can do whatever they want. In fact, there are some points where they'll be retiring old APIs and be like, you can't write this code anymore, you have to change, because they're iterating. It's really a contrast of two different styles there."

Favorable to DataB

- **DataB wins hands-down for its ease of use, and it has a short learning curve**
  "DataB is a bit like an iPhone. Everything's already set up for you and you're just supposed to just use it, where JMP is a little bit more of a learning curve. You don't even have to think about launching a cluster with DataB. It's just three clicks and you're running data. DataB is easier to pick up. There's no doubt about it."
- **DataB has fantastic visualization features that are easy to use**
  "The Speed and Scalability and visualization is hands-down better on DataB. There's no comparison there. It has this custom display function where you can just throw it some data. Then there's a lot of flexibility with what you can do for plotting it. If you're doing Speed and Scalability and you just want to

look at something, you don't want to think about it. You just make it easy. That's what DataB will do for you."

- **DataB' DBFS speeds up execution**
  "They've built this extra layer called, I think it's like DBFS, which is like the DataB file system, and it does a whole bunch of stuff. It's supposed to speed a whole bunch of things up. It makes a lot of things easier, but it only exists on DataB. They've come out with Delta, let me double check the Bricks. Yeah, it's a data management system they call it, and it works. It's actually like it's really well designed. I think it's a great product."

  <<ABRIDED>>

Summary of Scores

			Winner	
	Importance Rating	Gap	Amazing	DataB
Breadth of Use	10	**4**	**10**	**6**
Avoid Lock-In	10	**3**	**9**	**6**
Cost	8	**2**	**9**	**7**
Ease of Use	5	**-3**	**5**	**8**
<ABRIDGED>		**…**	**…**	**…**
<ABRIDGED>		**…**	**…**	**…**
Overall Weighted Rating			**…**	**…**

Transcript

What motivated your company to look into solutions in this space? What was your business need?

What motivated your company to look into solutions in this space?

The driver of looking at or comparing these vendors is really trying to find products that can do predictive analytics at scale. When I say at scale it's not just about just doing one run of big data, it's being able to scale up when needed and scale down when needed, so flexibility to process a 100 million records or a million records to the platform, it doesn't matter. It takes care of all of that. That's traditionally been quite challenging because predictive analytics of modeling in particular tends to be really really CPU intensive.

And so you have to come up with all sorts of clever solutions or just a lot of engineering to be able to scale this stuff up, but that's where Sparky is really interesting because it takes care of a lot of this stuff behind the scenes. And so then it came down to comparing well, how can AMS provide us with a platform to run Sparky versus how can DataB do that. So, it's really about finding a platform that can provide us huge scale in a flexible way with very little friction so that we can just run it without having to have an army of engineers.

And so, to get directly at your question about the types of problems, we have just a huge wealth of data and information and not enough resources in terms of this is like most companies in terms of personnel to look at it at all. Every company out there I'm sure wishes they had 100 times more technical staff than they do. And so, the business problem here is, actually being able to do really good predictive analytics on everything we have and so finding tools that can do that for us quickly is really quite valuable. If it can save the data scientist time in terms of implementation, then that's a big win.

I would imagine in your case, you're in the entertainment industry. I was thinking that your applications for predictive analytics would include anything from figuring out what titles you want to produce or what the content should be about so that it would be interesting to your consumers, but also maybe what to advertise to them, or to offer recommendations such as you watched this maybe you'd be interested in this, or things like that. I would imagine these types of use cases would be amongst some of the kinds of things that you would be doing with predictive analytics?

Yeah, and part of it just to maybe color a more diverse picture is that, we're a media company and we deal with everything. We deal with a streaming, we deal with cable, we deal with social on the Internet, we're everywhere. So, that means we've got many different types of data all coming in and none of it looks the same because it's all, the type of data coming in from website usage is just completely different than what you'd expect to see in a more traditional cable environment, you know, sort of the data coming in from different reporting companies.

So, again that creates its own challenges, but that's part of finding platforms that are really really really flexible, that they really come in with as few assumptions as possible about what the data looks like so that we can go in and do what we need.

And so, then Sparky, I mean it's interesting because it can allow you to handle all different types of data, right?

Yeah.

I know, DataB looks interesting. I looked that over and it made me think of Cloudworks and Hortoncloud, two other vendors, that are kind of similar.

Yeah, very similar space.

Yeah, and then whereas Amazing has, if you will, the infrastructure that you can run everything on the servers, the nodes, clusters, all of that stuff, the elastic scalable environment.

That's right, yeah.

Yeah, because I had been originally thinking people were installing Sparky and I guess this is going back a few years now, but people were using their own commodity servers and things like that. They can suddenly distribute this stuff.

Yeah.

Now I think what's happening is now that everyone wants to move all of the hardware to the cloud because they don't want to deal with this. It's cheaper that way I guess.

Well, yeah, it's kind of taken an interesting turn where you're absolutely correct, that was the data engineer's job three years ago, two years ago, even six months ago in some companies, was to sit down and connect these clusters and build the things. With the advent of what DataB has been selling and what JMP does, it's very much kind of Software as a Service to take care of all that configuration and all that complexity of what the engineer would normally need to do.

Yeah, now in this case, you ended up going with both, and I have a feeling that maybe you even end up using them together, but I'm not positive.

Who were the vendors you evaluated?

	1	2
Name of vendors	Amazing JMP	DataB

Yeah, so there are reasons for that, and just to dive in a bit deeper. It's two sides of a fence and this thing it's kind of complex. DataB actually uses AMS for their infrastructure. DataB doesn't own any bare metal. So already the one is dependent on the other, but then DataB is a company that was set up to actively promote, market and develop the Sparky programming language. So then AMS is dependent on them for that, right?

There's this funny loose interdependency between the two, even though they're competing in a way. There's also the aspect of open source, that was a huge concern for me and for my team because getting locked in it to any one platform which is, I think that happens in all companies when you get locked in, it can really cause problems, and especially in a data science environment where flexibility is paramount to being able to do interesting things. Getting locked in is just the worst-case scenario.

So really prioritizing open source solutions. But so on that note, DataB open sources most of Sparky code, maybe 99% of it, but then they have a few extra bits that come with the DataB subscription that aren't open source. That's their tools that you pay to use. And, it works the same with JMP. Most, say 99% of it, is open source. You could take the same code and run it wherever you want, but then there's this 1% of special stuff, which again, you're paying for. In comparing the two, a big part of it is just trying to figure out where am I not getting locked in, and both platforms lock you in in their own way. I forget now how we started on this.

I was just actually asking a little bit about the fact that, you started by looking at different products and you thought you were going to do one purchase, one or the other. Now, you ended up, which is not atypical, it's common where people get more and more educated, where they put together their own architectures, right, of solutions. That's what it sounded like you have, you were saying a little bit about the requirements for your solution.

Which vendor was selected?

Amazing JMP

Yeah, thank you for that reminder. Where we've sort of landed between the two is that AMS (JMP) wins by a narrow margin in that it's got better open source extensibility, but that's by no means a hard statement. But essentially you can take your Sparky code that runs on JMP and take it

somewhere else, and it will run completely fine. That's not the case for DataB because they've got a lot of these extra layers of bits that you pay extra for.

So, it's important though because those extra bits and tools and utilities, and DataB actually make the data Speed and Scalability side of things really quick and really pleasant. So, depending on what you need to do, so if you just want to rip through like huge datasets, just fire it up and do it and it's so easy. It's so great. It's kind of like what you pay for but then when it comes to building something for production, that's where getting locked in becomes a much bigger issue in my mind. That's where we're doing production in JMP, but then surface stuff in DataB.

Okay, so you're using DataB to explore, visualize, figure out what you might want to use in your models and understand your data. Then using the other tool to produce your solution, actually build your models, manage your data and that's a little bit more open if you will. It works out pretty well because you can always slide in a new visualization piece and it wouldn't affect all of your running models and things like that.

Yeah, that's just it, when you're scripting and analyzing data, it's all sort of improvised in the first place. There's no, you're not spending hours and hours and hours developing that code, but then production, you have to get everything right. It's quite a bit more complex so that's why that's on JMP.

<<ABRIDGED>>

Yeah, okay, that's a slip. Deeply we use AMS, the whole company is AMS, and then so what happens is once you've kind of hooked into a cloud provider, they're quite clever about it. Generally, all the data you transfer within a cloud provider you don't pay extra for. If I've got like a terabyte of something that I want to run a Sparky cluster on, if I do that all in the AMS ecosystem, I don't pay for that data transfer on the terabyte, which can be substantial over time.

So given that all our data is already in AMS, there was no option to really look at like Giggle Cloud, and then like Magenta and Giggle are all starting, I think they're starting to offer, if not yet, soon, their own in partnership with DataB. Then it becomes this complexity of what's the platform, what's the service?

Very good. So when were you looking at these solutions and comparing them? When did you start looking and when did you make a decision?

When did your company finish your selection?

February 2018

When did your company start your selection?

October 2017

Yeah, I think maybe it started about eight months ago and then we made a decision about four months ago.

All right, so it took about four months to decide, if you will?

Yeah.

Then in your selection, was there a selection committee, or how did your process work?

What was the composition of your selection committee? What roles were on it?

The department I work in, we're a small agile group within a larger organization. It's this funny thing where both are available to us through sort of corporate contracts.

What was your role in the selection process?

Recommender

Then so the decision of what we're going to use within the department ended up being made by me because I was the one doing the development, the feet on the ground type of thing.

Then is your department like a data science resource that the overall company uses or that type of thing?

Yeah, the company has, either some companies will do one big data science group for the whole company or you'll do sort of many small ones embedded within different departments. That's the second is what it's like here.

<<ABRIDGED>>

Okay, all right, good, and then your title is senior data scientist, right?

What is your title?

Senior Data Scientist

Yeah, senior.

Then how did you develop or source the list of vendors that you considered, maybe these two now in this case?

A big part is colleagues, so I know that these are similar problems that a lot of people are going through. Yeah, the DataB in particular had come with really high recommendations from a few different colleagues at different companies. Then just spending time and doing research on my own, just really doing a deep dive on what—

Web searching and that type of thing to allow you to learn?

Yeah.

Not particular conferences, or do these vendors all try to contact you and say, “Hey, we've got this, you're interested in hearing about it.”

When I was looking at both of these options, there were no conferences that had come up, otherwise I would have absolutely used that as a resource to talk to them. In both cases I've spoken with the AMS team and I've spoken with the DataB team and they're both extremely outgoing and helpful in offering all sorts of come in and do training and whatever we want. In terms of support and answering questions and all of that, both the companies were extremely responsive.

Great, and you must have had solutions. Was this solution replacing a solution?

Was the selection for a solution you were replacing?

In a way at our company there's currently a use of, I don't know how to put it, but there's a tool that aggregates and structures and has some BI functionality that was originally envisioned to be where all the data's is going to live and everybody's going to love it. It doesn't really do anything in terms of predictive analytics, so we've been working out independent of that initiative to figure out how we can move forward. DataB and JMP was one of the things that came out of that.

Okay, good, and did you use any industry analysts, system integrators, outside consultants as part of your selection process?

Did any outside consultants contribute to your selection process?

No

No, this was entirely internal.

<<ABRIDGED, interviewer asks for key criteria>>

Sure, yeah, there are definitely a few that come to mind. JMP is slightly cheaper.

Okay, cost.

It's slightly more flexible.

Okay, flexibility, which means exactly what in that regard? It means, so data or the way you configure it?

Yeah, configuration, customizability and configuration. You can sort of support comparison but the DataB is a bit like an iPhone. Everything's already set up for you and you're just supposed to just use it, where JMP is a little bit more of a learning curve, but you can pretty much change whatever you want.

Should I put in something like learning curve or?

Yeah, that could be a consideration for people. DataB is easier to pick up. There's no doubt about it.

Okay, so ease of use?

Yeah, DataB wins by a long shot for ease of use. In this case, because we were going to be using it long term but not as much as a concern.

Okay, What about lock-in? You had mentioned that in your discussion.

In my opinion DataB has a much a higher lock-in and this is only getting worse as they release new features. Where AMS to a degree has a bit of lock-in but for the most part, everything that runs on writing something for JMP is like writing it for any Sparky cluster. Where writing for DataB you're just writing for DataB.

It makes sense. Was there any other, did you think one was better for data Speed and Scalability or data discovery, or whatever you want to call it?

Yeah, that's right. The Speed and Scalability and visualization is hands-down better on DataB. There's no comparison there.

Okay, what about support? Is that something that you would compare the vendors in terms of their ability to support, and did you feel that one was better than the other? Or did they turn out to be about equal?

I wouldn't say they're equal. I think we've got kind of an enterprise AMS support plan, so it's always going to be a super attentive, but DataB has been great too. Definitely equal in that respect.

What about things like vision of where the vendor is going or where the product offering is going? Sometimes especially where things are changing, you really feel like someone's got a lot more power and a vision that's really unique that makes one stand out over the other. Was that a concern?

You know what, I think that's a really interesting question too because DataB is driving forward the development of Sparky where this JMP product that AMS has, it's old. It was originally developed for MapReduce and either just table the recycle of framework and use it for Sparky clusters as opposed to MapReduce clusters. It's old and it depends on what's called the YARN scheduling, which is also getting a little bit outdated.

They can't change because there are so many legacy products built on JMP the way that it is, the assumptions it has. You've got, which can be good, extreme stability on the JMP side, very minute change, versus DataB, who are coming up with the new stuff and they can do whatever they want. In fact, there are some points where they'll be retiring old APIs and be like, you can't write this code anymore, you have to change, because they're iterating. It's really a contrast of two different styles there.

I know at the beginning you were giving me your criteria. We've got flexibility. Was scalability different between these products?

No, identical.

Identical, okay. Was there anything else in your mind that was different? You've got ease of use. I think that probably includes the notion of how many developers do you need to help you for each these products. I know you might just have to pull in developers for something.

Yeah, before these products existed, you would need a data engineer like without hesitation on the team to be able to set these things up for you. That's no longer a need.

Do you think we covered all of the criteria?

I think so.

You can always add something if needed or as it comes up, but go ahead.

Yeah, there is just sort of the one theme, where DataB is doing a lot of interesting things and they keep coming up with new features, and they've got this new data format which looks amazing, but with every new feature it's more lock-in. That's really this problem of this really innovative company that's doing a lot of things to really make things easier, but with every innovation, the more you will depend on DataB, which is ultimately what we really realy want to avoid.

Right, yeah, they're in top spot for that because I'm sure they could come up with the innovation, but they don't want to just release it and make it available to everyone they compete with, you know.

Exactly.

All right, let's keep that as the criteria if something else comes up, we'll add it. Let's do this, let's go through each. We've got six items, six criteria. We'll give each an important score between one and 10. One is not important, 10 is very important. How important was cost to the decision?

Criterion	Cost	Importance	8

I would say eight.

An eight. What about the flexibility, customization and configurability capabilities?

Criterion	Breadth of Use	Importance	10

I'll put that at 10, that it was paramount.

Yeah, and ease of use.

Criterion	Ease of Use	Importance	5

Maybe five.

It's just a five. Let's just define this a little better. How was flexibility, customization and configurability, different than ease of use?

Flexibility means you can go in and change whatever you want. Ease of use means you turn it on, and you can just start doing things.

Okay, so one is really ease of getting started and generally how is it to use. The other one is like, it's very flexible. You can make these changes whenever you want, wherever you want, that kind of thing.

Yeah, and one generally comes at a cost of the other.

Yep. And what about the lock-in, how important was it? I'm going to call it avoiding lock-In.

Criterion	Avoiding Lock-In	Importance	10

Yeah, that's a 10 too.

That's a 10 too, avoiding lock-in. You guys can work with whatever comes out in the future. Speed and Scalability, visualization capabilities, how important was that?

<ABRIDGED>

Okay, all right, good. Then one of most important things was flexibility, customization and configurability you called it. How good do you think JMP is at that between zero and 10, zero is terrible and 10 is great?

Breadth of Use		
	Rating of vendor	
Importance Rating of Criterion	Amazing JMP	DataB
10	10	6

For flexibility, customization, it's a 10.

Okay, what about DataB?

Six.

A six, and can you give me an example of how DataB… What's the reason for the gap that maybe you can give me an illustration or example.

Sure, so just in a very simple case on JMP I think there are maybe 20 or 30 different machine types. You can make really small machines, you can make really large machines; it's all there. With DataB, you're limited to a subset that they've determined are the best ones to use.

The most common ones, that type of thing.

Yeah, they just say, "Oh, our engineers decided this is the best. These are the optimums, you should just use these. Don't use the other stuff. You can't use it."

Okay, that was a great example. All right, now, avoiding lock-in was also important. That was a 10. How good is JMP at avoiding lock in?

Avoiding Lock-In		
	Rating of vendor	
Importance Rating of Criterion	Amazing JMP	DataB
10	9	6

Yeah, I think with minimal changes, all of my code could run on a Sparky cluster that I configure myself.

Pretty free to do what you want with it then. What would you give them for a number then between zero and 10 for that?

I would give it a nine.

A nine, and what about DataB?

There, yeah, maybe it's getting lower. It depends; let's say it's six for now maybe five soon.

Okay, six I put in. Is this the kind of thing, we can just avoid this library, or is it kind of, oh, it becomes part of the overall solution so much that you get entangled with their proprietary stuff that you can't really separate yourself?

Yeah, if you're motivated you can avoid it, but then you're paying extra money for nothing.

Yeah, well, can you give me an example of something that you didn't like, that you felt there's lock-in for this even though it might be a good feature or something that's cool, but we don't want to go there?

Yeah, so they've built this extra layer called, I think it's like DBFS, which is like the DataB file system, and it does a whole bunch of stuff. It's supposed to speed a whole bunch of things up. It makes a lot of things easier, but it only exists on DataB.

In that case in terms of being locked in, would it just be a matter of moving your data off of that into a different file system or is this some of the implementation of what you write then depend so much on the file system that it becomes broken and you can't migrate off of it?

Yeah, the latter.

The latter. Okay.

Yeah, so you can but you'd have to make changes because all your assumptions are different. The code would change in certain places.

You're stuck with that once you start using that file system?

If you get into it, yeah.

Yeah, that's an example. Anything else in terms of the lock that you think is unfortunate?

<<ABRIDGED>