Making Data Science Simple

IBM Code Tech Talk
Oct 18th 2017


>> MARC-ARTHUR PIERRE LOUIS: Thank you, Kathy. Good morning or good afternoon, wherever you are, my name is Marc-Arthur Pierre Louis, your friendly moderator for these tech talk series. We have today a treat for you, a great topic, making data science simple. Scientific literacy.

Reach solution for tomorrow's problems. We need to make science simple so solutions can flow. People may think that data science involves a lot of math, a lot of statistical analysis. But the key is to make it simple. When I was in linear algebra class once, the professor was doing a lot of proofs and theory, heavy on the theory side. One guy said, professor, when are we going to do problems? He turned around and says, well, my friend, we are solving thousands of problems right now with our theory.

Yes, the theory solves the problems but making them simple, problems to solve, is the key to make it, come to down to earth and that is what we are aiming to do today in this presentation. We have our data signs guy, David Taieb, who is going to take you to how data science works and through the IBM data signs experience. Then Scott is going to come back and show you a real life data science and application and then Dave is going to come back.

I don't want to take from their time. I'm going to turn it over to David Taieb for furtherance this have great presentation. Enjoy. You have it, David.

>> DAVID TAIEB: Thank you, can you hear me okay?

>> MARC-ARTHUR PIERRE LOUIS: I can hear you.

>> DAVID TAIEB: Excellent. I'm trying to, yes, here is the agenda for today. We are going to take a few minutes to talk about data science and specifically about the pains and promises of data science especially in the enterprise world. We are going to talk about the Watson data platform and how Watson data platform tries to live up to the principle I'm going to enunciate today during this presentation.

We are going to have a overview of PixieDust which is an open source library that helps making data science simpler. Then Scott is going to take it away with a walk through of the
developer journey, analyzing San Francisco traffic data from open data with Jupyter and PixieDust.

Then I'm going to come back and show you a sneak preview of a brand-new feature that I found very exciting and promises called pixie gateway that allow you to take your analytics to the Web.

Let's start right away with a claim that I hope everybody can agree with me, is that data science is a team sport. It involves many disciplines. I would say it's not even a linear process. It's a lot of steps, lot of terms, sometimes expected, sometimes unexpected -- turns. It is very iterative in nature.

That creates a lot of problems within the teams, because you are spanning when you build data pipelines a lot of disciplines. We are going to look at what are those pains, what are those promises, and how we can start finding solution to them.

What do business care about when you come to data science? You probably are familiar with the concept of digital transformation and how companies are trying to be more data driven, and when they think about data science, the first thing they want is use it to gain competitive advantage, to better understand their market, their customers and their competitors.

They also want to be able to use data science to discover new insights, and maybe create new business models and new revenue streams. They also want at the same time to be able to be data driven in the decision-making, and they want it to be very fast, close to realtime.

They want all this, while at the same time they want reduced complexity and lower cost. This is really a tall order, and at the same time, they want to be able to have an accelerated time to market and deployment of the analytics so that they can leverage it and use it appropriately and deploy it to their customers.

If I were to assume what really they care about, is they want innovation and insights at speed. All of this is nice. What are the pains? Well, there is a lot of those. The first one that comes to mind when it comes to data science is the shortage of experts in those fields. According to a survey data science has become the top skills sought after in the job market. There is also the problems of once, if you are lucking to have an established team of data scientists, they spend 80 percent of their time finding, cleaning and organizing data, and only 20 percent on real data analysis.

That leads to frustrations and they have to deal with that. There is problems of the tools themselves that are overly complex and inaccessible. We need to democratize them. Finally there is also the lack of data governance and cataloging. When a data scientist is able to find good data, they ask themselves whether they have the right to do it, whether it's the right
data, is it up to date, is it curated. That is a problem. There is the problems of prototyping, a data scientist in nature is exploratory. His job is to explore, to research, to try to make hypothesis, but at the same time, a developer, his job is to work within a set of very well-defined constraints.

We have this dual forces here that quickly becomes a problem. That leads to a lack of collaborations, across all the business and across the developers and data science teams. We have seen it firsthand with scientists, with customers and in our team as well because we build on the day-to-day, end-to-end data science projects -- prototyping versus productizing.

If I have to assume this pains are obstacles to insights at speed. When I think about data science, I'm a TV buff and I want to do a metaphor to explain. When I think about data science today, the way I see it, I see it as a game of thrones, whether it's exciting, alliances being made and undone based on the task of the day. What we as a developer, as a team wants, to be able to identify with one of those, we would like to be more like Jon Snow, if you know what I'm talking about. Jon Snow is trying to unite those people to say you are going after the wrong problem. The real problems is the nightwalkers, in our case, solving data science.

How do we get there? Where do you want to be? We want to be more like friends, right, this is cute and beautiful, this team is working together, but they really understand each other. More importantly, when I look at them, what makes me think is they have fun, having fun into the data science world is really important as well.

Okay. Enough with the metaphor. Where do we start? Where do we start? When we think about the problems that I enunciated, we need to think about platform. There is three pillars in that platform that we need to think about. The first one is tools. Make them simple, make them standardized, clear input, output between them, and make them integrated.

So all of this will lead to better collaborations and breaking the silos between the different teams and disciplines.

The second pillar is services. As much as possible, we need to be, to go into a service-oriented architecture, to abstract and harness complexity of the past, so it can become scalable, because theoretically you need to be able to scale with big data and one suggestion here is the Cloud, with its infinite amount of compute and resources.

Finally, assume as much as possible, try to leverage open source, open source is a key component in what you do in data science. All of this leads to more reliable and repeatable data science.

The final pillar I would like to propose today which is
important is data. You need to have a data strategy. You need to build curated content so that, and cataloging so that data scientist doesn't spend too much time trying to find and clean this data.

You also need to have governance so that when the data science, when the data scientist is able to see a data set, they are allowed to use it. They don't have to think and wonder about permission. If they see it, they can use it.

Finally, you need to be able to make those data sets easy to share, build a community. All of this leads to better productivity, because better, less frustrations and better morale for the data scientists.

That brings me to the IBM Watson platform. All the design principles, we try to live with them when we build and design the IBM Watson and Cloud platform. It has multiple layers. It starts with the Cloud, based on IBM Cloud. There is a layer which is the IBM Watson data platform which is what I'm talking about to you today, there is the AI layer, which is services, conversation, reasoning, learning and so on. Finally there is the application layer where we try to be industry verticals and build cool innovation there.

Let's talk about the data platform for this topic. It is an integrated platform of tools, services and data. See those three things, tools, services and data. We try to lead to our principle of making them simple tools, service oriented when needed and a very strong data strategy based around the Watson data platform catalog which I'm going to talk about here.

You see all those personas here, they collaborate. The data platform is built and designed to foster collaboration between those five personas.

Let's talk about the flagship components of the Watson data platform which is the data science experience. Data science experience is a ID of sort that uses Jupyter notebooks as its main tools, that allow you to easily load data, access all the services that are available in Watson data platform and do your data science.

It provides other services as well like machine learning, Watson machine learning. It integrates our studio and it tries to be an inclusive as much as possible to integrate the data into many places they can come from. They also integrate with the Watson data catalog so that once the data is in the Watson data catalog and curated, you can easily load it and start analyzing it and process it and so on.

Finally it's built on the IBM Cloud platform.

There is the projects which allow you to work and collaborate very nicely between teams. It's like I say, it allows you to create connections to your data source, add data sets to the
catalog, and then access all the different services that are available. All of this while maintaining, you are able to bring collaborators and give them roles so that they can do what you allow them to do. We are going to do a preview of that while we are doing our demo today.

DSX is also providing machine learning. You can build, you can make it easier to train your models, to build your models, and to deploy them into the Watson machine learning, and build recipe-ize around them so you ask and easily invoke them with a simple API, and start scoring, using those models to score your data and start using it. It also provides you with a Wizard that assists you in the creation of training and models, using very common patterns and algorithms. It makes it easier for you to train, to test, and to iterate over your model by fine-tuning your data and parameters.

Finally, the last component I want to talk to you today about is the catalog. There is many others but we don't have time to go over them. The catalog is a very important piece. It is the data pillar that I was talking about before. It provides secure and data protected access. You are able to audit your data, to shape your data, to make sure that you trace the lineage of where it comes from, and it provides you with a flexible way of setting governance policy.

The catalog is a integral part of the DSX and everything, all the services add access to the catalog to be able to load the data and store the data back.

There is a very nice UI integrated with the data science experience, where people can browse their assets and based on their permission, start searching for it and load them, loading them into whatever tools you are using. So whether you are data scientist, data engineers, business analyst and so on, you can access the data catalog to access your curated data and start working with them.

>> MARC-ARTHUR PIERRE LOUIS: David, I have a quick question for you. Somebody might want to know where to find Watson machine learning infrastructure. Can you talk to that briefly?

>> DAVID TAIEB: Sorry, where to find it?

>> MARC-ARTHUR PIERRE LOUIS: Yes, the machine learning, the Watson machine learning infrastructure.

>> DAVID TAIEB: Watson machine learning is a service on IBM Cloud, that you can instantiate directly from DSX or go to the BlueMix IBM Cloud catalog, you can go and look for it into the catalog. There is a search. You can instantiate it. It is a free plan. Once you have provisioned that service, you will be given a set of credentials. You use credentials afterwards once you built your model, so you can build your models with let's say Spark machine learning or Python psychic learns, it doesn't
matter what machine learning support many formats. Once you build them we provide you with simple APIs to upload those models into the Watson machine learning, which in turn will give you a end point. This end point can be used at runtime to score your model.

So, it's easy to build, upload and deploy your models, and then use them within your application. We have a lot of content and blog post that sample application that use Watson machine learning.

>> MARC-ARTHUR PIERRE LOUIS: Great, thank you.
>> DAVID TAIEB: That brings me to PixieDust. PixieDust is an open source library that makes it easier to work with Jupyter notebook. Jupyter notebooks are powerful tools, popular in the enterprise. It's growing. But it can sometimes be felt as a data scientist tool, where what we claim is no, it is not only for data scientist. It can also be a nice complement to developer toolbox.

The first thing we did with PixieDust is build a bunch of simple APIs. There is a simple API to import your data, so that you don't have to learn Spark, Apache Spark APIs. There is a simple API to visualize your data and explore it, and create interactive charts so you don't have to learn bouquet or C born or Google maps. Here I show you an example of two lines of code that loads your data and display it, two lines. That's it.

Then finally PixieDust also provides you with a way to build powerful application that runs directly into the notebook, with minimal coding. It is called a pixie app, that allows you to build apps that run directly in the notebook and those are powerful. I show an example here of a dashboard that was built with a business partner called P stack that shows you a dashboard of indicators for conflicts in the Middle East.

So what I'm going to, what Scott is going to show you next is a end-to-end data pipeline using Jupyter notebooks and PixieDust. I'm going to come back with a treat, with a sneak preview of how to take those charts that you will build and those dashboards to free them from the notebook, liberate them and make them as Web pages, Web app that anybody, line of business user can use without the need to go into notebook. Stay tuned for that. It is going to be a very cool announcement.

So now I'm going to turn it over to Scott for the demo on the PixieDust.

>> SCOTT D'ANGELO: Thanks. We have got a code journey that allows us to create charts and graphs and maps that analyzes traffic data from City of San Francisco which provides a great open data set. The code journeys live at developer.IBM.com/code/journey. We have a lot of content out
there that includes complete code, documents and all the instructions you will need to sample a small proof of concept, all the way to getting some kind of idea of how you can actually provide some type of application into your own enterprise needs.

This is the sort of top level of the journey. It always includes a open source GitHub repository. That is where we are starting. This contains all the documentation you will need from start to finish, some architectural diagrams and such and usually a video. You will follow the steps and it will allow you to -- yes?

>> You are not sharing your screen.
>> SCOTT D'ANGELO: I'm not sharing my screen.
>> MARC-ARTHUR PIERRE LOUIS: No, you are not.
>> SCOTT D'ANGELO: Sorry. I missed the last button. Thanks for cutting me off (chuckles).

We do have the top level journey and then we have our GitHub repository, and following these steps will get you up and running quickly.

We in IBM data science experience account, there is a free tier, you can log in, get that going. You will follow the steps to create a new notebook. We will type a notebook name. Pixie traffic, optional description, and then basically you can provide the URL from our repository right here.

>> MARC-ARTHUR PIERRE LOUIS: Can you tell us again where you are? We missed that a little bit. We went to where to get to this page?
>> SCOTT D'ANGELO: The starting point for these journeys is developer.IBM.com. The link for this is at the end of the slides. That will be available to you. We have a slash code slash journey page that lists all of our journeys. They are numerous. There is various subsets of journeys for cognitive, data analytics, Internet of thing, Blockchain, Cloud. There is upwards of 50 at this point.

You can go there, you can survey our journeys. There is several for the data science experience. You can basically follow the instructions from start to finish. They don't take more than an hour to complete, if that. This one is fairly straightforward.

If you need help with things like logging on to data science experience, there is links for that. We create a notebook using the URL provided in our repository. You can copy that link, drop it in the notebook URL, and go.

I've actually saved the, the trouble here, by creating this ahead of time. Once we drop that URL and create the notebook, we are ready to go. We have data science experience with PixieDust. If you are not familiar with how notebooks work, you run them either, you can run all the cells or run one cell at a
time. Usually at the beginning there is things like importing, things with Pip, Python based notebook, importing other dependencies. We will walk you through those instructions.

I've already done that here. It doesn't take that long, but to save ourselves time I've walked through the basic things of installing the packages.

We import data from San Francisco's open data. They provide us with a ton of things, transportation, public safety, housing and buildings. There is a tremendous free and available data set here for the City of San Francisco. We use that for our example.

With one line of PixieDust we pull a net sample data. We have done a workshop here. The way the notebooks work, you can have some text and you have your cells that run the code. We asked you some questions to help you walk through this, to analyze things.

We have got our data. We have got a question on what police district do most traffic accidents occur? We have built a pie chart. We can build multiple charts. As we build our charts we have keys and values. What I've done is the police district and the incident number, the aggregate and count up the incidents, and we basically see that we have 13 percent of the accidents in bay view, 13 percent Tara Val, 13 percent southern district. It's all very interesting.

Next we can think about another question, how is the accident resolved? We will go back to our keys or actually, we will go back to our clustering option.

My options, to look at resolutions, I don't see it here. We will just go on to this other one, which is a bar chart. We are going to look at what days of the week do the most traffic accidents occur.

It loads up the data. All this is interactive. You have options for various types of charts. You have options we are displaying different things on the X and Y axis, so now we have a chart of resolution, whether it's an arrest, juvenile, we can look at the various districts and see how things are sorted. I like this stacked view. You can change it to grouped views, that separate these out into different charts.

>> MARC-ARTHUR PIERRE LOUIS: Scott, you said it was the day of the week when the accident occurred. Can you scroll to the bottom so we can see that?

>> SCOTT D'ANGELO: Yeah. But first I need to -- good proofreading there. You can change these quickly. It reloads the data. We don't have to change any code. It is all built into the display.

You can see things like days of the week, total seems to ramp
up Thursday, Friday, Saturday, Sunday -- Towards the weekends, traffic accidents. You have a ton of tools to visualize the data in a variety of ways. We move on and we basically do a couple of different things. We want to look at the southern Tara Val police districts, look at a couple different days. We have cleansed the data so that our dates are all in numerical order. We have collated things so we don't have so many different fields. We basically say does it result in the arrest or not. We do data exploration here.

We wonder about police districts, are there more arrests in one district than another. We have arrests in red, no resolution in green. We do see a bit of a peak in the mission district for arrests. We can form our hypothesis based on that, whether or not there is more crime that result in arrest there, whether the police are more aggressive there. You can do as you will with that data.

Next is we want to look at how do the number of accidents change over the course of the week. We can go ahead and create a line chart, looking at days of the week and incident number. Police district, we will change quickly to days of the week, simple as that. We will redisplay it. Now we have got our days of the week on the X axis and the number of incidents on the Y. You can see that there is a peak at Wednesday, in accidents. But the accidents that result in arrest are fairly low until the weekend. Then they tend to go upwards.

I guess that is to be expected, assuming people on the weekend are behaving more badly, you are going to see an increase in arrests, but not necessarily a increase in accidents. There is still a large amount of accidents in the middle of the week.

We have a lot of tools here to play with various fields. One thing we will do next is we will look at the Taraval police district. This is easy to use, sequel notation, so selecting the accidents where the district is Taraval. We want to use a map. Instead of a previous rendering tool, we will use mat box. We support the Google maps API also. We will use mat box to see where in the Tara Val district the accidents happen. We get a nice map with the aggregation of the accidents, clustered around.

If you are familiar with San Francisco, you can zoom in at the street level and see where the accidents are occurring. You can get by hovering over, you can see hidden fields. We have a question, what time of day do most accidents occur? We will look at a line chart. We can quickly switch from map to line. We can look at the hour. Oops, I'm going to have to change that.

Get rid of these guys, and just reloads. We can see certain
things about the hour of the day, obviously a peak here around -- (overlapping speakers) 1300, people are on their lunch hour perhaps, what's happening here, it looks like it's around 6, 7:00 at night, maybe people are at home eating dinner and then things ramp up again. Probing into data you can see how traffic is affected, how accidents are affected, times of day, hour.

We can look deeper, we want to get more data in, and that is where pixie apps come in. What pixie apps allows you to do is pull in new data sets and put it into another interactive map. There is some information here about how we create the map layers. We use open data's TO JSON information to add new layers to the map. We have got documentation for pixie app linked into our code. PixieDust and pixie apps, I'll give you tremendous tools for building this and the information will all be here.

We have added various layers for speeding, traffic calming, police district and crimes. You can see that the code is simple. You give it a name. You grab the URL from the open data platform and it pulls all these in. The setup is easy. We get this map JSON options and we do that simply from the metadata. It is available in our URL or excuse me, in our notebook.

Then we do HTML formatting here, and that creates an interactive map. We take layers so we can look at a hypothesis, does traffic calming affect the number of accidents or does speeding affect the number of accidents. You can drill in on a street by street basis and get the data as you hover over.

Adding and removing layers is quick and easy. Add, remove, redraws the map. You can see there is lots of speeding happening, where there are not accidents. There is accidents where there is not a lot of speeding. Maybe that hypothesis is false. Maybe speeding has nothing to do with the number of accidents. Maybe you look at traffic calming. Get rid of the speeding. See if the presence of police is either a cause or effect of accidents.

Once again, you can form your hypothesis, if the police came first or the accidents came first, it's kind of hard to tell. These tools basically allow you to pull in various layers of your information from any data source, and then you can play with the layers. You can choose different ways of formatting, whether it's line chart, pie chart, graph.

All of the documentation is available in PixieDust. Follow this journey, at your leisure. It is quick and easy. You can play around with it. You can see what the tools allow.

>> MARC-ARTHUR PIERRE LOUIS: Quick question for you, Scott.
>> SCOTT D'ANGELO: You bet.
MARC-ARTHUR PIERRE LOUIS: The data that you get from the City of San Francisco, in what format is it? Is it through a database like SQL database or what kind of format you get that?

SCOTT D'ANGELO: You can export this data. Let me see if I can find a good one. You can export the data in almost every format you want, comma separated values or JSON. But basically it dumps out a complete data set. As you choose upload, you get CSV, JSON, RSS, XML, you name it.

MARC-ARTHUR PIERRE LOUIS: That's really great.

SCOTT D'ANGELO: It drops it right in. This is a neat data set. But the tools, they will take anything, as long as you can find a way to export it, you can put it into your notebook.

MARC-ARTHUR PIERRE LOUIS: Excellent.

SCOTT D'ANGELO: David is going to show us how to embed these pixie app tools into a way that you can view it in a Web Page.

MARC-ARTHUR PIERRE LOUIS: David, you have it.

DAVID TAIEB: Yes, I'm going to reshare my screen. Thank you, Scott. We have completed the demo. Hopefully you have a better sense of how Jupyter, how PixieDust can help make the most of Jupyter so that data scientists don't spend most of their time writing charts with Googling continuously to find code sample.

Then you have seen about pixie apps which allow you to build quickly, to turn those analytics that you have built into the notebook in a linear manner, to turn them and operationalize them into something that is more accessible to say line of business users, or executives who wants to look at what, at the results of your analytics.

One of the biggest problems also that I mentioned before was that the lack of communication and data scientists have a hard time finding a way to communicate the results of their analytics, and PixieDust is a nice way for them to turn their very cool data analytics into dashboards and results that can be consumed.

So now I'm going to introduce a new thing, because a lot of people who we have talked to have said pixie app is cool, charts are cool, but really I want to free them from there.

That is where pixie gateway comes in. I want to set expectation, this is a brand-new component, we just released it with version 1.1 of PixieDust. That came out last week. It is still a mode, if you decide to try it, please be careful. Do not put that in production just yet. We are going as fast as we can to improve it. There is a lot of work left. But you are welcome to try it out and give us your feedback. This is an invitation from me to you guys, to give us feedback how we can make this cool feature more robust.
It is unique, in the Jupyter world. But it is not unique in
the, there is other applications like plot line dash who do the
same thing. The only difference is this is tightly integrated
with Jupyter notebook.

Let's talk about the architecture. The pixie gateway is
built on top of Jupyter kernel gateway. It is a tornado based
server that you can deploy very quickly. I published a blog
last week, I put the URL down there, that shows you how to
deploy it locally, very simply, or on the IBM Cloud using a
Kubernetes container. We have provided you with a Docker image.
There is five or six CLI commands to do. And you get a pixie
gateway running on IBM Cloud.

The demo I'm going to show you is actually a pixie gateway
running on the IBM Cloud. How does it work? From the Jupyter
notebook, we have added, instrumented, added the actual output
of the PixieDust results with buttons that allow you to publish
and share those pixie app into the pixie gateway.

We are sending the notebook, the pixie gateway in turn is
storing those notebook, processing them. There is some static
code analysis to ensure that they can run correctly into the
kernel, and manages the actual one or multiple kernels, and then
run those notebooks into the kernels.

So when a browser client comes in and tries to access those
apps, they can do it like a regular Web app, and then the pixie
gateway will manage the session for them, running the actual
pixie app or chart into a kernel and going back to you.

The advantage is that your users do not even know that they
are running notebooks in the back end, and for you, for the
developer there is a complete leverage of all the investment you
have made in the notebook right away.

So let's do a demo, without further ado. From here, I'm in
data science experience like Scott was. I have my notebook that
I've loaded with a slightly newer version than him. You
probably remember this. What you see here is a new button for
the charts called share.

That buttons allow you to turn any of those charts into a Web
Page. If I click on share, you have now a new dialogue where
you are able to provide the server or host name, IP address or
host name with the ports. You are able to provide a
description. This is a cool chart for the San Francisco
analysis. Then you click share. That's it.

If it's successful, you are given a URL. This URL is
accessible by anyone, like you would share on box or like you
would share on Slack. You can do that. When I click on it, I
am now able to see my charts within a Web Page. I can give you
this URL. You can see it yourself.

I can turn into a line chart or pie chart. I got a lot of
line charts. I can do the same thing very quickly, share it, get a new URL and you have it.

That is how you are able to share the results of your analysis to the broader community.

>> MARC-ARTHUR PIERRE LOUIS: David, you know that you are going to get that question, that I'm going to ask you. So you are able to share a chart that you built. How about giving the user the ability to build its own chart on that data?

>> DAVID TAIEB: Yes. So you probably know that PixieDust has a very cool extension mechanism. You can build very quickly, if you go to the PixieDust documentation, you can build your own visualization, and inject it into the menus here.

Once you do that, everything else works. If you want to know more about how that works, you go to the PixieDust repo on GitHub. There is a link here, link to the documentation. In the develop section, you have write a new PixieDust visualization. The good thing about it is that there is a Wizard that generate the code for you, a boilerplate code so that you can simply generate a skeleton project and then stick in your own visualization in there, import the file. Everything is documented there.

>> MARC-ARTHUR PIERRE LOUIS: Excellent. Thanks.

>> DAVID TAIEB: Great. It works with bouquet, which is less interactive, it is more like an image. I can do the same thing. Share it. Then you get an image. What is cool about this is that now you can with a simple image tag, you can put that into your website. You can make, you can see the blog post or whatever, you can make it a reference to that chart.

Then what we are going to work on next, in the next few weeks, is the ability to provide a schedule refreshing of this chart. In other words, if the chart that you built is using data that moves, you will be able upon publishing to say, I want to update that chart every night. PixieDust will automatically rerun your stack to rebuild the charts.

If you have a blog post, that points at the charts, guess what? It is going to update every day automatically. You don't have to do anything.

That is pretty cool. You got the chart sharing. We also have the ability to publish your app with the logic in it. Here I have the San Francisco pixie app, the dashboard, that shows me, hopefully it's working -- if not, I have a local version as well. Yes, there it is. I have this app, it's cool, it's interactive. I can add layers. There is another button right there, that allow you to publish your pixie app. If you click on it the same way, you are going to get a bunch of options. First the pixie app, pixie gateway server, the page title, analysis, a page icons, whether you want this pixie app to run
in a dedicated kernel, because by default we run multiple pixie app, we can based on user load run multiple pixie app into the same kernel.

PixieDust will automatically do a scan of the code and will detect all the dependencies that you have. If you imported whatever package, it will show up here. Upon publishing, it will query the pixie gateway server to decide whether a particular package needs to be automatically deployed and installed, so that your app can work correctly.

We will make that manually as well, so that you can manually inject your own import and dependency upon publishing. We also analyze the kernel spec. If you have specific conditions, custom things that you would like to do to run your app, we will take advantage of that as well.

Then you click publish and in the same way it's going to connect. It is going to decide what to do. Is packaging the whole notebook, sending it to the pixie gateway, decide whether it needs to be restarted, the kernel needs to be restarted because it already, it's a new version of a particular app. And you see here, I was running the app before, testing so it's restarting the kernel to take advantage of the new version.

You get a URL. Eventually, right now and that is why I'm saying it, there is no security so don't use it in production. But eventually we are going to have a plug-in mechanism for you to plug in your own 'then the indication and authorization mechanism but then you have the URL that you can use as a Web app -- authentication.

Now I have the exact same thing with no code change, running as a Web app. It's fully functional. I can see my layers and everything.

It's pretty cool, because you can also embed those into your own website. You do not have to send them to the pixie gateway server. You can have an iFrame for instance into your own website. Then you basically made the whole steps from basic data science in the notebook to development of the pixie app to deployment a very streamlined process.

If you want the know how the app was deployed, there is a index of all the apps that are deployed. I deployed three of them. One is a stock viewer that is a cool app that will allow you to know the stock ticker. This is all built by PixieDust. I can have multiple ones. Then I can look at them as line charts. I think IBM is having a good day in the stock market today. So you can see the actual things. But all of this is really built with simple PixieDust display, no JavaScript, no real heavy coding that you would have to do in there. Just an example. We built a weather app as well. My colleague Margaret has a blog post on that.
You can see here we have a very cool view, if it's working. We may have a little problem. You have a dashboard where you can click. There is a little problem with the app, it's the same data but in the notebook it works better. We are still debugging. There is a styling issue that we need to work on. But mostly it's functional.

You can drill down between the two and so on and so forth. If you want to know more about all this, you can go to a blog post that I've written to you on the video. I explained the whole thing, how to deploy it. And I even give you the stock viewer application as a sample that you can directly deploy yourself. That is pretty exciting. You can start looking at that, and give us feedback, please. We crave for this feedback.

I'm going to go back to the presentation. That's the demo.

If you want to learn more about the platform, you go to the URL here. We have a lot of articles and blog posts from our advocacy team. If you want to know more about PixieDust itself we have a Web Page that has a lot of latest information. The data catalog is important. It is going to be GA, it is going to be open beta I think starting next month. If you want to know more about the journeys, specifically the code journey that Scott has shown you, you can go to this URL, and it's all kinds of journeys as well. So thank you very much.

If there are any questions, we have ten minutes. I'll be happy, Scott and I will be happy to stay a little longer to answer those questions.

>> MARC-ARTHUR PIERRE LOUIS: So Steve, who is manning the Q&A booth, Steve, are there any particular questions that you feel could be answered live? You are welcome to bring it up now.

>> Sure. I guess one of the first questions that I saw was how, for Scott, how long would it have taken if you didn't have the PixieDust library available? To do a simple --

>> SCOTT D'ANGELO: How long to build those charts?

>> Yeah. How complicated is it?

>> SCOTT D'ANGELO: A lot of times like a mat lab chart it's 30 lines of code instead of one. It's lines of code that I know I personally would have to do research to make sure I got right. It is not something that, you need a lot of fluency in the tools to come up with it from memory. So I think the complexity to create a chart if you are not familiar with doing it might take you 20 minutes to do something you can do in one line in a couple seconds using PixieDust.

(clicking noise).

>> Thanks for the answer, Scott. It's something that saves you a good chunk of time and simplifies things. (overlapping speakers).

>> DAVID TAIEB: More importantly, there is definitely the
time it takes to build those which is really painful, but more importantly is the interactivity that matters, because when you explore data, it's exploratory. You try to go fast and try different directions. If you have to code it, you will necessarily leave options on the table. The fact that you can easily go and switch lets you explore much more and much more alternatives than you would if you had to code it yourself. That is more important as well.

> Along those lines can you use PixieDust, some of the questions I'm reading, for something aside from mapping, you know, mapping data on like an actual physical map. Can you use it for other things?

> DAVID TAIEB: Yes, absolutely. You have seen the visualization out of the box. We have had, there is a blog post that shows you how to build your own visualization.

We have an example of a visualization that visualize graphs, using Spark graphics. We did a custom visualization that explores plane trips between two airports, and show a map with all the lines showing you the actual airlines' routes. So that is a good example.

We have people building bollinger charts and so on, so from the chart perspective PixieDust extensibility allow you to build your own charts. The pixie app on the other hand is, think of it as an app development framework that allows you to compose those visualizations and build them together, link them if you want to drill down to build a real app with the logic.

Between those two, I think you are covering a lot of cases. If you don't see what you like, you can go and build it yourself.

> So yeah, thanks. I'm going to keep rolling with more questions, if you don't mind, for David and Scott. I'm reading through all the other questions here.

One of the ones that I think there's some folks are, it would help if you identify, our bosses will appreciate this, but can you identify competing tools that offer similar capabilities? I think a lot of folks are getting various technologies kind of mismatched. So are there any similar tools out there?

(overlapping speakers) can we close this window or go back or do something related?

> DAVID TAIEB: Absolutely. Notebooks are all the rage right now. Jupyter is not the only one in town. There is zeppelin, another type of notebooks, and there is also data breaks, data science addition, committee edition which gives you Spark related notebooks. Those two notebooks zeppelin and data breaks provide you with also a simple API for visualization. It is not new. I agree.

What is new is that Jupyter doesn't have something imminent,
so when we built PixieDust we wanted to [inaudible] (pings).

Jupyter has a lot of advantage in other area, but it was lacking into the auto visualization. We built PixieDust to fill that gap at the beginning.

Where we went beyond is when we started to talk about pixie apps, make it extensible so you can build your own visualizations, and then publishing on the Web. For the publishing on the Web, I mentioned before, R sheening and plot dash do similar things. They allow you to use data science homeworks like Python and R to build data rich application very similarly. The only advantage, the only difference I would say or one major difference between R, shining and pash is that PixieDust is tightly integrated with Jupyter notebooks. So that you can, you have a continuity between the algorithm and the analytics that you build in the notebooks, and the development of your apps.

>> Okay, one more question. Maybe Scott, you can chime in on this one too. What skills are necessary for data science, and what really separates the every day data science to a data science expert?

>> DAVID TAIEB: That is a $64,000 question. (chuckles).
My answer would be, you recognize data scientist when you see it. I don't claim to be data scientist, I don't think anyone here claims to be data scientist. We all have college degrees in statistics. Data science is a field that mixes statistics with machine learning, and so on. It's a very gray area. I call that an art really, because it needs a lot of experience, you need instinct. We all can dabble in data science. I think it's hard to be a real expert in data science itself. Scott, you want to add something about that?

>> SCOTT D'ANGELO: Yeah, I think back when I started my IT career, that they gave me the assignment to look at something for a day, they came back the next day and said Scott, you are the expert. I said I'm the expert? I only looked at it for one day. They said you know more than everybody else. It's the way it is. It's a fairly new, and there is just not a lot of history to say data scientist follow this path, and we can look and see good data science and good data scientists and recognize them. But I think it's something you keep building on. It is moving fast and you gain your expertise every time you find new information.

>> DAVID TAIEB: Exactly. I compare that to be like a police detective. What is a good police detective? I think that working with data is like working with a perp. You have to torture them until they talk. You have to torture the data until they talk. How do you do it? Experience, training,
practice, and tools. Tools is really important.
I can't even start to fathom how hard it would be without the tools. And making the tools simple, even for data scientist, is extremely important. That is our mission.

>> I guess to put it one way, it's not really so much about doing, it's not doing data science, not being a data scientist --

>> DAVID TAIEB: We are trying to bring the tool of data science to the developers. We are not trying to make developers data scientists. But we are trying to (overlapping speakers) give them the tools so that they can be independent and they can collaborate and communicate with the data scientist world.

>> Awesome. I'm going to give it back to promote the next tech talk.

>> MARC-ARTHUR PIERRE LOUIS: Thanks for a great tech talk, Scott and David. Those charts and the video is going to be available. You can replay them.

Next we have one also a good one for you, about using swift to create sample photo sharing app. That is the most downloaded app that we have on our site. You want to come next week, bright and early, to see how you can use Swift to create that sample photo sharing application and our presenter is going to be Julio, sorry, Ricardo Olivieri who wrote the application and will take you through it how he did it. We thank you for your time and until the next tech talks we ask that you have a great week. Talk to you next week. Thank you.

>> Good-bye.

(end of session at 11:00 a.m. CST)

***
This text is being provided in a rough draft format. Communication Access Realtime Translation (CART) is provided in order to facilitate communication accessibility and may not be a totally verbatim record of the proceedings.

***