**Tools for Data Science-course-2**

by IBM

Timeline

**Instructor's Note**





Welcome to Open Source tools for Data Science! You’re joining thousands of learners currently enrolled in the course. We're excited to have you in the class and look forward to your contributions to the learning community.

To begin, we recommend taking a few minutes to explore the course site. Review the material we’ll cover each week, and preview the assignments you’ll need to complete to pass the course. Click **Discussions** to see forums where you can discuss the course material with fellow students taking the class.

If you have questions about course content, please post them in the forums to get help from others in the course community. For technical problems with the Coursera platform, visit the [**Learner Help Center**](http://learner.coursera.help/).

Good luck as you get started, and we hope you enjoy the course!

**Less**

#### Course Introduction

Welcome to the course!

You've begun one of the most complete overviews on data science tooling that you’ll currently

find on the internet.

This doesn’t mean that we cover each and every tool, but later in the course we’ll

introduce a comprehensive list of tasks a data scientist needs to perform and give you

the top two or three open source and commercial tools available to complete them.

We also explain how the tools overlap in functionality, what their pros and cons are, and how these

tools can address the whole data science pipeline.

Let’s start with data.

Data is obviously central to data scientists.

In this course, we’ll show you how to manage, extract, transform, analyze, and visualize

data.

Now, you might be able to survive data science without programming skills if you use the

right set of tools.

However, we highly recommend getting familiar with programming and the related programming

frameworks for data science.

To help you along, we’ll introduce you to the most commonly used programming languages

and frameworks available for data science.

That said, there is a lot of automation available in the latest tooling that a data scientist

can use.

In this course, we’ll explain how to make use of those tools to save time and uncover

inspiration.

Visual programming is available in many tools.

In this course, you’ll learn how visual programming can be used to speed up development

time and to help non-programmers enter the field of data science.

Open source software is leading the field of data science, but its total costs of ownership,

or "TCO," can be higher at times due to configuration, customization and maintenance costs.

As a result, commercial software also has its place, especially since the new generation

of commercial data science software leverages open source software and open standards.

This makes it easy to migrate between tools and can reduce overall TCO.

In this course, we’ll introduce you to both open source and commercial software and point

out their strengths and weaknesses for data science.

We'll also show you ways that you can take advantage of their respective strengths.

Finally, we'll show you how cloud computing can be used to further speed up and facilitate

data scientists' work.

We'll introduce you to the most commonly used and newly emerging cloud tools for data science.

In addition to lectures, this course, has numerous labs to make you more familiar with

the material and get hands-on experience.

There are also multiple quizzes to test your learning.

Nothing more to say than we’re glad to have you in the course and happy learning.

In case you have trouble in any way, please don’t hesitate to contact us in the discussion

forum.

There's nothing left but to begin!

We're very happy to have you with us as you start your data science journey.

If you have any trouble with any of the course material, please don’t hesitate to contact

us in the discussion forum.

Let's get started!

#### Languages of Data Science

The languages of Data Science

For anyone just getting started on their data science journey, the range of technical options

can be overwhelming. There is a dizzying amount of choice when it comes to programming languages.

Each has it's own strengths and weaknesses and there is no one right answer to the question

of which one you should learn first. The answer to that question depends largely on your needs,

the problems you are trying to solve, and who you are solving them for.

Python, R, and SQL are the languages that we recommend you consider first and foremost.

But there are so many others that have their own strengths and features.

Scala, Java, C++, and Julia are some of the most popular.

Javascript, PHP, Go, Ruby, and Visual Basic all have their own unique use cases as well.

The language you choose to learn will depend on the things you need to accomplish and the

problems you need to solve. It will also depend on what company you work for, what role you

have, and the age of your existing application. We’ll explore the answers to this question

as we dive into the popular languages in the data science industry.

There are many roles available for people who are interested in getting involved in

data science. Business Analyst

Database Engineer Data Analyst

Data Engineer Data Scientist

Research Scientist Software Engineer

Statistician Product Manager

Project Manager and many more.

Let’s dive into what we will learn in Lesson 1. We will put most of our focus on the top

three Data Science languages: Python, R, and SQL, which each have their own lessons. Then

we will go on to highlight other noteworthy languages and what makes them special. Then

we’ll finish with a short quiz to test your knowledge!

● 1.1.1 - Python ● 1.1.2 - R

● 1.1.3 - SQL ● 1.1.4 - Other Noteworthy Data Science

Languages ● 1.1.5 - Practice Quiz

#### Introduction to Python

In this video, we will review the high-level features of the Python programming language.

Python is a powerhouse language.

It is by far the most popular programming language for data science.

According to the 2019 Kaggle Data Science and Machine Learning Survey, 75% of the over

10,000 respondents from around the world reported that they use Python on a regular basis.

Glassdoor reported that in 2019 more than 75% of data science positions listed included

Python in their job descriptions.

When asked which language an aspiring data scientist should learn first, most data scientists

say Python.

You are probably thinking, why on earth is Python so popular?

Well, let’s start with the people who use Python.

If you already know how to program, then Python is great for you because it uses clear, readable

syntax.

You can do many of the things you are used to doing in other programming languages but

with Python you can do it with less code.

If you want to learn to program, it’s also a great starter language because of the huge

global community and wealth of documentation.

In fact, several different surveys in 2019 found that over 80% of data professionals

worldwide use Python.

Python is useful for many situations, including data science, AI and machine learning, web

development, and IoT devices like the Raspberry Pi.

Large organizations that use Python heavily include IBM, Wikipedia, Google, Yahoo!, CERN,

NASA, Facebook, Amazon, Instagram, Spotify, and Reddit.

Python is a powerful general-purpose programming language that can do a lot of things.

It is widely supported by a global community and shepherded by the Python Software Foundation.

1.

Python is a high-level general-purpose programming language that can be applied to many different

classes of problems.

2.

It has a large, standard library that provides tools suited to many different tasks, including

but not limited to databases, automation, web scraping, text processing, image processing,

machine learning, and data analytics.

3.

For data science, you can use Python's scientific computing libraries such as Pandas, NumPy,

SciPy, and Matplotlib.

4.

For artificial intelligence, it has TensorFlow, PyTorch, Keras, and Scikit-learn.

5.

Python can also be used for Natural Language Processing (NLP) using the Natural Language

Toolkit (NLTK).

Another great selling point is the Python community, which has a well documented history

of paving the way for diversity and inclusion efforts in the tech industry as a whole.

The Python language has a code of conduct executed by the Python Software Foundation

that seeks to ensure safety and inclusion for all, in both online and in person python

communities.

There are also communities like PyLadies that seek to create spaces for people interested

in Python to learn in safe and inclusive environments.

PyLadies is an international mentorship group with a focus on helping more women become

active participants and leaders in the Python open source community.

#### Introduction to R Language

In this video, we will give a brief overview of the R programming language.

After our last video on Python, where we discussed its wide adoption, you might be wondering

why on earth you should consider learning any other language.

Well, according to the results of the 2019 Kaggle Data Science survey, which had over

10k respondents from around the world, learning up to three languages can increase your salary!

And R has a lot to offer you.

Like Python, R is free to use, but it's a GNU project -- instead of being open source,

it's actually free software.

So if Python is open source and R is free software, what’s the difference?

Well, Both open source and free software commonly refer to the same set of licenses.

Many open source projects use the GNU General Public License, for example.

Both open source and free software support collaboration.

In many cases (but not all), these terms can be used interchangeably.

The Open Source Initiative (OSI) champions open source while the Free Software Foundation

(FSF) defines free software.

Open source is more business focused, while free software is more focused on a set of

values.

Back to why you should learn R. Because this is a free software project, you can use the

language in the same way that you contribute to open source, and it allows for public collaboration

and private and commercial use.

Plus, R is another language supported by a wide global community of people passionate

about making it possible to use the language to solve big problems.

Who is R for?

It's most often used by statisticians, mathematicians, and data miners for developing statistical

software, graphing, and data analysis.

The language’s array-oriented syntax makes it easier to translate from math to code,

especially for someone with no or minimal programming background.

According to Kaggle’s Data Science and Machine Learning Survey, most folks learn R when they're

a few years into their data science career, but it remains a welcoming language to those

who don’t have a software programming background.

R is popular in academia but companies that use R include IBM, Google, Facebook, Microsoft,

Bank of America, Ford, TechCrunch, Uber, and Trulia.

● R has become the world’s largest repository of statistical knowledge.

● As of 2018, R has more than 15,000 publicly released packages, making it possible to conduct

complex exploratory data analysis.

● R integrates well with other computer languages, such as C++, Java, C, .Net, and

Python.

● Common mathematical operations such as matrix multiplication work straight out of

the box.

● R has stronger object-oriented programming facilities than most statistical computing

languages.

There are many ways to connect with other R users around the globe.

Communities such as user!, WhyR?, SatRdays, and R-Ladies are all great to connect with.

And you can also check out the R project website for R conferences and events.

#### Introduction to SQL

In this video, we'll take a high-level look at SQL.

SQL is a bit different from the other languages we’ve covered so far.

First off, it's formally pronounced “ess cue el,” although some people say “sequel.”

While the acronym stands for “Structured Query Language,” many people do not consider

SQL to be like other software development languages because it's a non-procedural language

and its scope is limited to querying and managing data.

While it is not a “data science” language per se, data scientists regularly use it because

it's simple and powerful!

Another couple of neat facts about SQL: it's much older than Python and R, by about 20

years, having first appeared in 1974.

And, SQL was developed at IBM!

This language is useful in handling structured data; that is, the data incorporating relations

among entities and variables.

SQL was designed for managing data in relational databases.

Here you can see a diagram showing the general structure of a relational database.

A relational database is formed by collections of two-dimensional tables; for example, datasets

and Microsoft Excel spreadsheets.

Each of these tables is then formed by a fixed number of columns and any number of rows.

BUT!

Even though SQL was originally developed for use with relational databases, because it's

so pervasive and easy to use, SQL interfaces for many NoSQL and big data repositories have

also been developed.

The SQL language is subdivided into several language elements, including clauses, expressions,

predicates, queries, and statements.

So what makes SQL great?

Knowing SQL will help you do many different jobs in data science, including business and

data analyst, and it's a must in data engineering.

When performing operations with SQL, you access the data directly.

There's no need to copy it beforehand.

This can speed up workflow executions considerably.

SQL is the interpreter between you and the database.

SQL is an American National Standards Institute, or "ANSI," standard, which means if you learn

SQL and use it with one database, you will be able to easily apply that SQL knowledge

to many other databases.

There are many different SQL databases available, including MySQL, IBM Db2, PostgreSQL, Apache

OpenOffice Base, SQLite, Oracle, MariaDB, Microsoft SQL Server, and more.

The syntax of the SQL you write might change a little bit based on the relational database

management system you’re using.

If you are looking to learn SQL you would be best served to focus on a specific relational

database and then plug into the community for that specific platform.

There are also many great introductory courses on SQL available!

#### Other Languages

So far, we’ve reviewed Python, R, and SQL.

In this video, we will review some other languages that have compelling use cases for data science.

Ok, so indisputably, Python, R, and SQL are the three most popular languages that data

scientists use.

But, there are many, many other languages that are worth your time when considering

which language to use to solve a particular data science problem.

Scala, Java, C++, and Julia are probably the most traditional data science languages on

this slide.

But JavaScript, PHP, Go, Ruby, Visual Basic, and others have all found their place in the

data science community as well!

I won’t dive as deeply into each of these languages, but I'll mention some notable highlights.

Java is a tried-and-true general-purpose object oriented programming language.

It's been widely adopted in the enterprise space and is designed to be fast and scalable.

Java applications are compiled to bytecode and run on the Java Virtual Machine, or "JVM."

Some notable data science tools built with Java include Weka, for data mining; Java-ML,

which is a machine learning library; Apache MLlib, which makes machine learning scalable;

and Deeplearning4j, for deep learning.

Apache Hadoop is another Java-built application.

It manages data processing and storage for big data applications running in clustered

systems.

Scala is a general-purpose programming language that provides support for functional programming

and a strong static type system.

Many of the design decisions in the construction of the Scala language were made to address

criticisms of Java.

Scala is also interoperable with Java, as it runs on the JVM.

The name "Scala" is a combination of "scalable" and "language."

This language is designed to grow along with the demands of its users.

For data science, the most popular program built using Scala is Apache Spark.

Spark is a fast and general-purpose cluster computing system.

It provides APIs that make parallel jobs easy to write, and an optimized engine that supports

general computation graphs.

Spark includes Shark, which is a query engine; MLlib, for machine learning; GraphX, for graph

processing; and Spark Streaming.

Apache Spark was designed to be faster than Hadoop.

C++ is a general-purpose programming language.

It is an extension of the C programming language, or "C with Classes.”

C++ improves processing speed, enables system programming, and provides broader control

over the software application.

Many organizations that use Python or other high-level languages for data analysis and

exploratory tasks still rely on C++ to develop programs that feed that data to customers

in real-time.

For data science, a popular deep learning library for dataflow called TensorFlow was

built with C++.

But while C++ is the foundation of TensorFlow, it runs on a Python interface, so you don’t

need to know C++ to use it.

MongoDB, a NoSQL database for big data management, was built with C++.

Caffe is a deep learning algorithm repository built with C++, with Python and MATLAB bindings.

A core technology for the World Wide Web, JavaScript is a general-purpose language that

extended beyond the browser with the creation of Node.js and other server-side approaches.

Javascript is NOT related to the Java language.

For data science, the most popular implementation is undoubtedly TensorFlow.js.

TensorFlow.js makes machine learning and deep learning possible in Node.js as well as in

the browser.

TensorFlow.js was also adopted by other open source libraries, including brain.js and machinelearn.js.

The R-js project is another great implementation of JavaScript for data science.

R-js has re-written linear algebra specifications from the R Language into Typescript.

This re-write will provide a foundation for other projects to implement more powerful

math base frameworks like Numpy and SciPy of Python.

Typescript is a superset of JavaScript.

Julia was designed at MIT for high-performance numerical analysis and computational science.

It provides speedy development like Python or R, while producing programs that run as

fast as C or Fortran programs.

Julia is compiled, which means that the code is executed directly on the processor as executable

code; it calls C, Go, Java, MATLAB, R, Fortran, and Python libraries; and has refined parallelism.

The Julia language is relatively new, having been written in 2012, but it has a lot of

promise for future impact on the data science industry.

JuliaDB is a particularly useful application of Julia for data science.

It's a package for working with large persistent data sets.

That's as far as we’ll dig into the many languages that are used to solve data science

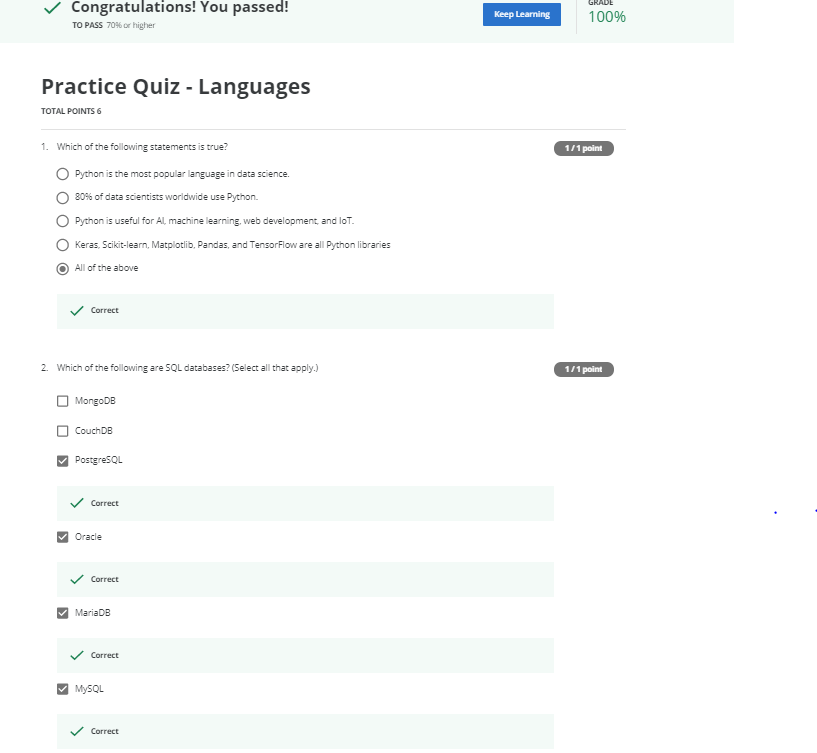
problems.

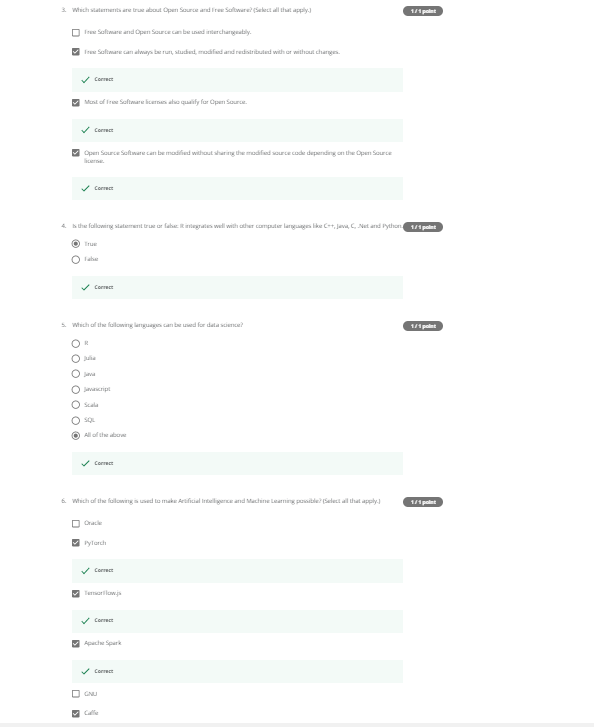
If you have experience with a particular language, I recommend you do a web search to see what

might already be possible in terms of using it for data science.

You might be surprised at the possibilities!

QUIZ





#### Categories of Data Science Tools

Open source tools are available for various data science tasks.

In this video, we’ll have a look at the different data science tasks.

In subsequent videos we’ll walk through the most commonly used open source tools for

those tasks.

The most important tools are covered throughout this course.

Data Management is the process of persisting and retrieving data.

Data Integration and Transformation, often referred to as Extract, Transform, and Load,

or “ETL,” is the process of retrieving data from remote data management systems.

Transforming data and loading it into a local data management system is also part of Data

Integration and Transformation.

Data Visualization is part of an initial data exploration process, as well as being part

of a final deliverable.

Model Building is the process of creating a machine learning or deep learning model

using an appropriate algorithm with a lot of data.

Model deployment makes such a machine learning or deep learning model available to third-party

applications.

Model monitoring and assessment ensures continuous performance quality checks on the deployed

models.

These checks are for accuracy, fairness, and adversarial robustness.

Code asset management uses versioning and other collaborative features to facilitate

teamwork.

Data asset management brings the same versioning and collaborative components to data.

Data asset management also supports replication, backup, and access right management.

Development environments, commonly known as Integrated Development Environments, or “IDEs”,

are tools that help the data scientist to implement, execute, test, and deploy their

work.

Execution environments are tools where data preprocessing, model training, and deployment

take place.

Finally, there is fully integrated, visual tooling available that covers all the previous

tooling components, either partially or completely.

This concludes this video.

In the next video we’ll start looking at open source tools for data science tasks.

#### Open Source Tools for Data Science - Part 1

In part one of this two-part series, we’ll cover data management, open source data integration,

transformation, and visualization tools.

The most widely used open source data management tools are relational databases such as

MySQL and PostgreSQL; NoSQL databases such as MongoDB Apache CouchDB, and Apache Cassandra;

and file-based tools such as the Hadoop File System or Cloud File systems like Ceph.

Finally,Elasticsearch is mainly used for storing text data and creating a search index for

fast document retrieval.

The task of data integration and transformation in the classic data warehousing world is called

ETL, which stands for “extract, transform, and load.”

These days, data scientists often propose the term “ELT” – Extract, Load, Transform“ELT”,

stressing the fact that data is dumped somewhere and the data engineer or data scientist themself

is responsible for data.

Another term for this process has now emerged: “data refinery and cleansing.”

Here are the most widely used open source data integration and transformation tools:

Apache AirFlow, originally created by AirBNB; KubeFlow, which enables you to execute data

science pipelines on top of Kubernetes; Apache Kafka, which originated from LinkedIn;

Apache Nifi, which delivers a very nice visual editor;

Apache SparkSQL (which enables you to use ANSI SQL and scales up to compute clusters

of 1000s of nodes), and NodeRED, which also provides a visual editor.

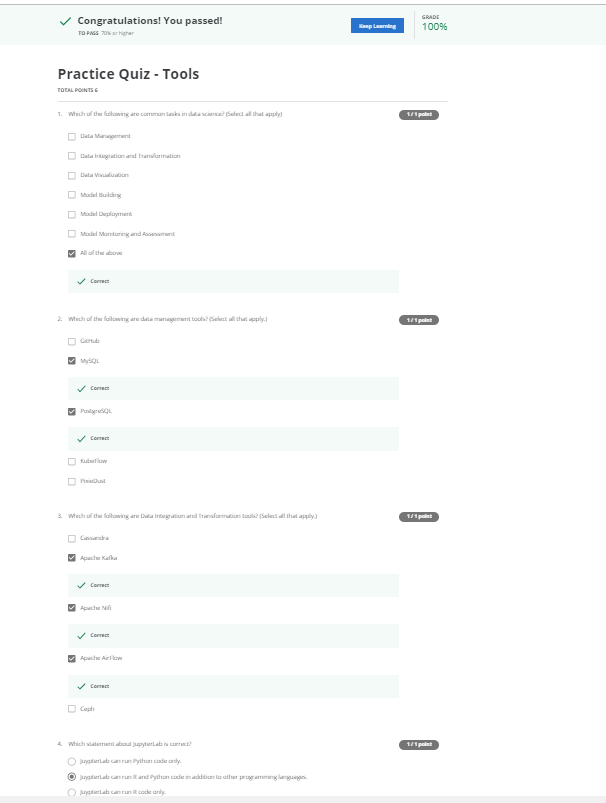
NodeRED consumes so little in resources that it even runs on small devices like a Raspberry

Pi.

We’ll now introduce the most widely used open source data visualization tools.

We have to distinguish between programming libraries where you need to use code and tools

that contain a user interface.



The most popular libraries are covered in the next videos.

A similar approach uses Hue, which can create visualizations from SQL queries.

Kibana, a data exploration and visualization web application, is limited to Elasticsearch

(the data provider).

Finally, Apache Superset is a data exploration and visualization web application.

Model deployment is extremely important.

Once you’ve created a machine learning model capable of predicting some key aspects of

the future, you should make that model consumable by other developers and turn it into an API.

Apache PredictionIO currently only supports Apache Spark ML models for deployment, but

support for all sorts of other libraries is on the roadmap.

Seldon is an interesting product since it supports nearly every framework, including

TensorFlow, Apache SparkML, R, and scikit-learn.

Seldon can run on top of Kubernetes and Redhat OpenShift.

Another way to deploy SparkML models is by using MLeap.

Finally, TensorFlow can serve any of its models using the TensorFlow service.

You can deploy to an embedded device like a Raspberry Pi or a smartphone using TensorFlow

Lite, and even deploy to a web browser using TensorFlow dot JS.

Model monitoring is another crucial step.

Once you’ve deployed a machine learning model, you need to keep track of its prediction

performance as new data arrives in order to maintain outdated models.

Following are some examples of model monitoring tools:

ModelDB is a machine model metadatabase where information about the models are stored and

can be queried.

It natively supports Apache Spark ML Pipelines and scikit-learn.

A generic, multi-purpose tool called Prometheus is also widely used for machine learning model

monitoring, although it’s not specifically made for this purpose.

Model performance is not exclusively measured through accuracy.

Model bias against protected groups like gender or race is also important.

The IBM AI Fairness 360 open source toolkit does exactly this.

It detects and mitigates against bias in machine learning models.

Machine learning models, especially neural-network-based deep learning models, can be subject to adversarial

attacks, where an attacker tries to fool the model with manipulated data or by manipulating

the model itself.

The IBM Adversarial Robustness 360 Toolbox can

be used to detect vulnerability to adversarial attacks and help make the model more robust.

Machine learning modes are often considered to be a black box that applies some mysterious

“magic.”

The IBM AI Explainability 360 Toolkit makes the

machine learning process more understandable by finding similar examples within a dataset

that can be presented to a user for manual comparison.

The IBM AI Explainability 360 Toolkit can also illustrate training for a simpler machine

learning model by explaining how different input variables affect the final decision

of the model.

Options for code asset management tools have been greatly simplified:

For code asset management – also referred to as version management or version control

– Git is now the standard.

Multiple services have emerged to support Git, with the most prominent being GitHub,

which provides hosting for software development version management.

The runner-up is definitely GitLab, which has the advantage of being a fully open source

platform that you can host and manage yourself.

Another choice is Bitbucket.

Data asset management, also known as data governance or data lineage, is another crucial

part of enterprise grade data science.

Data has to be versioned and annotated with metadata.

Apache Atlas is a tool that supports this task.

Another interesting project, ODPi Egeria, is managed through the Linux Foundation and

is an open ecosystem.

It offers a set of open APIs, types, and interchange protocols that metadata repositories use to

share and exchange data.

Finally, Kylo is an open source data lake management software platform that provides

extensive support for a wide range of data asset management tasks.

This concludes part one of this two-part series.

Now let’s move on to part two.

#### Open Source Tools for Data Science - Part 2

Welcome to part two of this series.

In this section, we’ll cover the development environment, open source data integration,

transformation, and visualization tools.

One of the most popular current development environments that data scientists are using

is “Jupyter.”

Jupyter first emerged as a tool for interactive Python programming; it now supports more than

a hundred different programming languages through “kernels.”

Kernels shouldn’t be confused with operating system kernels.

Jupyter kernels are encapsulating the different interactive interpreters for the different

programming languages.

A key property of Jupyter Notebooks is the ability to unify documentation, code, output

from the code, shell commands, and visualizations into a single document.

JupyterLab is the next generation of Jupyter Notebooks and in the long term, will actually

replace Jupyter Notebooks.

The architectural changes being introduced in JupyterLab makes Jupyter more modern and

modular.

From a user’s perspective, the main difference introduced by JupyterLab is the ability to

open different types of files, including Jupyter Notebooks, data, and terminals.

You can then arrange these files on the canvas.

Although Apache Zeppelin has been fully reimplemented, it’s inspired by Jupyter Notebooks and provides

a similar experience.

One key differentiator is the integrated plotting capability.

In Jupyter Notebooks, you are required to use external libraries in Apache Zeppelin,

and plotting doesn’t require coding.

You can also extend these capabilities by using additional libraries.

RStudio is one of the oldest development environments for statistics and data science, having been

introduced in 2011.

It exclusively runs R and all associated R libraries.

However, Python development is possible and R is therefore tightly integrated into this

tool to provide an optimal user experience.

RStudio unifies programming, execution, debugging, remote data access, data exploration, and

visualization into a single tool.

Spyder tries to mimic the behaviour of RStudio to bring its functionality to the Python world.

Although Spyder does not have the same level of functionality as RStudio, data scientists

do consider it an alternative.

But in the Python world, Jupyter is used more frequently.

This diagram shows how Spyder integrates code, documentation, visualizations, and other components

into a single canvas.

Sometimes your data doesn’t fit into a single computer’s storage or main memory capacity.

That’s where cluster execution environments come in.

The well known cluster-computing framework Apache Spark is among the most active Apache

projects and is used across all industries, including in many Fortune 500 companies.

The key property of Apache Spark is linear scalability.

This means, if you double the number of servers in a cluster, you’ll also roughly double

its performance.

After Apache Spark began to gain market share, Apache Flink was created.

The key difference between Apache Spark and Apache Flink is that Apache Spark is a batch

data processing engine, capable of processing huge amounts of data file by file.

Apache Flink, on the other hand, is a stream processing image, with its main focus on processing

real-time data streams.

Although engine supports both data processing paradigms, Apache Spark is usually the choice

in most use cases.

One of the latest developments in the data science execution environments is called “Ray,”

which has a clear focus on large-scale deep learning model training.

Let’s look at open source tools for data scientists that are fully integrated and visual.

With these tools, no programming knowledge is necessary.

Most important tasks are supported by these tools; these tasks include data integration,

transformation, data visualization, and model building.

KNIME originated at the University of Konstanz in 2004.

As you can see, KNIME has a visual user interface with drag-and-drop capabilities.

It also has built-in visualization capabilities.

Knime can be be extended by programming in R and Python, and has connectors to Apache

Spark.

Another example of this group of tools is Orange.

It’s less flexible than KNIME, but easier to use.

In this video, you’ve learned about the most common data science tasks and which open

source tools are relevant to those tasks.

In the next video, we’ll describe some established commercial tools that you’ll encounter in

your data science experience.

Let’s move on to the next video to get more details.

#### Commercial Tools for Data Science

We previously covered open source tools for data science.

Now, let’s look at the commercial options you’ll find in many enterprise projects.

Let’s revisit our overview of different tool categories.

In data management, most of an enterprise’s relevant data is stored in an

Oracle Database, Microsoft SQL Server, or IBM Db2.

Although open source databases are gaining popularity, those three data management products

are still considered the industry-standard.

They won’t disappear in the near future.

It’s not just about functionality.

Data is at the heart of every organization, and the availability of commercial supports

plays a major role.

Commercial supports are delivered directly from software vendors, influential partners,

and support networks.

When we focus on commercial data integration tools, we’re talking about “extract, transform,

and load,” or “ETL” tools.

According to a Gartner Magic Quadrant, Informatica Powercenter and IBM InfoSphere DataStage are

the leaders, followed by products from SAP, Oracle, SAS, Talend, and Microsoft.

These tools support design and deployment of ETL data-processing pipelines through a

graphical interface.

They also provide connectors to most of the commercial and open source target information

systems.

Finally, Watson Studio Desktop includes a component called Data Refinery, which enables

the defining and execution of data integration processes in a spreadsheet style.

In the commercial environment, data visualizations are utilizing business intelligence, or “BI”,

tools.

Their main focus is to create visually attractive and easy-to-understand reports and live dashboards.

The most prominent commercial examples are: Tableau, Microsoft Power BI, and IBM Cognos

Analytics.

Another type of visualization targets data scientists rather than regular users.

A sample problem might be “How can different columns in a table relate to each other?”

This type of functionality is contained in Watson Studio Desktop.

If you want to build a machine learning model using a commercial tool, you should consider

using a data mining product.

The most prominent of these types of products are: SPSS Modeler and SAS Enterprise Miner.

In addition, A version of SPSS Modeler is also available in Watson Studio Desktop, based

on the cloud version of the tool.

We’ll talk more about cloud-based tools in the next video.

In commercial software, model deployment is tightly integrated in the model building process.

This diagram shows an example of the SPSS Collaboration and Deployment Services which

are used to deploy any type of asset created by the SPSS software tools suite.

Other vendors use the same type of process.

Commercial software can also export models in an open format.

For example, SPSS Modeler supports the exporting of models as Predictive Model Markup Language,

or PMML, which can be read by many other commercial and open software packages.

Model monitoring is a new discipline and there are currently no relevant commercial tools

available.

As a result, open source is the first choice.

The same is true for code asset management.

Open source with Git and GitHub is the effective standard.

Data asset management, often called data governance or data lineage, is a crucial part of enterprise

grade data science.

Data must be versioned and annotated using metadata.

Vendors, including Informatica Enterprise Data Governance and IBM, provide tools for

these specific tasks.

The IBM InfoSphere Information Governance Catalog covers functions like data dictionary,

which facilitates discovery of data assets.

Each data asset is assigned to a data steward -- the data owner.

The data owner is responsible for that data asset and can be contacted.

Data lineage is also covered; this enables a user to track back through the transformation

steps followed in creating the data assets.

The data lineage also includes a reference to the actual source data.

Rules and policies can be added to reflect complex regulatory and business requirements

for data privacy and retention.

Watson Studio is a fully integrated development environment for data scientists.

It’s usually consumed through the cloud, and we’ll cover more about it in a later

lesson.

There is also a desktop version available.

Watson Studio Desktop combines Jupyter Notebooks with graphical tools to maximize data scientists’

performance.

Watson Studio, together with Watson Open Scale, is a fully integrated tool covering the full

data science life cycle and all the tasks we’ve discussed previously.

We’ll talk more about both in the next lesson.

but just keep in mind that they can be deployed in a local data center on top of Kubernetes

or RedHat OpenShift.

Another example of a fully integrated commercial tool is H2O Driverless AI, which covers the

complete data science life cycle.

In this lesson, you’ve learned how most common data science tasks are supported by

commercial tools.

In the next video, we’ll discover data science tools that are available exclusively on the

cloud.

Since we’ve previously covered open source tools for data science, let’s look at the

commercial options you’ll find in many enterprise projects.

Take another look at the overview of different tool categories.

Since cloud products are a newer species, they follow the trend of having multiple tasks

integrated in tools.

This especially holds true for the tasks marked green in the diagram.

Let’s start with the fully integrated visual tools category.

Since these tools introduce a component where large scale execution of data science workflows

happens in compute clusters, we’ve changed the title here and added the word “Platform.”

These clusters are composed of multiple server machines, transparently for the user, in the

background.

Watson Studio, together with Watson OpenScale, covers the complete development life cycle

for all data science, machine learning, and AI tasks.

Another example is Microsoft Azure Machine Learning.

This is also a fully cloud-hosted offering supporting the complete development life cycle

of all data science, machine learning, and AI tasks.

And finally, another example is H2O Driverless AI, which we’ve already introduced in the

last video.

Although it is a product that you download and install, one-click deployment is available

for the common cloud service providers.

Since operations and maintenance are not done by the cloud provider, as is the case with

Watson Studio, Open Scale, and Azure Machine Learning, this delivery model should not be

confused with Platform or Software as a Service -- PaaS or SaaS.

In data management, with some exceptions, there are SaaS versions of existing open source

and commercial tools.

Remember, SaaS stands for “software as a service.”

It means that the cloud provider operates the tool for you in the cloud.

As an example, the cloud provider operates the product by backing up your data and configuration

and installing updates.

As mentioned, there is proprietary tooling, which is only available as a cloud product.

Sometimes it’s only available from a single cloud provider.

One example of such a service is Amazon Web Services DynamoDB, a NoSQL database that allows

storage and retrieval of data in a key-value or a document store format.

The most prominent document data structure is JSON (pronounced “jay-sun”).

Another flavour of such a service is Cloudant, which is a database-as-a-service offering.

But, under the hood it is based on the open source Apache CouchDB.

It has an advantage: although complex operational tasks like updating, backup, restore, and

scaling are done by the cloud provider, under the hood this offering is compatible with

CouchDB.

Therefore, the application can be migrated to another CouchDB server without changing

the application.

And IBM offers Db2 as a service as well.

This is an example of a commercial database made available as a software-as-a-service

offering in the cloud, taking operational tasks away from the user.

When it comes to commercial data integration tools, we talk not only about “extract,

transform, and load,” or “ETL” tools, but also about “extract, load, and transform,”

or “ELT,” tools.

This means the transformation steps are not done by a data integration team but are pushed

towards the domain of the data scientist or data engineer.

Two widely used commercial data integration tools are Informatica Cloud Data Integration

and IBM’s Data Refinery.

Data Refinery enables transformation of large amounts of raw data into consumable, quality

information in a spreadsheet-like user interface.

Data Refinery is part of IBM Watson Studio.

The market for cloud data visualization tools is huge, and every major cloud vendor has

one.

An example of a smaller company’s cloud-based data visualization tool is DataMeer.

IBM offers it’s famous Cognos Business intelligence suite as cloud solution as well.

IBM Data Refinery also offers data exploration and visualization functionality in Watson

Studio.

Again, these are just some examples of a rapidly changing and growing commercial ecosystem

among a huge number of established and emerging vendors.

In Watson Studio, an abundance of different visualizations can be used to better understand

data.

For example, this 3D bar chart enables you to visualize a target value on the vertical

dimension, which is dependent on two other values on the horizontal dimensions.

Coloring enables you to visualize a third dimension.

Hierarchical edge bundling enables you to visualize correlations and affiliations between

entities.

If sufficient, a classic bar chart can do the job as well, whereas a 2D scatter plot

with a heat map shows two dependent data fields, one on the y axis and one as color intensity.

A tree map shows distribution of subsets within a set, the famous pie chart does the same

but in a non-hierarchical manner, and finally, a word cloud pops out significant terms in

a document corpus.

Model building can be done using a service such as Watson Machine Learning.

Watson Machine Learning can train and build models using various open source libraries.

Google has a similar service on their cloud called AI Platform Training.

Nearly every cloud provider has a solution for this task.

Model deployment in commercial software is usually tightly integrated to the model building

process.

Here is an example of the SPSS Collaboration and Deployment Services, which can be used

to deploy any type of asset created by the SPSS software tools suite.

The same holds for other vendors.

In addition, commercial software can export models in an open format.

As an example, SPSS Modeler supports exporting models as Predictive Model Markup Language,

or “PMML,” which can be read by numerous other commercial and open software packages.

Watson Machine Learning can also be used to deploy a model and make it available to consumers

using a REST interface.

Amazon SageMaker Model Monitor is an example of a cloud tool that continuously monitors

deployed machine learning and deep learning models.

Again, every major cloud provider has similar tooling.

This is also the case for Watson OpenScale.

OpenScale and Watson Studio…

…unify the landscape.

Everything marked in green can be done using Watson Studio and Watson OpenScale.

We’ll cover Open Scale will be covered in a later video.

You’ve learned how the most common tasks in data science are supported by commercial

cloud tools.

Integration provides us the ability to use the same tools for multiple tasks.

In the next videos, we’ll look at packages, APIs, datasets, and models for data science.

#### Cloud Based Tools for Data Science

Since we’ve previously covered open source tools for data science, let’s look at the

commercial options you’ll find in many enterprise projects.

Take another look at the overview of different tool categories.

Since cloud products are a newer species, they follow the trend of having multiple tasks

integrated in tools.

This especially holds true for the tasks marked green in the diagram.

Let’s start with the fully integrated visual tools category.

Since these tools introduce a component where large scale execution of data science workflows

happens in compute clusters, we’ve changed the title here and added the word “Platform.”

These clusters are composed of multiple server machines, transparently for the user, in the

background.

Watson Studio, together with Watson OpenScale, covers the complete development life cycle

for all data science, machine learning, and AI tasks.

Another example is Microsoft Azure Machine Learning.

This is also a fully cloud-hosted offering supporting the complete development life cycle

of all data science, machine learning, and AI tasks.

And finally, another example is H2O Driverless AI, which we’ve already introduced in the

last video.

Although it is a product that you download and install, one-click deployment is available

for the common cloud service providers.

Since operations and maintenance are not done by the cloud provider, as is the case with

Watson Studio, Open Scale, and Azure Machine Learning, this delivery model should not be

confused with Platform or Software as a Service -- PaaS or SaaS.

In data management, with some exceptions, there are SaaS versions of existing open source

and commercial tools.

Remember, SaaS stands for “software as a service.”

It means that the cloud provider operates the tool for you in the cloud.

As an example, the cloud provider operates the product by backing up your data and configuration

and installing updates.

As mentioned, there is proprietary tooling, which is only available as a cloud product.

Sometimes it’s only available from a single cloud provider.

One example of such a service is Amazon Web Services DynamoDB, a NoSQL database that allows

storage and retrieval of data in a key-value or a document store format.

The most prominent document data structure is JSON (pronounced “jay-sun”).

Another flavour of such a service is Cloudant, which is a database-as-a-service offering.

But, under the hood it is based on the open source Apache CouchDB.

It has an advantage: although complex operational tasks like updating, backup, restore, and

scaling are done by the cloud provider, under the hood this offering is compatible with

CouchDB.

Therefore, the application can be migrated to another CouchDB server without changing

the application.

And IBM offers Db2 as a service as well.

This is an example of a commercial database made available as a software-as-a-service

offering in the cloud, taking operational tasks away from the user.

When it comes to commercial data integration tools, we talk not only about “extract,

transform, and load,” or “ETL” tools, but also about “extract, load, and transform,”

or “ELT,” tools.

This means the transformation steps are not done by a data integration team but are pushed

towards the domain of the data scientist or data engineer.

Two widely used commercial data integration tools are Informatica Cloud Data Integration

and IBM’s Data Refinery.

Data Refinery enables transformation of large amounts of raw data into consumable, quality

information in a spreadsheet-like user interface.

Data Refinery is part of IBM Watson Studio.

The market for cloud data visualization tools is huge, and every major cloud vendor has

one.

An example of a smaller company’s cloud-based data visualization tool is DataMeer.

IBM offers it’s famous Cognos Business intelligence suite as cloud solution as well.

IBM Data Refinery also offers data exploration and visualization functionality in Watson

Studio.

Again, these are just some examples of a rapidly changing and growing commercial ecosystem

among a huge number of established and emerging vendors.

In Watson Studio, an abundance of different visualizations can be used to better understand

data.

For example, this 3D bar chart enables you to visualize a target value on the vertical

dimension, which is dependent on two other values on the horizontal dimensions.

Coloring enables you to visualize a third dimension.

Hierarchical edge bundling enables you to visualize correlations and affiliations between

entities.

If sufficient, a classic bar chart can do the job as well, whereas a 2D scatter plot

with a heat map shows two dependent data fields, one on the y axis and one as color intensity.

A tree map shows distribution of subsets within a set, the famous pie chart does the same

but in a non-hierarchical manner, and finally, a word cloud pops out significant terms in

a document corpus.

Model building can be done using a service such as Watson Machine Learning.

Watson Machine Learning can train and build models using various open source libraries.

Google has a similar service on their cloud called AI Platform Training.

Nearly every cloud provider has a solution for this task.

Model deployment in commercial software is usually tightly integrated to the model building

process.

Here is an example of the SPSS Collaboration and Deployment Services, which can be used

to deploy any type of asset created by the SPSS software tools suite.

The same holds for other vendors.

In addition, commercial software can export models in an open format.

As an example, SPSS Modeler supports exporting models as Predictive Model Markup Language,

or “PMML,” which can be read by numerous other commercial and open software packages.

Watson Machine Learning can also be used to deploy a model and make it available to consumers

using a REST interface.

Amazon SageMaker Model Monitor is an example of a cloud tool that continuously monitors

deployed machine learning and deep learning models.

Again, every major cloud provider has similar tooling.

This is also the case for Watson OpenScale.

OpenScale and Watson Studio…

…unify the landscape.

Everything marked in green can be done using Watson Studio and Watson OpenScale.

We’ll cover Open Scale will be covered in a later video.

You’ve learned how the most common tasks in data science are supported by commercial

cloud tools.

Integration provides us the ability to use the same tools for multiple tasks.

In the next videos, we’ll look at packages, APIs, datasets, and models for data science.

#### Practice Quiz - Tools

**TOTAL POINTS 6**

1.

Question 1

Which of the following are common tasks in data science? (Select all that apply)

1 point



Data Management



Data Integration and Transformation



Data Visualization



Model Building



Model Deployment



Model Monitoring and Assessment



All of the above

2.

Question 2

Which of the following are data management tools? (Select all that apply.)

1 point



GitHub



MySQL



PostgreSQL



KubeFlow



PixieDust

3.

Question 3

Which of the following are Data Integration and Transformation tools? (Select all that apply.)

1 point



Cassandra



Apache Kafka



Apache Nifi



Apache AirFlow



Ceph

4.

Question 4

Which statement about JupyterLab is correct?

1 point



JuypterLab can run R and Python code only.



JuypterLab can run R code only.



JuypterLab can run R and Python code in addition to other programming languages.



JuypterLab can run Python code only.

5.

Question 5

Which statement about RStudio is correct?

1 point



RStudio is the primary choice for development in the R programming language.



RStudio is the primary choice for web development.



RStudio is the primary choice for development in the Python programming language.

6.

Question 6

Which statements about IBM Watson Studio and OpenScale are correct? (Select all that apply.)

1 point



Watson Studio together with Watson OpenScale is a database management system.



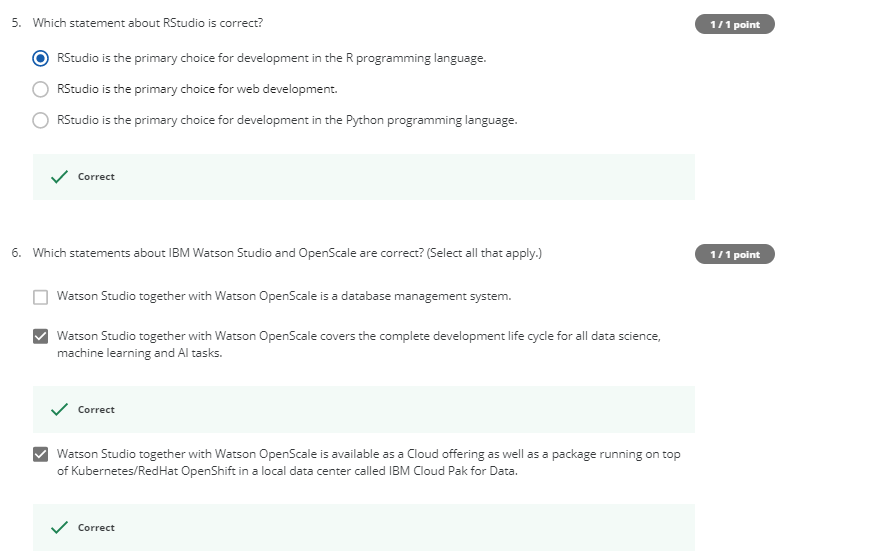
Watson Studio together with Watson OpenScale covers the complete development life cycle for all data science, machine learning and AI tasks.



Watson Studio together with Watson OpenScale is available as a Cloud offering as well as a package running on top of Kubernetes/RedHat OpenShift in a local data center called IBM Cloud Pak for Data.

Honor Code Agreement

I, **Dr.SIREESHA NANDURI**, understand that submitting work that isn’t my own may result in permanent failure of this course or deactivation of my Coursera account. Learn more about Coursera’s Honor Code



Libraries for Data Science

[Prev](https://www.coursera.org/learn/open-source-tools-for-data-science/quiz/A8y9q/practice-quiz-tools)

In this video, we will review several data science libraries.

Libraries are a collection of functions and methods that enable you to perform a wide

variety of actions without writing the code yourself.

We will focus on Python libraries: Scientific Computing Libraries in Python

Visualization Libraries in Python High-Level Machine Learning and Deep Learning

Libraries – “High-level” simply means you don’t have to worry about details, although

this makes it difficult to study or improve Deep Learning Libraries in Python

Libraries used in other languages

Libraries usually contain built-in modules providing different functionalities that you

can use directly; these are sometimes called “frameworks.”

There are also extensive libraries, offering a broad range of facilities.

Pandas offers data structures and tools for effective data cleaning, manipulation, and

analysis.

It provides tools to work with different types of data.

The primary instrument of Pandas is a two-dimensional table consisting of columns and rows.

This table is called a “DataFrame” and is designed to provide easy indexing so you

can work with your data.

NumPy libraries are based on arrays, enabling you to apply mathematical functions to these

arrays.

Pandas is actually built on top of NumPy

Data visualization methods are a great way to communicate with others and show the meaningful

results of analysis.

These libraries enable you to create graphs, charts and maps.

The Matplotlib package is the most well-known library for data visualization, and it’s

excellent for making graphs and plots.

The graphs are also highly customizable.

Another high-level visualization library, Seaborn, is based on matplotlib.

Seaborn makes it easy to generate plots like heat maps, time series, and violin plots.

For machine learning, the Scikit-learn library contains tools for statistical modeling, including

regression, classification, clustering and others.

It is built on NumPy, SciPy, and matplotlib, and it’s relatively simple to get started.

For this high-level approach, you define the model and specify the parameter types you

would like to use.

For deep learning, Keras enables you to build the standard deep learning model.

Like Scikit-learn, the high-level interface enables you to build models quickly and simply.

It can function using graphics processing units (GPU), but for many deep learning cases

a lower-level environment is required.

TensorFlow is a low-level framework used in large scale production of deep learning models.

It’s designed for production but can be unwieldy for experimentation.

Pytorch is used for experimentation, making it simple for researchers to test their ideas

Apache Spark is a general-purpose cluster-computing framework that enables you to process data

using compute clusters.

This means that you process data in parallel, using multiple computers simultaneously.

The Spark library has similar functionality as

Pandas Numpy

Scikit-learn

Apache Spark data processing jobs can use Python

R Scala, or SQL

There are many libraries for Scala, which is predominately used in data engineering

but is also sometimes used in data science.

Let’s discuss some of the libraries that are complementary to Spark

Vegas is a Scala library for statistical data visualizations.

With Vegas, you can work with data files as well as Spark DataFrames.

For deep learning, you can use BigDL.

R has built-in functionality for machine learning and data visualization, but there are also

several complementary libraries: ggplot2 is a popular library for data visualization

in R. You can also use libraries that enable you

to interface with Keras and TensorFlow.

R has been the de-facto standard for open source data science but it is now being superseded

by Python.

#### Application Programming Interfaces (API)

In this video we will discuss Application Programming Interfaces, or APIs. Specifically,

we will discuss: What an API is, API Libraries, REST APIs, including:

Request and Response. An API lets two pieces of software talk to each other. For example

you have your program, you have some data, you have other software components. You use

the API to communicate with the other software components.You don’t have to know how the

API works, you just need to know its inputs and outputs. Remember, the API only refers

to the interface, or the part of the library that you see. The “library” refers to

the whole thing. Consider the pandas library. Pandas is actually a set of software components,

many of which are not even written in Python. You have some data.

You have a set of software components. We use the pandas API to process the data

by communicating with the other software components. There can be a single software component at

the back end, but there can be a separate API for different languages. Consider TensorFlow,

written in C++. There are separate APIs in Python, JavaScript,

C++ Java, and Go. The API is simply the interface.

There are also multiple volunteer-developed APIs for TensorFlow; for example Julia, MATLAB,

R, Scala, and many more. REST APIs are another popular type of API. They enable you to communicate

using the internet, taking advantage of storage, greater data

access, artificial intelligence algorithms, and many other resources. The RE stands for

“Representational,” the S stands for “State,” the T stand for “Transfer.” In rest APIs,

your program is called the “client.” The API communicates with a web service that you

call through the internet. A set of rules governs Communication, Input or Request, and

Output or Response. Here are some common API-related terms. You or your code can be thought of

as a client. The web service is referred to as a resource. The client finds the service

through an endpoint. The client sends the request to the resource and the response to

the client. HTTP methods are a way of transmitting data over the internet We tell the REST APIs

what to do by sending a request. The request is usually communicated through an HTTP message.

The HTTP message usually contains a JSON file, which contains instructions for the operation

that we would like the service to perform. This operation is transmitted to the web service

over the internet. The service performs the operation. Similarly, the web service returns

a response through an HTTP message, where the information is usually returned using

a JSON file. This information is transmitted back to the

client. The Watson Speech to Text API is an example

of a REST API. This API converts speech to text. In the API call, you send a copy of

the audio file to the API; this process is called a post request. The API then sends

the text transcription of what the individual is saying. The API is making a get request.

The Watson Language-Translator API provides another example. You send the text you would

like to translate into the API, the API translates the text and sends the translation back to

you. In this case we translate English to Spanish. In this video, we’ve discussed

what an API is, API Libraries, REST APIs, including Request and Response. Thank you

for watching this video.

#### Data Sets - Powering Data Science

In this video we’ll discuss data sets: what they are, why they are important in data science,

and where to find them.

Let’s first loosely define what a data set is.

A data set is a structured collection of data.

Data embodies information that might be represented as text, numbers, or media such as images,

audio, or video files.

A data set that is structured as tabular data comprises a collection of rows, which in turn

comprise columns that store the information.

One popular tabular data format is "comma separated values," or CSV.

A CSV file is a delimited text file where each line represents a row and data values

are separated by a comma.

For example, imagine a data set of observations from a weather station.

Each row represents an observation at a given time, while each column contains information

about that particular observation, such as the temperature, humidity, and other weather

conditions.

Hierarchical or network data structures are typically used to represent relationships

between data.

Hierarchical data is organized in a tree-like structure, whereas network data might be stored

as a graph.

For example, the connections between people on a social networking website are often represented

in the form of a graph.

A data set might also include raw data files, such as images or audio.

The MNIST dataset is popular for data science.

It contains images of handwritten digits and is commonly used to train image processing

systems.

Traditionally, most data sets were considered to be private because they contain proprietary

or confidential information such as customer data, pricing data, or other commercially

sensitive information.

These data sets are typically not shared publicly.

Over time, more and more public and private entities such as scientific institutions,

governments, organizations and even companies have started to make data sets available to

the public as “open data," providing a wealth of information for free.

For example, the United Nations and federal and municipal governments around the world

have published many data sets on their websites, covering the economy, society, healthcare,

transportation, environment, and much more.

Access to these and other open data sets enable data scientists, researchers, analysts, and

others to uncover previously unknown and potentially useful insights.

They can create new applications for both commercial purposes and the public good.

They can also carry out new research.

Open data has played a significant role in the growth of data science, machine learning,

and artificial intelligence and has provided a way for practitioners to hone their skills

on a wide variety of data sets.

There are many open data sources on the internet.

You can find a comprehensive list of open data portals from around the world on the

Open Knowledge Foundation’s datacatalogs.org website.

The United Nations, the European Union, and many other governmental and intergovernmental

organizations maintain data repositories providing access to a wide range of information.

On Kaggle, which is a popular data science online community, you can find and contribute

data sets that might be of general interest.

Last but not least, Google provides a search engine for data sets that might help you find

the ones that have particular value for you.

It’s important to recognize that open data distribution and use might be restricted,

as defined by its licensing terms.

In absence of a license for open data distribution, many data sets were shared in the past under

open source software licenses.

These licenses were not designed to cover the specific considerations related to the

distribution and use of data sets.

To address the issue, the Linux Foundation created the Community Data License Agreement,

or CDLA.

Two licenses were initially created for sharing data: CDLA-Sharing and CDLA-Permissive.

The CDLA-Sharing license grants you permission to use and modify the data.

The license stipulates that if you publish your modified version of the data you must

do so under the same license terms as the original data.

The CDLA-Permissive license also grants you permission to use and modify the data.

However, you are not required to share changes to the data.

Note that neither license imposes any restrictions on results you might derive by using the data,

which is important in data science.

Let’s say, for example, that you are building a model that performs a prediction.

If you are training the model using CDLA-licensed data sets, you are under no obligation to

share the model, or to share it under a specific license if you do choose to share it.

In this video you’ve learned about open data sets, their role in data science, and

where to find them.

We’ve also introduced the Community Data License Agreement, which makes it easier to

share open data.

One important aspect that we didn’t cover in this video is data quality and accuracy,

which might vary greatly depending on who collected and contributed the data set.

While some open data sets might be good enough for personal use, they might not meet enterprise

requirements due to the impact they might have on the business.

In the next module, you will learn about the Data Asset eXchange, a curated open data repository.

#### Sharing Enterprise Data - Data Asset eXchange

Despite the growth of open data sets that are available to the public, it can still

be difficult to discover data sets that are both high quality and have clearly defined

license and usage terms.

To help solve this challenge, IBM created the Data Asset eXchange, or "DAX,”, which

we’ll introduce in this video.

DAX provides a trusted source for finding open data sets that are ready for to use in

enterprise applications.

These data sets and which cover a wide variety of domains, including images, video, text,

and audio.

Because DAX provides a high level of curation for data set quality, as well as licensing

and usage terms, DAX data sets are typically easier to adopt, whether in research or commercial

projects.

Wherever possible, DAX aims to make data sets available under one of the variants of the

CDLACommunity Data License Agreement, in order to foster data sharing and collaboration.

DAX also provides a single place to access unique data sets, in particular from IBM Research

projects.

To make it easier for developers to get started with using the data sets, DAX also provides

tutorials in the form of notebooks that walk through the basics of data cleaning, pre-processing,

and exploratory analysis.

For some data sets, there are also notebooks illustrating how to perform more complex analysis,

such as creating charts, statistical analysis, time-series analysis, training machine learning

models, and integrating deep learning via using the Model Asset eXchange, (a project

closely related to DAX and also available on the IBM Developer website).

In this way, DAX helps developers to create end-to-end analytic and machine learning workflows

and to consume open data and models with confidence under clearly defined license terms.

Let’s say you’ve found a data set that might be of interest to you.

On the data set page you can download the compressed data set archive from cloud storage,

explore the data set using Jupyter Notebooks, review the data set metadata, such as format,

licensing terms and size, and preview some parts of the data set.

Most data sets on DAX are complemented by one or more Jupyter Notebooks that you can

use to perform data cleaning, pre-processing, and exploratory analysis.

These notebooks run "as is"as is in Watson Studio, IBM’s Data Sciencedata science platform.

Jupyter Notebooks and Watson Studio are covered later during in this course.

In this video, you’ve learned about IBM’s open data repository, the Data Asset eXchange.

In the hands-on lab you’ll have a chance to explore the repository.

#### Machine Learning Models

In this video, we’ll introduce you to machine learning and deep learning models.

Data contains a wealth of information that can be used to solve certain types of problems.

Traditional data analysis approaches, such as a person manually inspecting the data or

a specialized computer program that automates the human analysis, quickly reach their limits

due to the amount of data to be analyzed or the complexity of the problem.

Machine learning uses algorithms – also known as ”models” - to identify patterns

in the data.

The process by which the model learns these patterns from data is called “model training."

Once a model is trained, it can then be used to make predictions.

When the model is presented with new data, it tries to make predictions or decisions

based on the patterns it has learned from past data.

Machine learning models can be divided into three basic classes: supervised learning,

unsupervised learning, and reinforcement learning.

Supervised learning is one of the most commonly used type of machine learning models.

In supervised learning, a human provides input data and the correct outputs.

The model tries to identify relationships and dependencies between the input data and

the correct output.

Generally speaking, supervised learning is used to solve regression and classification

problems.

Let’s look at an example for each problem type:

Regression models are used to predict a numeric, or “real," value.

For example, given information about past home sales, such as geographic location, size,

number of bedrooms, and sales price, you can train a model to predict the estimated sales

price for other homes with similar characteristics.

Classification models are used to predict whether something belongs to a category, or

“class."

For example, given a set of emails along with a designation of whether or not they are considered

spam, an algorithm can be trained to identify unsolicited emails.

In unsupervised learning, the data is not labelled by a human.

The models must analyze the data and try to identify patterns and structure within the

data based only on the characteristics of the data itself.

Clustering and anomaly detection are two examples of this learning style.

Clustering models are used to divide each record of a data set into one of a small number

of similar groups.

An example of a clustering model could be providing purchase recommendations for an

e-commerce store based on past shopping behavior and the content of a shopping basket.

Anomaly detection identifies outliers in a data set, such as fraudulent credit card transactions

or suspicious online log-in attempts.

The third type of learning, reinforcement learning, is loosely based on the way human

beings and other organisms learn.

Think about a mouse in a maze.

If the mouse gets to the end of the maze it gets a piece of cheese.

This is the “reward” for completing a task.

The mouse learns – through trial and error – how to get through the maze to get as

much cheese as it can.

In a similar way, a reinforcement learning model learns the best set of actions to take,

given its current environment, in order to get the most reward over time.

This type of learning has recently been very successful in beating the best human players

in games such as go, chess, and popular strategy video games.

Deep learning is a specialized type of machine learning.

It refers to a general set of models and techniques that tries to loosely emulate the way the

human brain solves a wide range of problems.

It is commonly used to analyze natural language, both spoken and text, as well as images, audio,

and video, to forecast time series data and much more.

Deep learning has had a lot of recent success in these and other areas and is therefore

becoming an increasingly popular and important tool for data science.

Deep learning typically requires very large data sets of labeled data to train a model,

is compute-intensive, and usually requires special purpose hardware to achieve acceptable

training times.

You can build a custom deep learning model from scratch or use pre-trained models from

public model repositories.

Deep learning models are implemented using popular frameworks such as TensorFlow, PyTorch,

and Keras.

Deep learning frameworks typically provide a Python API, and many support other programming

languages, such as C++ and JavaScript.

You can download pre-trained state-of-the-art models from repositories that are commonly

referred to as model "zoos."

Popular model zoos include those provided by TensorFlow, PyTorch, Keras, and ONNX.

Models are also published by academic and commercial research groups.

While it is beyond the scope of this video to explain in detail how you would go about

building a model, let’s briefly outline the high-level tasks using an example.

Assume you want to enable an application to identify objects in images by training a deep

learning model.

First, you collect and prepare data that will be used to train a model.

Data preparation can be a time-consuming and labor-intensive process.

In order to train a model to detect objects in images, you need to label the raw training

data by, for example, drawing bounding boxes around objects and labeling them.

Next, you build a model from scratch or select an existing model that might be well suited

for the task from a public or private resource.

You then train the model on your prepared data.

During training, your model learns from the labeled data how to identify objects that

are depicted in an image.

Once training has commenced, you analyze the training results and repeat the process until

the trained model performance meets your requirements.

When the trained model performs as desired, you deploy it to make it available to your

applications.

In this video, you’ve learned about machine learning and deep learning, what they are

used for, and where to find open source models.

In the next video, we’ll introduce you to the Model Asset eXchange, a curated collection

of ready-to-use and customizable deep learning models.

#### The Model Asset Exchange

n this video, we will introduce you to the Model Asset eXchange on IBM Developer, a free

open source resource for deep learning models.

Throughout the video we will refer to the Model Asset eXchange as "MAX."

In the previous video, we briefly outlined the high-level tasks you need to complete

to train a model from scratch.

Due to the amount of data, labor, time, and resources required to complete the tasks,

time to value can be quite long.

To reduce time to value, consider taking advantage of pre-trained models for certain types of

problems.

These pre-trained models can be ready to use right away, or they might take less time to

train.

The Model Asset eXchange is a free open source repository for ready-to-use and customizable

deep learning microservices.

These microservices are configured to use pre-trained or custom-trainable state-of-the-art

deep learning models to solve common business problems.

These models have been reviewed, tested, and can be quickly deployed in local and cloud

environments.

All models in MAX are available under permissive open source licenses, making it easier to

use them for personal and commercial purposes and reducing the risk of legal liabilities.

On MAX, you can find models for a variety of domains, including image, audio, video,

and natural language analysis.

This list includes a small selection.

In the lab for this module, you’ll have a chance to explore those models.

Let’s take a look at the components of a typical model-serving microservice.

Each microservice includes the following components:

A pre-trained deep learning model.

Code that pre-processes the input before it is analyzed by the model and code that post-processes

the model output.

A standardized public API that makes the services’ functionality available to applications.

The MAX model-serving microservices are built and distributed as open-source Docker images.

Docker is a container platform that makes it easy to build applications and to deploy

them in a development, test, or production environment.

The Docker image source is published on GitHub and can be downloaded, customized as needed,

and used in personal or commercial environments.

You can deploy and run these images in a test or production environment using Kubernetes,

an open-source system for automating deployment, scaling, and management of containerized applications

in private, hybrid, or public clouds.

A popular enterprise-grade Kubernetes platform is Red Hat OpenShift, which is available on

IBM Cloud, Google Cloud Platform, Amazon Web Services, and Microsoft Azure.

The model-serving microservices expose a REST API that developers can use to incorporate

deep learning into their applications and services.

Because REST APIs can be consumed using any programming language, you can easily integrate

these services into your existing ecosystem.

The API exposes a prediction endpoint and one or more metadata endpoints.

This example shows the endpoints for the Object Detection microservice.

The /model/predict endpoint takes an image as input and returns as a response a list

of objects that were detected in the image, along with bounding box coordinates that identify

where the detected object is located.

Some prediction endpoints can also accept additional input parameters that impact the

produced results, such as filters.

This microservice exposes two metadata endpoints, /model/labels and /model/metadata.

These endpoints provide information such as the objects that can be detected and the deep

learning model used to derive the answer given the input.

In the lab portion of this module, you will have a chance to explore and test these endpoints

using a web browser.

Each endpoint accepts application-friendly inputs, such as an image in JPG, PNG, or GIF

format, instead of a model-specific data structure.

Each endpoint also generates application-friendly outputs, such as standardized JSON, which

is a lightweight data-interchange format.

Let’s take a closer look at what happens when an application invokes the prediction

endpoint.

In this example, a user has selected an image in a web application, the prediction endpoint

is invoked, and the image is uploaded.

The microservice prepares the input image for processing, runs the deep learning model

that identifies objects in the image, generates a response using the prediction results, and

returns the result to the application.

The application renders the results by drawing bounding boxes and labels.

In this video, we’ve introduced the Model Asset eXchange, a free and open source repository

for microservices that make deep learning functionality available to applications and

services in local and cloud environments.

In the lab, you will have a chance to try a model-serving microservice, explore its

API, and learn more about how you can leverage it from a web application and an Internet

of Things application.

# **Lab: Getting Started with the Model Asset Exchange and the Data Asset Exchange**

In this lab you will explore the Model Asset Exchange (MAX) and the Data Asset Exchange (DAX), which are two open source Data Science resources on IBM Developer.

Upon completion of part 1 of this lab (“Explore deep learning models”) you will be able to:

* Find ready-to-use deep learning models on the Model Asset Exchange
* Locate resources that guide you through deployment in a local or cloud environment
* Explore the deep learning model-serving microservice API using your web browser
* Articulate how developers can consume those microservices

Upon completion of part 2 (“Explore deep learning data sets”) you will know:

* Where to find open data sets on IBM Developer
* How to explore those data sets

It will take you approximately 30 minutes to complete the lab. Only a web browser is required to complete the tasks.

## Part 1: Explore deep learning models

The Model Asset Exchange is a curated repository of open source deep learning models for a variety of domains, such as text, image, audio, and video processing.

1. Open <https://developer.ibm.com/> in your web browser.

2. From the main menu select **Open Source at IBM** > **Model Asset eXchange**. The MAX home page is displayed.

#### Practice Quiz - Packages, APIs, Data Sets, Models

**TOTAL POINTS 6**

1.

Question 1

Which scientific computing library provides data structures and data analysis tools for Python?

1 point



TensorFlow



Pandas



YumPies



Seahorse

2.

Question 2

What does the acronym API stand for?

1 point



Abstract Python Interface



Abstract Programming Interface



Algorithmic Programming Interface



Application Programming Interface

3.

Question 3

True or False: Open data is always distributed under a Community Data License Agreement.

1 point



True



False

4.

Question 4

Which of the following is not a type of Machine Learning?

1 point



Reinforcement learning



Supervised teaching



Supervised learning



Unsupervised learning

5.

Question 5

Which of the following is NOT a deep learning framework?

1 point



Tommy



TensorFlow



Keras



PyTorch

6.

Question 6

Fill in the blank: The MAX model-serving microservices expose a \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ that applications use to consume a model.

1 point



Java API



REST API



Python API



Scala API

Honor Code Agreement

I, **Dr.SIREESHA NANDURI**, understand that submitting work that isn’t my own may result in permanent failure of this course or deactivation of my Coursera account. Learn more about Coursera’s Honor Code

#### Graded Quiz

**TOTAL POINTS 10**

1.

Question 1

Which are the three most used languages for data science? (Select all that apply.)

1 point



SQL



R



Scala



Java



Python

2.

Question 2

Which of these is a database query language?

1 point



Julia



Python



SQL



All of the Above

3.

Question 3

Is it possible to use machine learning within a web browser with Javascript?

1 point



Yes



No

4.

Question 4

Which of these is not a machine learning or deep learning library for Python?

1 point



PyTorch



Keras



Scikit-learn



NumPy

5.

Question 5

Comma Separated Values (CSV) is a commonly used format to store:

1 point



Hierarchical or network data



Tabular data



All of the above

6.

Question 6

Classification models can be used to determine whether:

1 point



An email is likely spam.



A video contains a specific sound.



An image contains a dog.



All of the above.

7.

Question 7

Generally speaking, which type of model is used to predict a numerical value, such as the potential sales price of a used car?

1 point



Classification model



Clustering model



Regression model

8.

Question 8

Fill in the blank: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is the heart of every organization.

1 point



Open source



The cloud



Integration



Data

9.

Question 9

What does the “BI” in BI Tools stand for?

1 point



Business Integration



Business Intelligence



Build Integration



Build Information

10.

Question 10

Which of the following are true about Data Asset Management?

1 point



Also known as data governance.



A crucial part of data science at the enterprise level.



To be done effectively data must be versioned and annotated with meta data.



All of the above.

WEEK-2

**Key Concepts**

* Explain how to use GitHub to create and manage source code for data science projects.
* Describe the features of Jupyter Notebook that make it popular for data science projects.
* Describe the features of RStudio IDE that make it popular for data science projects.

#### Practice Quiz - GitHub

**TOTAL POINTS 3**

1.

Question 1

Which of the following statements are true? (Select all that apply.)

1 point



Git is very useful for data science as well, since data science often involves a lot of source code to be written and managed.



Git is an integrated development environment for data science.



Git is a system for version control of source code.

2.

Question 2

Which of the following statements about repositories are correct? (Select all that apply.)

1 point



The remote repository is only accessible by myself.



The local repository is only accessible by myself.



The staging is only accessible by myself.



The remote repository is accessible by all contributors.



The local repository is accessible by all contributors.

3.

Question 3

What is the best process for contributing a bugfix to a foreign repository?

1 point



Ask the repository owner for write access to the repository.



Fork the repository, update the fork and create a pull request.



Send the fix via email to the author.

Honor Code Agreement

I, **Dr.SIREESHA NANDURI**, understand that submitting work that isn’t my own may result in permanent failure of this course or deactivation of my Coursera account. Learn more about Coursera’s Honor Code

#### Practice Quiz - Jupyter Notebook

**TOTAL POINTS 3**

1.

Question 1

Which of the following functions do Jupyter Notebook unify?

1 point



Editing and execution of source code.



Editing and display of documentation.



Visualization of charts.



All of the above.

2.

Question 2

Which statement is true about Jupyter Notebook?

1 point



Jupyter Notebook is a commercial product of IBM.



Jupyter Notebook is free and open source.

3.

Question 3

What is a Jupyter Notebook kernel?

1 point



It is part of the operating system the Jupyter server runs on.



It is a wrapper running on the Jupyter server encapsulating the programming language interpreter.

Honor Code Agreement

I, **Dr.SIREESHA NANDURI**, understand that submitting work that isn’t my own may result in permanent failure of this course or deactivation of my Coursera account. Learn more about Coursera’s Honor Code

#### Graded Quiz

**TOTAL POINTS 10**

#### Practice Quiz - RStudio IDE

**TOTAL POINTS 3**

1.

Question 1

Which of the following functions does RStudio unify? (Select all that apply.)

1 point



Storing of data.



Editing and execution of source code.



Display of the R Console.



Visualization of plots.



Visualization of data in table form.

2.

Question 2

Which statement is true about the RStudio IDE?

1 point



RStudio is free and open source.



RStudio is a commercial product of IBM.

3.

Question 3

Which statement about R packages is correct?

1 point



R currently supports more than 15,000 packages which can be installed to extend R's functionality.



R doesn't require any packages to be installed since it contains all functionality necessary which a data scientists ever requires.

Honor Code Agreement

I, **Dr.SIREESHA NANDURI**, understand that submitting work that isn’t my own may result in permanent failure of this course or deactivation of my Coursera account. Learn more about Coursera’s Honor Code

1.

Question 1

Which are the two most used open source tools for data science?

1 point



Notepad



RStudio



Jupyter Notebooks / JupyterLab



Spyder



VSCode

2.

Question 2

What tool do most R developers use?

1 point



RStudio



Jupyter Notebooks / JupyterLab

3.

Question 3

What tool do most Python developers use?

1 point



Jupyter Notebooks / JupyterLab



RStudio

4.

Question 4

True or false? Jupyter Notebooks / JupyterLab support development in R.

1 point



True



False

5.

Question 5

Which tool unifies documentation, source code and data visualizations into a single document?

1 point



Notepad



Jupyter Notebooks / JupyterLab



VSCode

6.

Question 6

Which command is used to install packages in R?

1 point



package("package name")



install.package("package name")



install("package name")



install.packages("package name")

7.

Question 7

Which of the following functions does RStudio provide?

1 point



Creating relationships between data tables.



Documenting R code applications.



Editing and execution of R code.



Storing data in tables.

8.

Question 8

True or False: The Jupyter Notebook kernel must be installed on a local server.

1 point



True



False

9.

Question 9

Which of the following statements about Jupyter Notebook is correct?

1 point



Jupyter Notebook supports the Visualization of data in charts.



Jupyter Notebook provides storage of massive quantities of data in data lakes.



Jupyter Notebook is a commercial product of IBM.



Jupyter Notebook is only available if installed locally on your computer.

10.

Question 10

True or false? RStudio supports development in Python.

1 point



True



False

Honor Code Agreement

I, **Dr.SIREESHA NANDURI**, understand that submitting work that isn’t my own may result in permanent failure of this course or deactivation of my Coursera account. Learn more about Coursera’s Honor Code

**Less**

Overview of Git/GitHub

In this video, you’ll get an overview of Git and GitHub, which are popular environments

among developers and data scientists for performing version control of source code files and projects

and collaborating with others.

You can’t talk about Git and GitHub without a basic understanding of what version control is.

Play video starting at 30 seconds and follow transcript0:30

A version control system allows you to keep track of changes to your documents.

This makes it easy for you to recover older versions of your document if you make a mistake,

and it makes collaboration with others much easier.

Here is an example to illustrate how version control works.

Let’s say you’ve got a shopping list and you want your roommates to confirm the things

you need and add additional items.

Without version control, you’ve got a big mess to clean up before you can go shopping.

With version control, you know EXACTLY what you need after everyone has contributed their ideas.

Play video starting at 1 minute 9 seconds and follow transcript1:09

Git is free and open source software distributed under the GNU General Public License.

Git is a distributed version control system, which means that users anywhere in the world

can have a copy of your project on their own computer; when they’ve made changes, they

can sync their version to a remote server to share it with you.

Git isn’t the only version control system out there, but the distributed aspect is one

of the main reasons it’s become one of the most common version control systems available.

Version control systems are widely used for things involving code, but you can also version

control images, documents, and any number of file types.

You can use Git without a web interface by using your command line interface, but

GitHub is one of the most popular web-hosted services for Git repositories.

Others include GitLab, BitBucket, and Beanstalk.

There are a few basic terms that you will need to know before you can get started.

The SSH protocol is a method for secure remote login from one computer to another.

A repository contains your project folders that are set up for version control.

A fork is a copy of a repository.

A pull request is the way you request that someone reviews and approves your changes

before they become final.

A working directory contains the files and subdirectories on your computer that are associated

with a Git repository.

There are a few basic Git commands that you will always use.

When starting out with a new repository, you only need create it once: either locally,

and then push to GitHub, or by cloning an existing repository by using the command "git init".

Play video starting at 3 minutes 3 seconds and follow transcript3:03

"git add" moves changes from the working directory to the staging area.

"git status" allows you to see the state of your working directory and the staged snapshot

of your changes.

"git commit" takes your staged snapshot of changes and commits them to the project.

"git reset" undoes changes that you’ve made to the files in your working directory.

"git log" enables you to browse previous changes to a project.

"git branch" lets you create an isolated environment within your repository to make changes.

"git checkout" lets you see and change existing branches.

"git merge" lets you put everything back together again.

To learn how to use Git effectively and begin collaborating with data scientists around

the world, you will need to learn the essential commands.

Luckily for us, GitHub has amazing resources available to help you get started.

Go to try.github.io to download the cheat sheets and run through the tutorials.

In the following modules, we'll give you a crash course on setting up your local environment

and getting started on a project.

WEEK-3

#### What is IBM Watson Studio?

Every business wants to work smarter, and to do that you need to tap into your

company's greatest resource, your data. But extracting the full value out of

your data isn't always an easy process. First. you end up juggling an incredibly

large and complex collection of tools that are used for finding and cleaning

data, analyzing and generating visualizations of that data, and using

the data to build and deploy machine learning models. And to make matters

worse these tools are often a time drain to individually manage, and can be

difficult to integrate into your system, which can really slow down the workflow.

But not anymore. Using Watson Studio you can simplify

your data projects with a streamlined process, that allows you to extract value

and insights from your data to help your business get smarter, faster. It delivers

an easy-to-use collaborative data science and machine learning environment

for building and training models, preparing and analyzing data, and sharing

insights, all in one place. Watson Studios easy to create

visualizations and drag-and-drop code put the power of database

decision-making into the hands of any member of your organization with no need

for IT assistance. And if you need access to open source tools, the environment

offers some of the most popular and powerful ones available. Watson Studio

single environment also creates a workflow that's incredibly efficient so

data scientists can share assets and work to solve problems within the system

rather than starting from scratch every time a new issue arises. And developers

can use that efficiency to quickly dive into building machine learning and deep

learning algorithms. In fact, in the area of deep learning,

Watson Studio supports some of the most popular frameworks and can deploy that

deep learning on to the latest GPUs to help accelerate modeling by making it

easier to use. The environments built-in neural network modeler also helps you

build models with a simplified graphical interface even if you don't have the

dedicated resources to build a model from scratch, Watson's Studio can help

you get started with modeling templates for areas such as visual recognition,

language classification, and other tools from IBM Watson services.

Because Watson Studio is seamlessly integrated with the IBM Watson Knowledge

Catalog, an intelligent asset discovery tool, you can transform data and models

into trusted enterprise resources and collaborate with confidence, without

compromising compliance, security or access control. Watson Studio provides

many benefits for organizations helping to infuse AI into the business and drive

innovation. You can train Watson Studio with embedded AI services including

watson visual recognition. You can customize your models and deploy them as

APIs or Core ML by using open source tools like Jupyter, Notebook, Anaconda

and RStudio. Watson Studio supports most popular code libraries as well as

no code visual modeling with neural network modeler for designing neural

architectures using the most popular deep learning frameworks. In Watson

Studio you can interactively discover, cleanse, and transform your data using

data refinery. It helps you understand the quality and distribution of your

data with built-in charts and statistics, and provides visualized results through

interactive dashboards. Watson Studio includes an intuitive drag-and-drop

interface that enables a non programmer to speed up the bottle building process

by visually selecting, configuring, designing and auto coding neural

networks. From development and training to production and evaluation, Watson

Studio tracks your models over time to ensure you have the best performance for

any given task using the best solutions across the entire lifecycle of your

machine learning models.

#### Watson Studio Introduction

Watson Studio is an integrated platform of tools services and data that

helps companies accelerate their shift to become data-driven organizations you

can start with a free account to explore its capabilities data science is a team

sport we have different types of people interested in the insights that data

science can provide this includes business analysts data engineers data

stewards data scientist and developers data needs to be located in cleanse

models have to be created tested monitored and updated all this requires

teamwork for this reason watson city was built as

a collaborative platform a community of like-minded people there is a lot to

cover in this introduction and will only scratch the surface you can find more

information on the digital technical engagement site at ibm.com slash demos

once you are logged in you may see the get started welcome screen you can

minimize the screen by clicking on the get started button in the upper right

one important item that is easy to miss is the hamburger button in the upper

left it gives you direct access to projects catalogs and services among

other things the gallery is particularly interesting it is a collection of assets

including tutorials notebooks datasets articles and papers from multiple

sources new assets are constantly added assets can be searched using filters for

Thai language technology topics and so on the results can be sorted by features

or by date manage gives you quick access to specific areas to manage finally we

have integrated support and documentation in the Watson environment

as mentioned earlier the project is the center of the collaboration it is very

simple to create a project you click create a project in the welcome screen

or new project and either create an empty project or

one from an existing one then you give it a name possibly add a description and

you're ready to go at the project level we also have a menu of options it starts

with the overview where you can see basic information on the project this

tab also includes a readme section where you can get more details on what the

project is about the next one is assets where you can see the data assets models

notebooks and other assets that are part of the project you can go to add

specific assets using the add to project drop down menu at the top of the screen

we won't go into all of those menu items but one important want to know is

connection this allows you to access data that comes from outside Watson

studio as you can see it includes a lot of data services from IBM but also quite

a few from third parties such as Amazon and Microsoft going back to our project

I'd like to point out the environment section one important tool for that

exploration data manipulation and model creation is the notebook depending on

the amount of work that needs to be done we have a choice of resource allocation

we can also tailor the environment to include additional libraries so we have

a complete environment from the start I want to point out two more selections

from the top menu access control and settings the access control allows you

to control collaborators and their permissions and more in the settings

section you can among other things add services for example you click on the

add service drop-down menu select Watson and add a machine-learning service you

have the choice to add an existing service you may have created earlier in

another project or create a new one note that most services include a light free

version this means that you can experiment with all sorts of

capabilities for free

you

#### Creating an Account on IBM Watson Studio

This video shows you how to try out IBM Watson for free.

IBM Watson gives you access to IBM Watson Studio, IBM Watson Knowledge

Catalog, the data refinery, machine and deep learning, visual recognition models,

dashboards and streams flows. At https://dataplatform.cloud.ibm.com/

you can sign up for a free trial. When you sign up for an IBM Watson account

you are automatically signed up for a free IBM Cloud account. Here you see the

Watson applications that will be provisioned. IBM Watson Studio and IBM

Watson Knowledge Catalog. If you already have an IBM cloud account

then use your IBM ID to sign up for IBM Watson.

Otherwise, type your email address which will be used to create an IBM cloud

account for you. On the next screen you are redirected to

the IBM cloud registration page where you need to provide the typical basic

information for an account then click create account.

Now check your email and confirm your account.

Now that you've registered for IBM cloud you can use those same credentials to

sign-in. Next you'll see the process create the

IBM Watson user account using your IBM cloud credentials and finally you'll see

that your account was successfully created.

This IBM account has only one Associated IBM cloud account and one resource group.

If you have more than one associated account or one account with multiple

resource groups then during the Watson Studio account setup you'll see this

screen giving you the option to select an account and resource group to use.

IBM Cloud uses resource groups as a way for you to organize your account resources

in customizable groupings so that you can quickly assign users access to more

than one resource at a time. View the settings to verify the

applications and services that are provisioned.

Now you're ready to start working in IBM Watson.

#### Jupyter Notebook in Watson Studio - Part 1

[Music]

this video covers the basics for working

with Jupiter notebooks in Watson studio

start in a Watson studio project and add

to the project a notebook

just provide a name in a description

and create the notebook

let's first load a file so you have some

data to work with from the files

slide-out panel browse to select the

file after the file is added to the

project its available to work with in

this notebook just click insert to code

and insert a panda's data frame

Play video starting at 47 seconds and follow transcript0:47

for running the notebook it's a best

practice to insert a cell at the top to

describe what the notebook does change

the cell type to markdown so this cell

will not be treated as code and then add

the description

now you're ready to run the notebook the

inserted code loads the data set into a

data frame using your credentials for

your cloud object storage instance and

then displays the first five rows of the

data set before returning to the project

save the notebook

Play video starting at 1 minute 23 seconds and follow transcript1:23

I'm the assets tab you'll find the

notebook if you open the notebook it

will be in read-only mode but you can

edit the notebook and make changes

for example you can access the info

panel and change the name of the

notebook and on the environment tab you

could change the environment used to run

the notebook

as well as stop or restart the runtime

environment

if you'd like to share a read-only

version of the notebook you can do that

from here you can select how much of the

content you'd like to share and how you

want to share the notebook either

through a link or social media

if you'd like to schedule the notebook

to run at a different time you can

create a job

just provide a name for the job and

select the scheduling options like

specifying a date for the job to run

and whether you'd like the job run to

repeat

after you create and run the job

you can see the status on the jobs tab

in the project

[Music]

#### Jupyter Notebook in Watson Studio - Part 2

[Music]

this video shows you how to create a

jupiter notebook let's start by adding a

data asset to the project you can either

browse to select files or drag files

into the panel

great now the data file is uploaded to

object storage and available as a data

asset in this project

next create a notebook

provide a name and a description

and then select the runtime to use when

running this notebook here you see the

environments you could use you'll learn

more about environments later so for now

just select the default spark Python

environment and verify the language and

spark version when you ready create the

notebook

now wait while the runtime environment

is instantiated once the environment is

ready in the notebook access the data

sources and locate the file click insert

to code and choose how you want to

insert the data the choices in this

drop-down box are dependent upon the

language used in this notebook and the

file type notice that the inserted code

includes the credentials you'll need to

read the data file from the object

storage instance

when you run the code the first five

rows display

now let's take a closer look at

environments on the environments tab you

can define the hardware size and

software configuration for the runtime

associated with Watson studio tools such

as notebooks you can see that there is

one active environment runtime namely

the runtime being used by the notebook

you just created and here are the other

default environments you can view any of

the default environments to see a

summary of the configuration

and also create a new environment

definition

first provide a name in a description

if you select spark for the type you'll

see some additional configuration

options in this case just accept the

defaults and choose Scala for the

software version when you ready create

the new environment

the environment is ready for you to use

with a notebook

to switch a notebook to use a different

environment you need to first stop the

colonel

Play video starting at 2 minutes 42 seconds and follow transcript2:42

then you can change the environment

and select the custom environment you

just created and associate that with the

notebook

now open the notebook in edit mode and

wait for the new environment to be

instantiated

since this notebook was last saved using

a different kernel you need to set the

new kernel

let's delete the existing cell

locate the source data file and insert a

spark session data frame

Play video starting at 3 minutes 21 seconds and follow transcript3:21

when you run the code the first five

rows display

now you're ready to explore the

community and find sample notebooks and

datasets to get started analyzing data

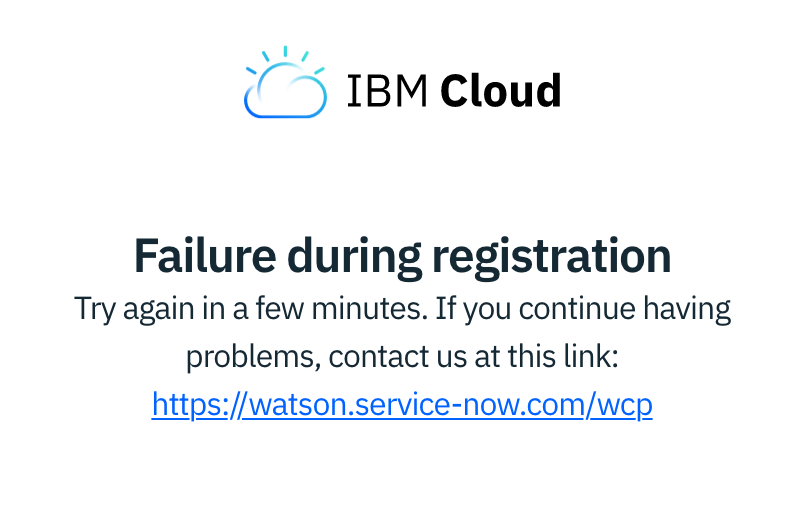
[Music]

# **Lab: Creating a Watson Studio Project with Jupyter Notebook**

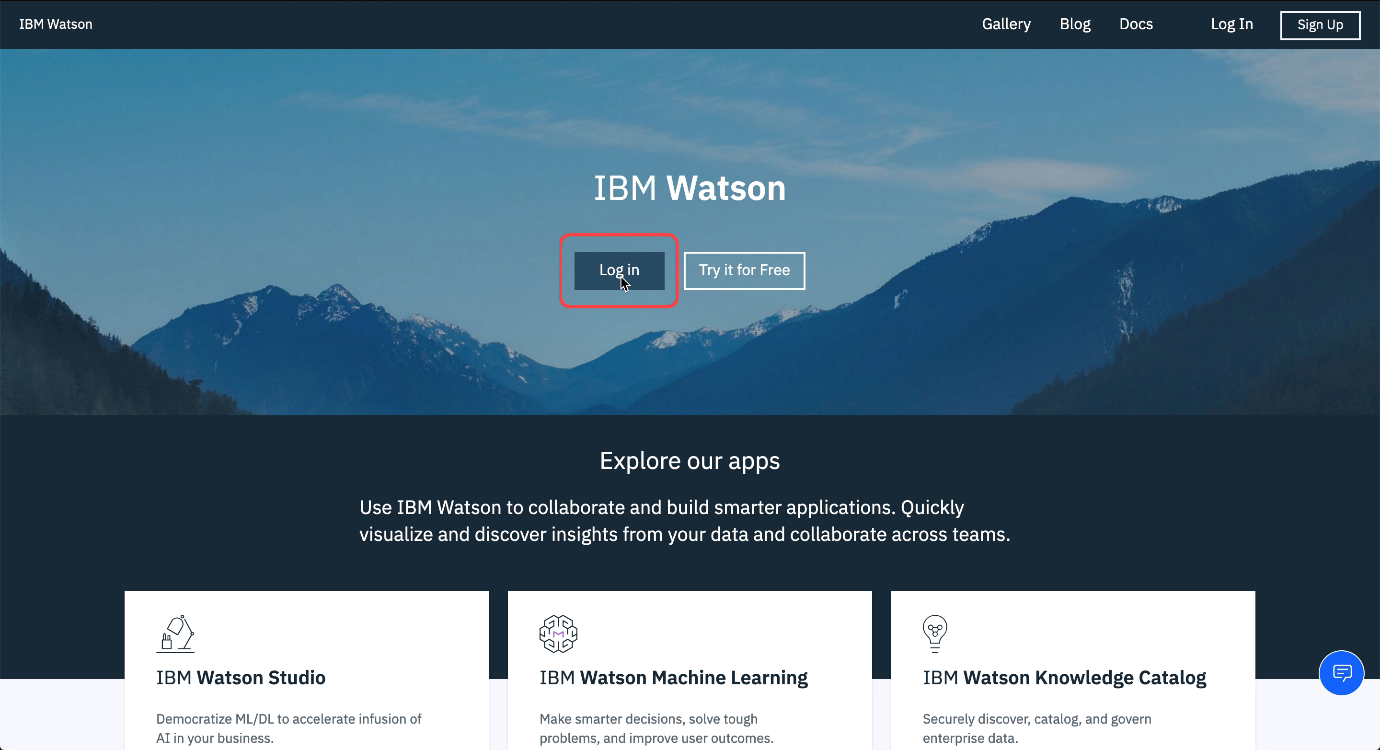
This tutorial walks you through setting up an account on the IBM Cloud and a project in Watson Studio such that you can use Jupyter Notebook for your work.

1. Please follow this link to create an [IBM Cloud Account](https://cloud.ibm.com/registration). It's completely for free, you don't need a credit card and the account never expires.

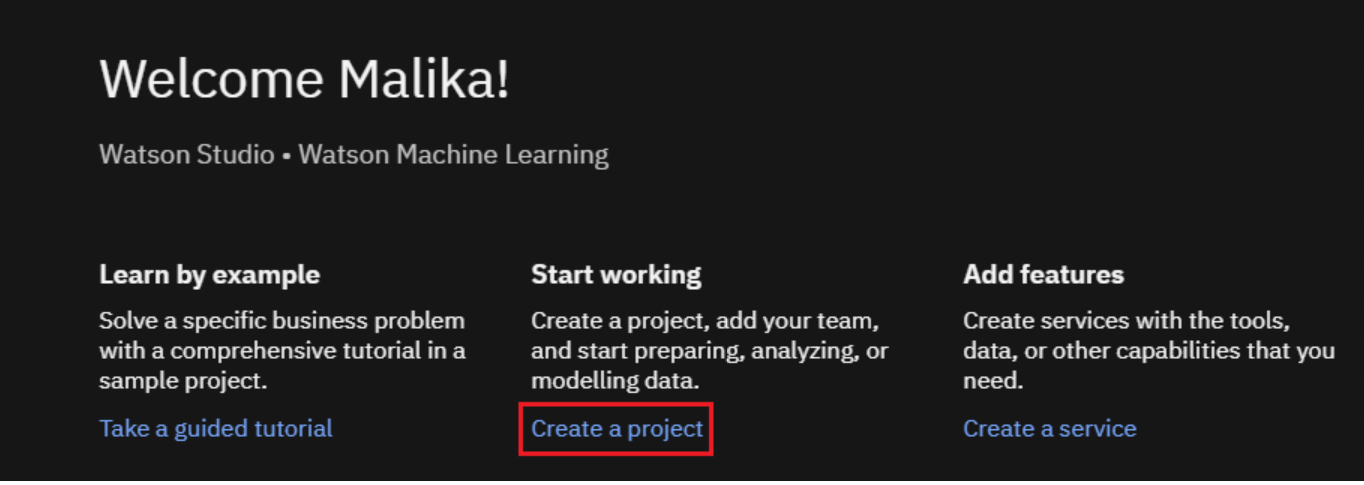
In case you are getting an error like below, please try with an different email address before you contact support. E.g. a non - gmail address.



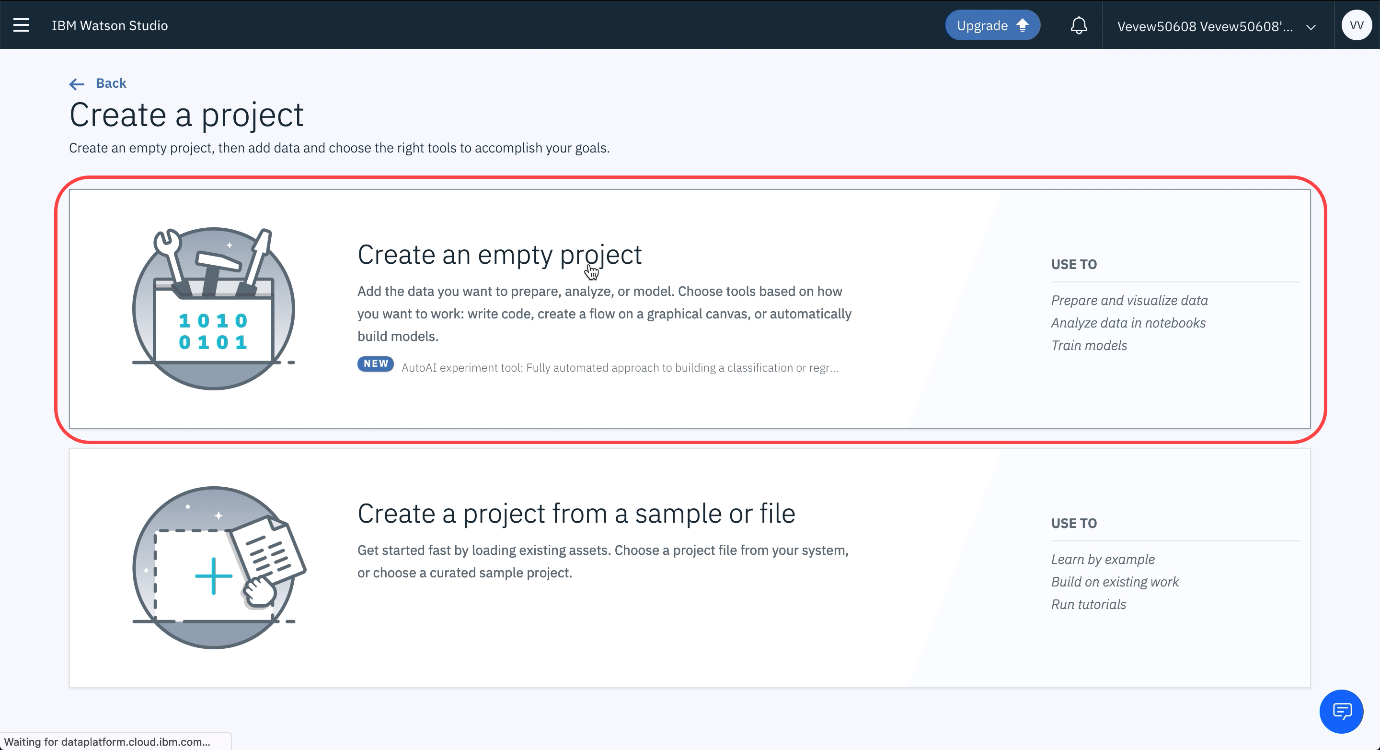
2. Once you've completed registration and confirmed your email address, please open [dataplatform.cloud.ibm.com](https://dataplatform.cloud.ibm.com/) and click **Log In**.



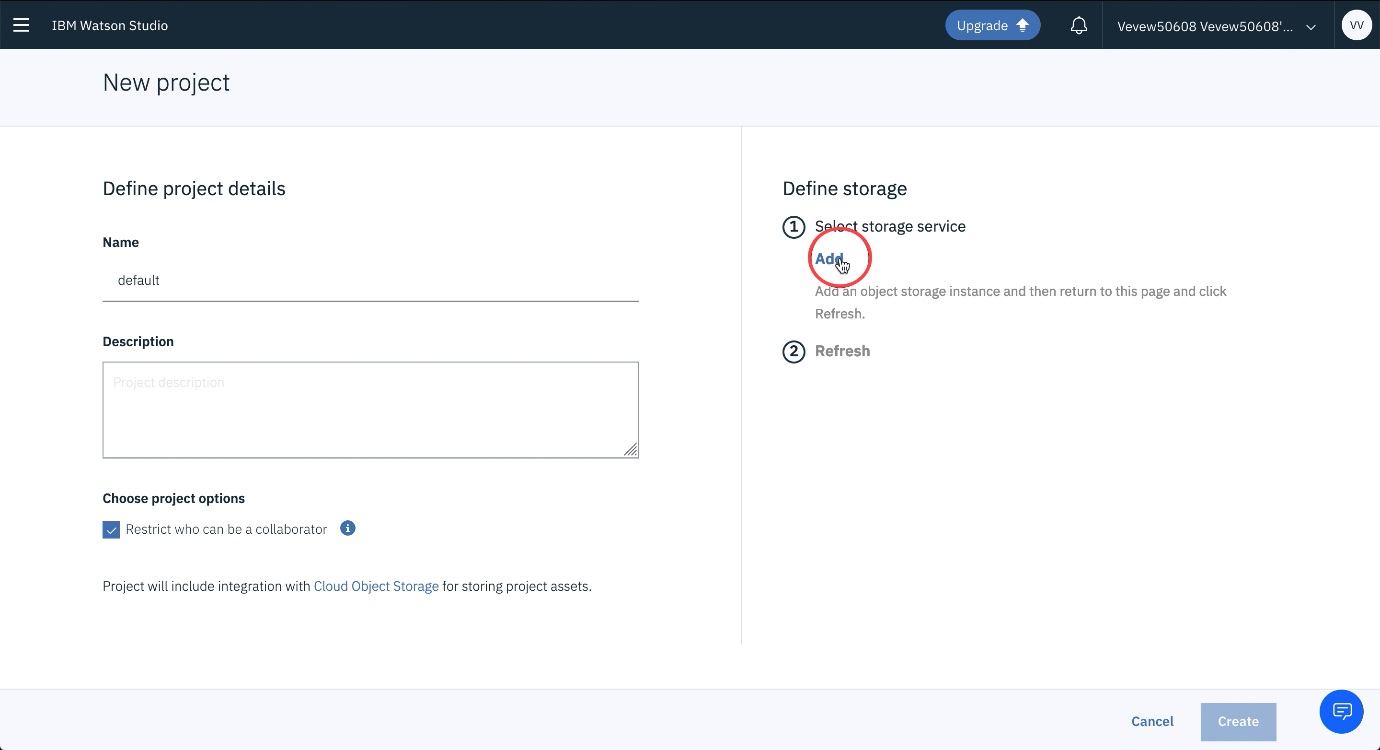
3. Now please click on **Create a project**.



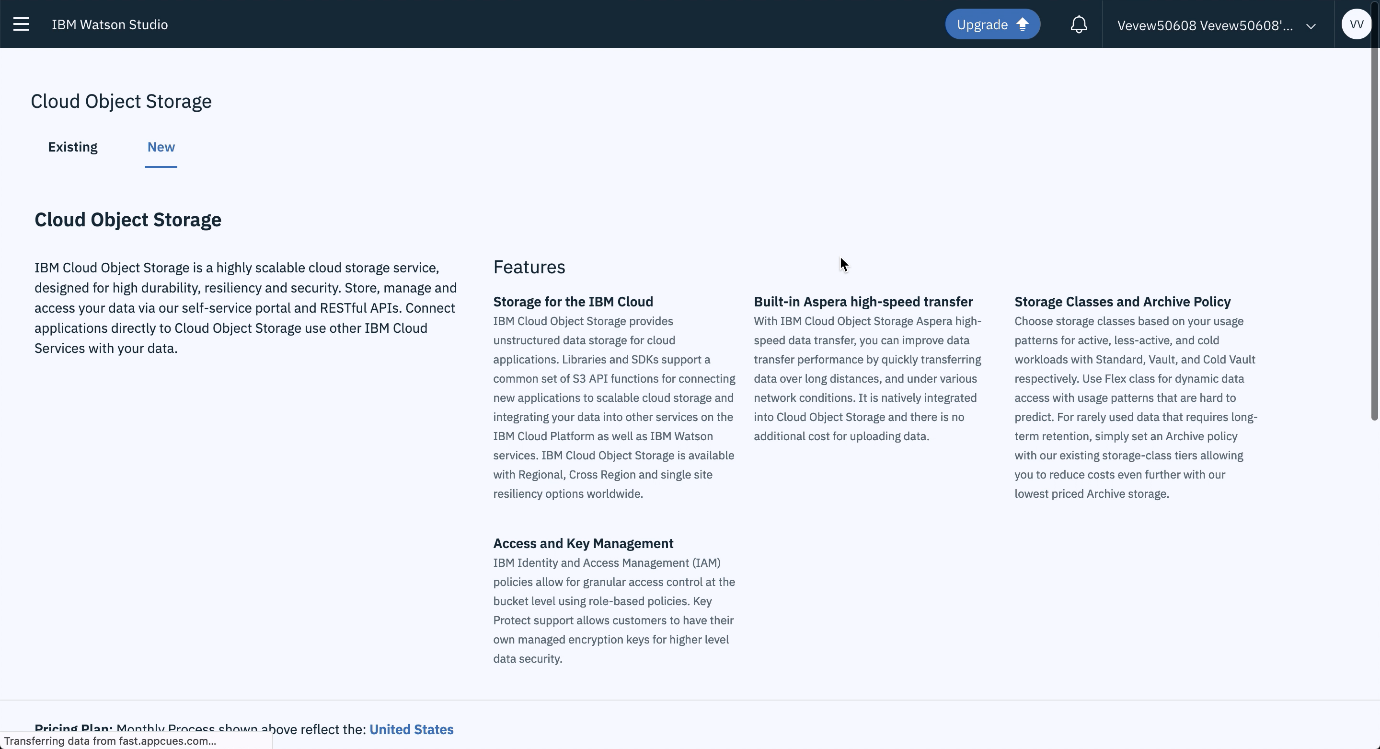
4. Now please click **Create an empty project**.



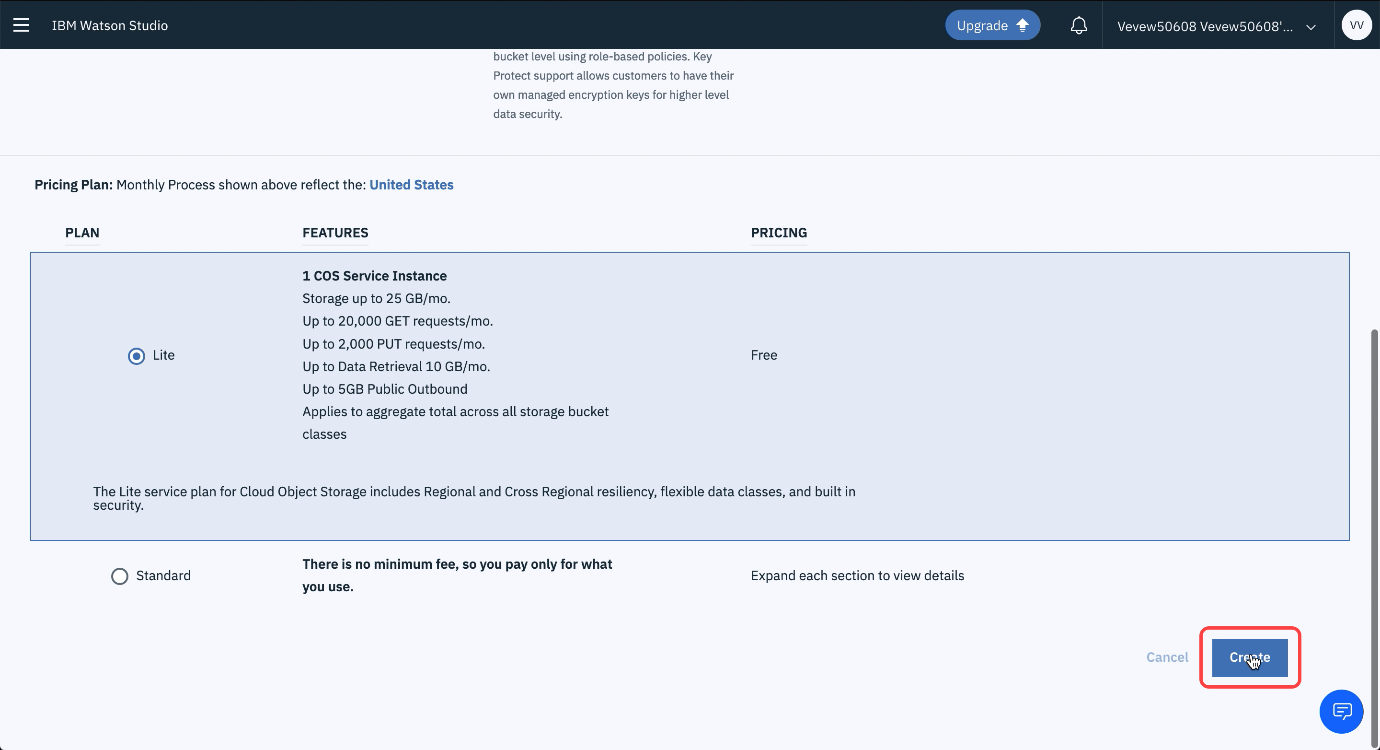
5. Under **Name**, please type "default", then please click on **Add** under **Define Storage**.



6. Once you see the screen below, please scroll down.

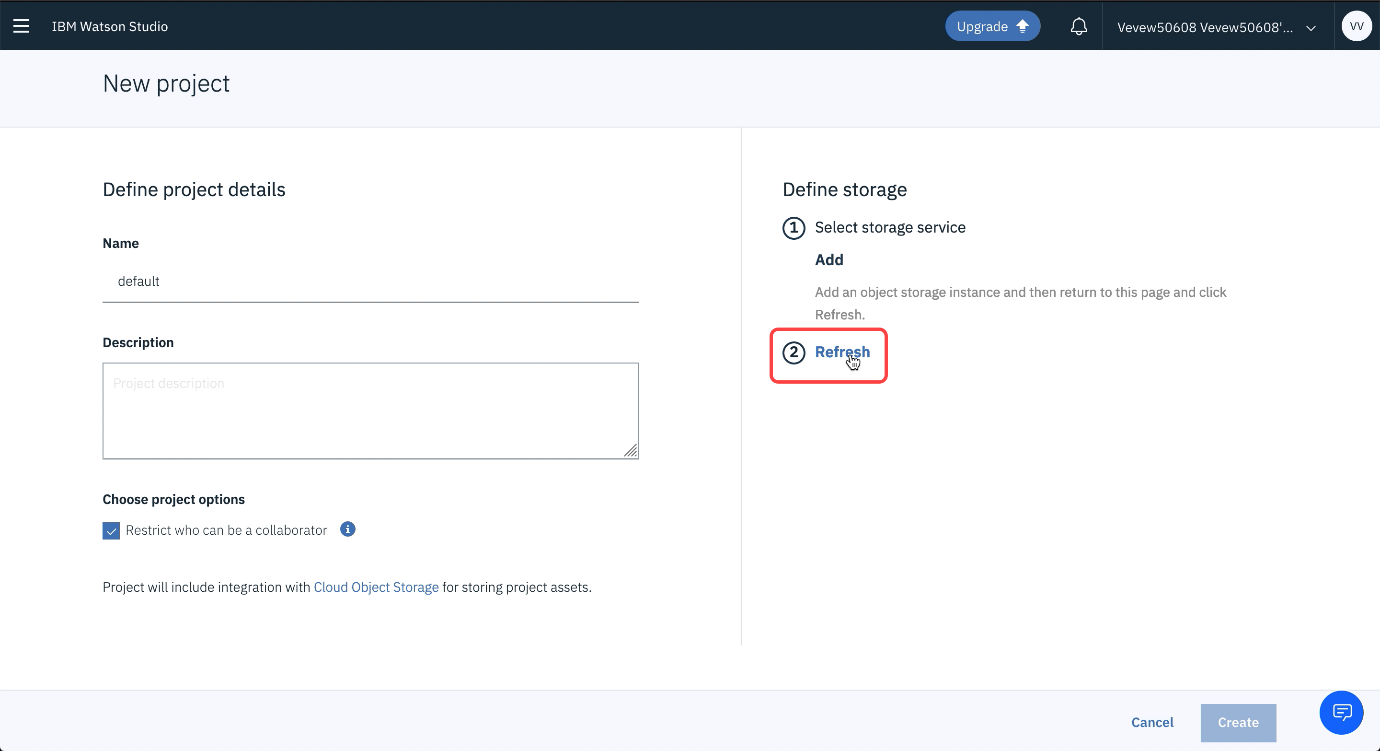


7. Please make sure the **Lite** plan is selected, then please click **Create**.

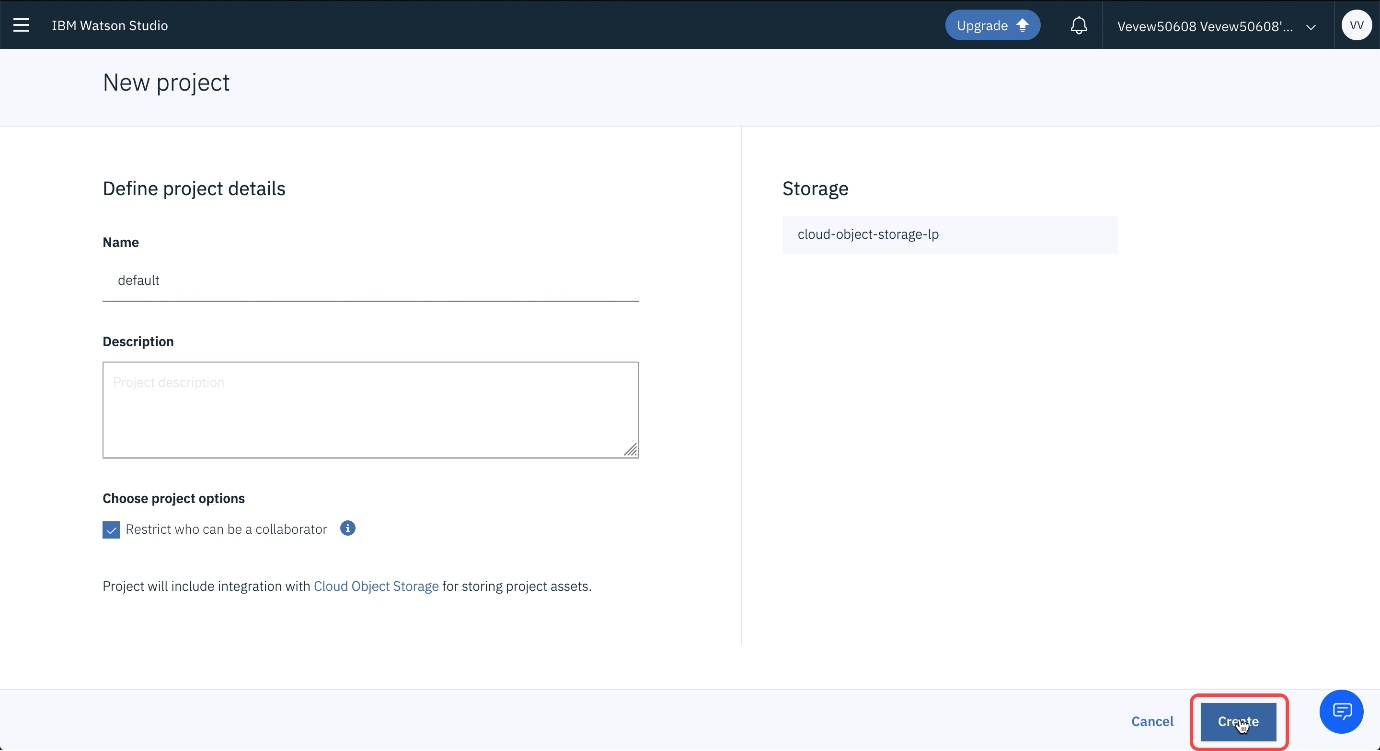


8. Please click **Confirm**.

9. Please click **Refresh**.

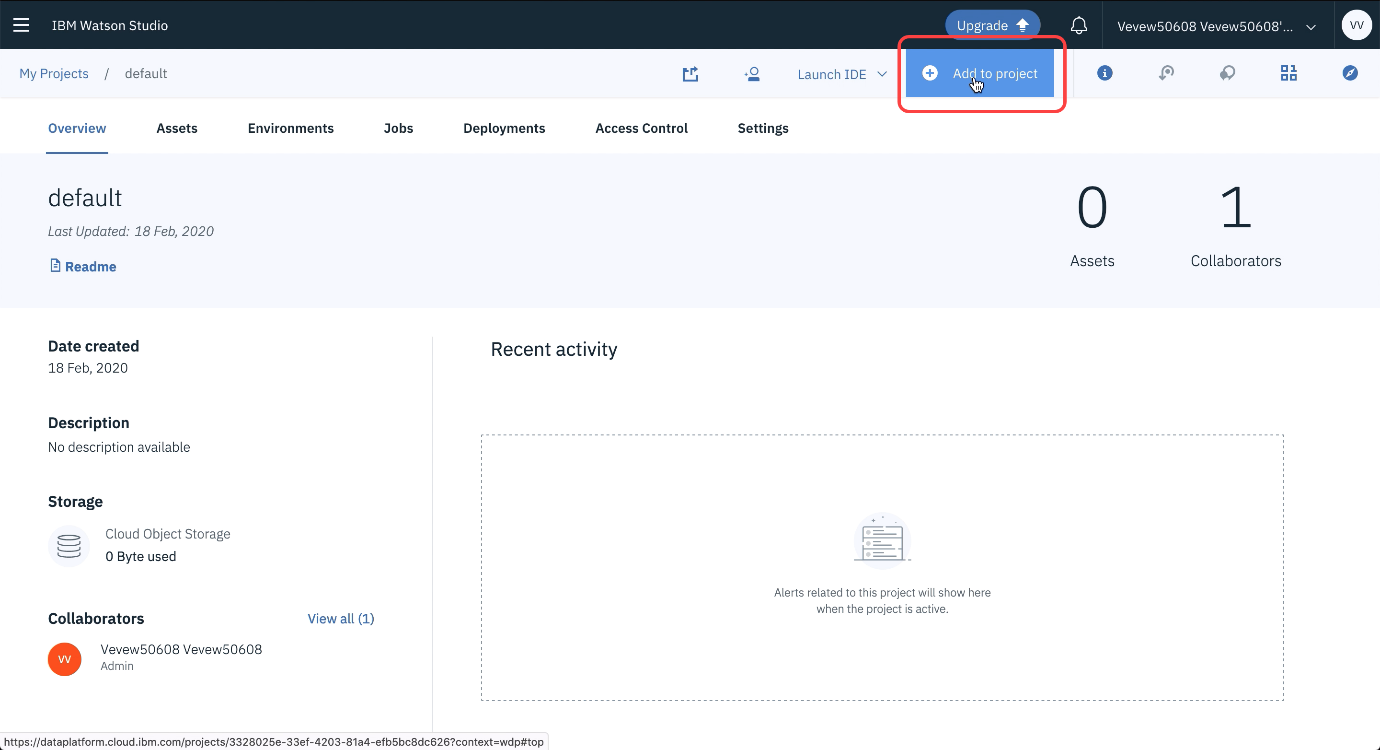


10. Please click **Create** to finalize project creation.

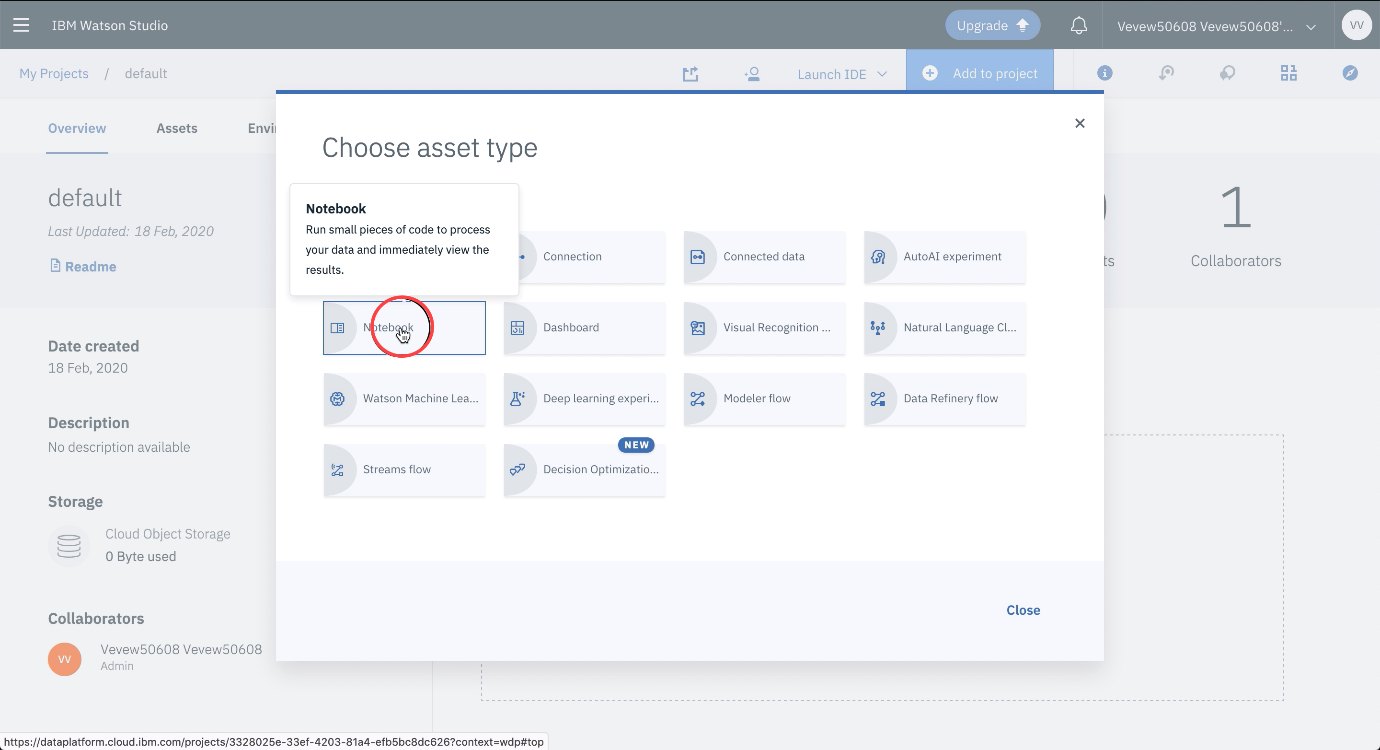


Congratulations, this concludes the first part of the tutorial. Please take a moment to follow through the next steps to learn how you can use Watson Studio Jupyter Notebooks.

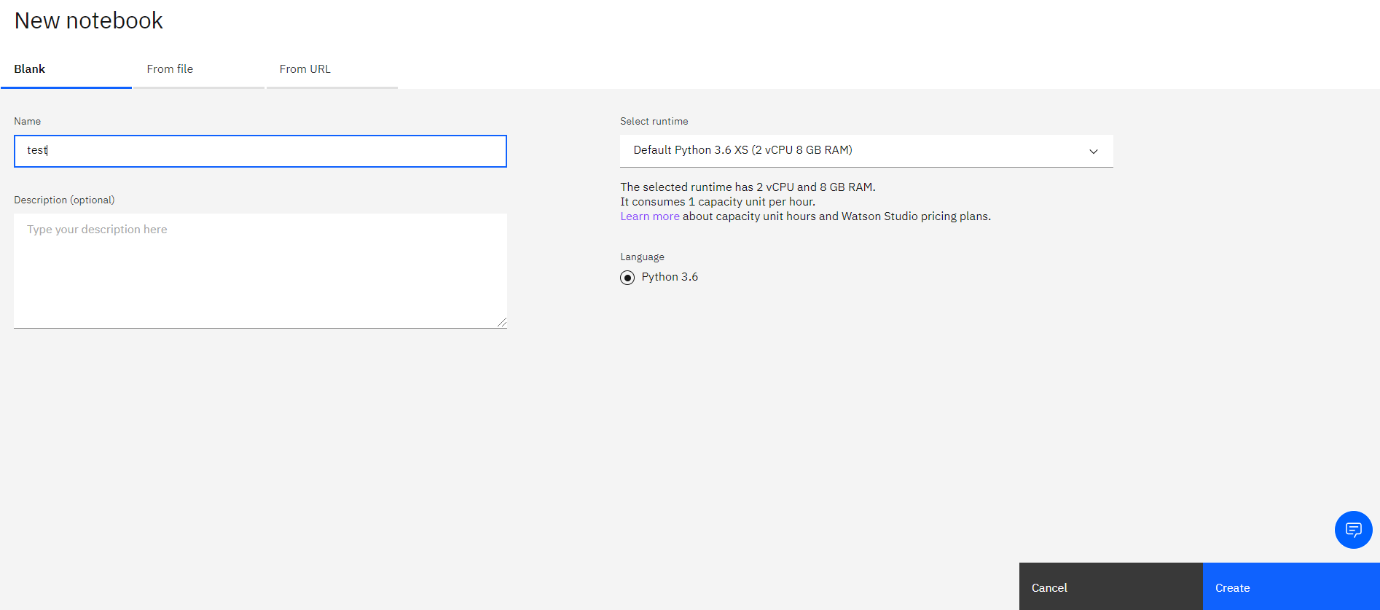
1. Please click on **Add to project**.



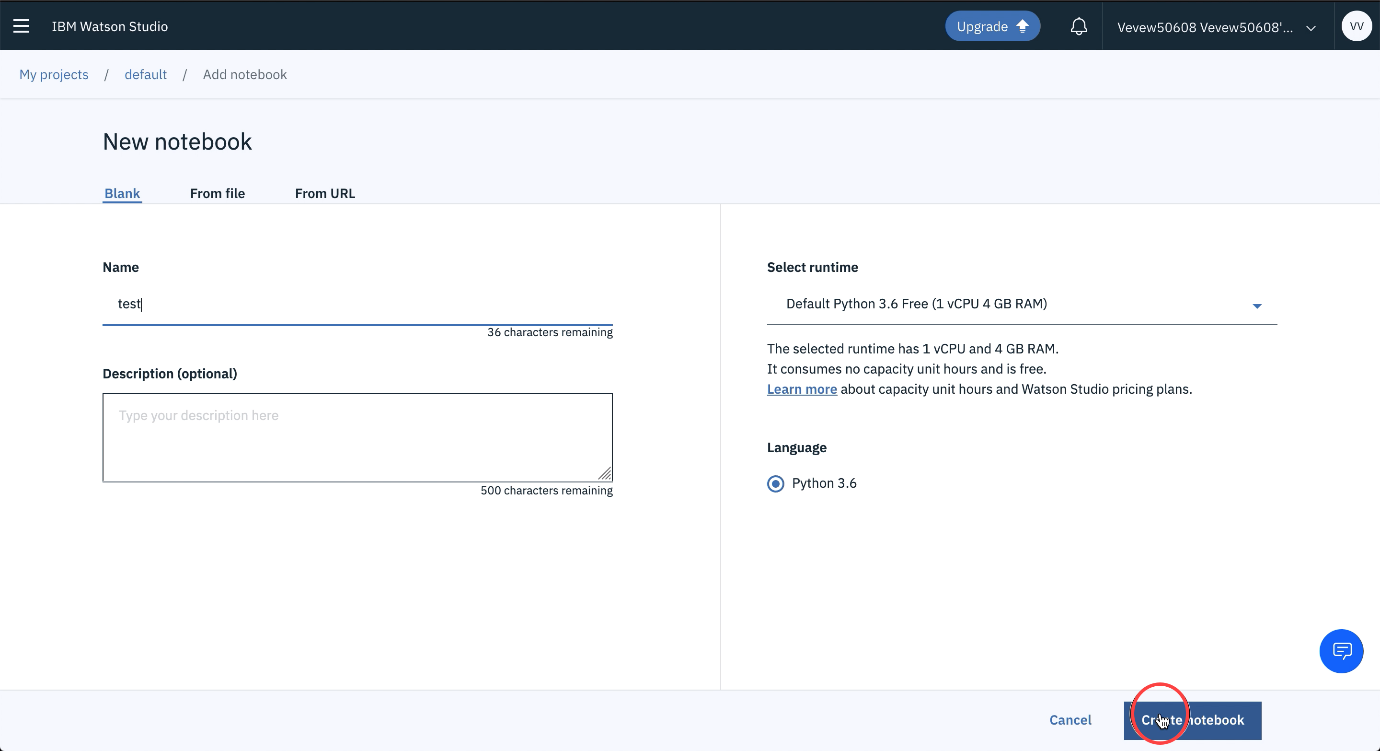
2. Here you can select an abundance of tools, but let's go for Jupyter Notebooks first. Please click **Notebook**.



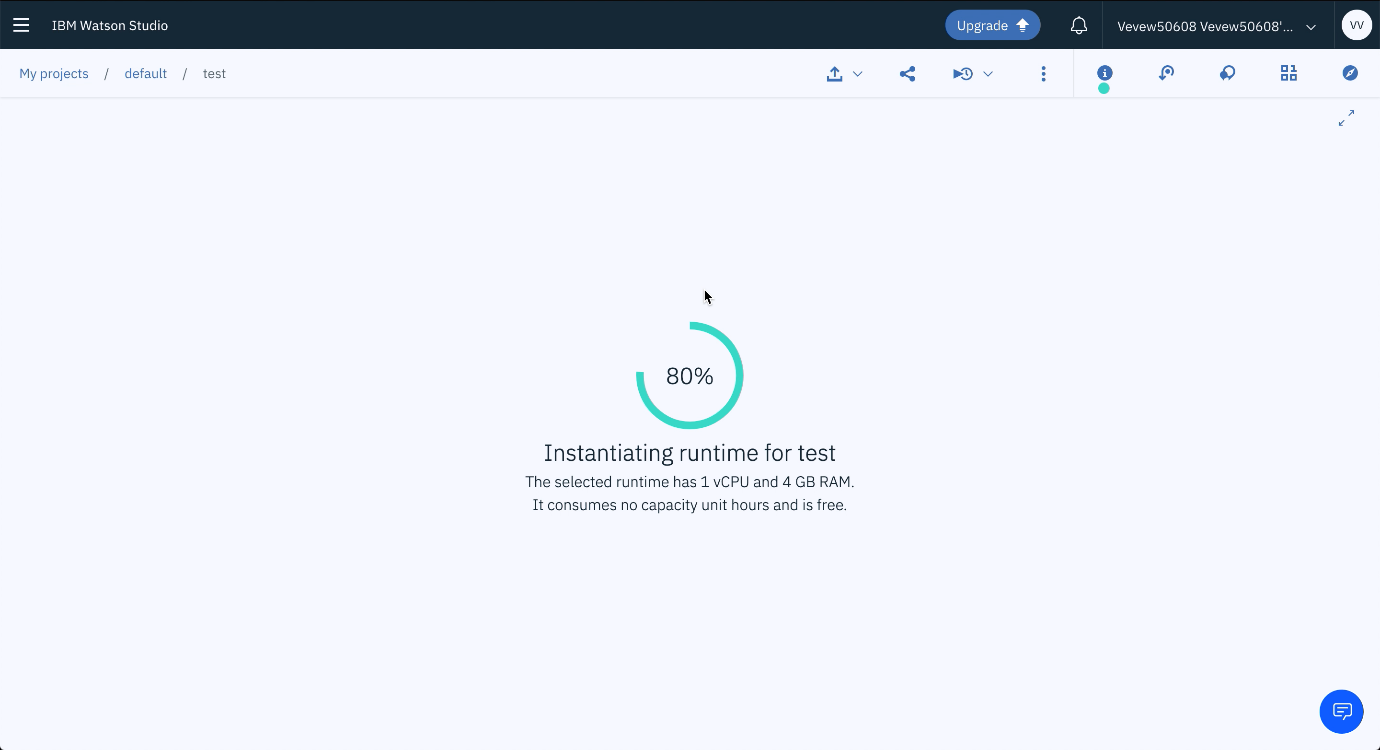
3. In order to not use up your monthly free compute credits just select the **Default Python 3.6 Free** runtime.



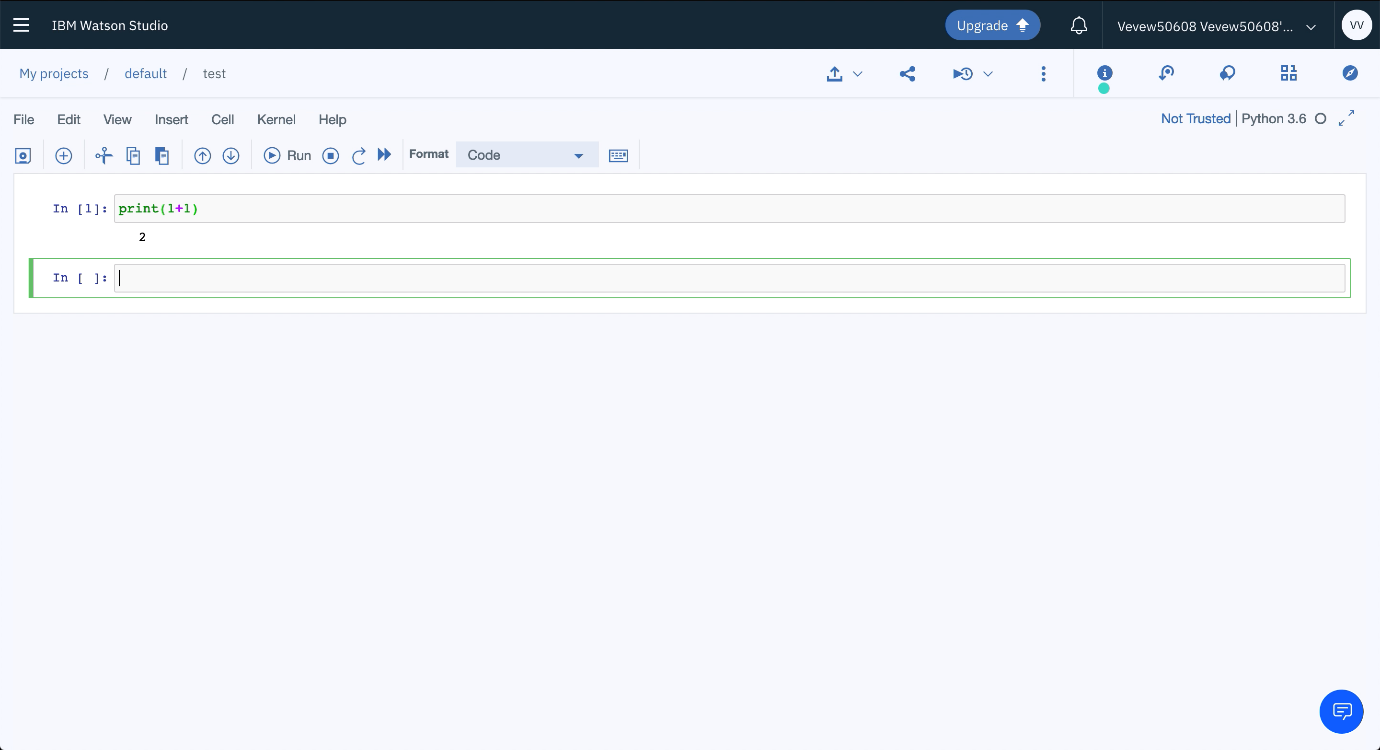
4. Please click **Create notebook**.



5. Just wait until the Notebook appears. In case you are interested. The Jupyter enterprise gateway has requested resources on the Kubernetes cluster IBM hosts for serving the Jupyter kernel backing your Notebook.



6. Now you're ready to code!



This concludes this tutorial.

#### Linking GitHub to Watson Studio

[Music]

this video shows you how to connect your

IBM Watson studio account with your

github account in Watson studio navigate

to your profile settings on the

integrations tab visit the link to

generate a github personal access token

provide a descriptive name for the token

and select the repo scope

then generate the token copy the token

return to the github integration

settings and paste the token the token

is validated when you save it to your

profile settings now navigate to your

projects you enable github integration

at the project level on the settings tab

simply scroll to the bottom and paste

the existing github repository URL once

the URL is validated click connect

go to the assets tab and open the

notebook you want to publish

Play video starting at 1 minute 14 seconds and follow transcript1:14

notice that this notebook has the

credentials replaced with X's it's a

best practice to remove or replace

credentials before publishing to github

so this notebook is ready for publishing

you can provide the target path along

with a commit message you also have the

option to publish content without hidden

code which means that any cells in the

notebook that began with the hidden cell

comment will not be published when

you're ready click publish the message

tells you that the notebook was

published

successfully and provides links to the

notebook the repository in the commit

let's take a look at the commits so

there's the commit and you can navigate

to the repository to see the publish

notebook

lastly you can publish as a gist just

are another way to share your work on

github every gist is a git repository so

it can be forked and cloned there are

two types of gists public and secret if

you start out with a secret gist you can

convert it to a public gist later and

again you have the option to remove

hidden cells

so that's the basics of Watson Studios

github integration

[Music]

#### Practice Quiz - Watson Studio

**TOTAL POINTS 10**

1.

Question 1

Fill in the blank: In Watson Studio, a \_\_\_\_\_\_\_\_\_\_\_\_ is how you organize your resources to achieve a particular goal. Resources can include data, collaborators, and analytic assets like notebooks and models.

1 point



Asset



Notebook



Project



Job

2.

Question 2

Fill in the blank: It's a best practice to remove or replace \_\_\_\_\_\_\_\_\_\_\_\_\_ before publishing to GitHub.

1 point



Credentials



Code cells



Markdown text



Charts

3.

Question 3

Which of the following do you need to create in order to publish a notebook to your GitHub repository?

1 point



Apps



Profile



Access token



Login credential

4.

Question 4

Fill in the blank: If you'd like to schedule a notebook in Watson Studio to run at a different time you can create a(n) \_\_\_\_\_\_\_\_\_\_\_\_\_.

1 point



API



Job



Asset



Markdown cell

5.

Question 5

Fill in the blank: On the environments tab you can define the \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

1 point



Hardware size.



Software configuration.



Runtime configuration for notebook editor.



Runtime configuration for flow editor.



All of the above.

6.

Question 6

Fill in the blank: When sharing a read only version of a notebook, you can choose to share \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

1 point



Only text and output.



All content, excluding sensitive code cells.



All content including code.



A permalink.



All of the above.

7.

Question 7

Fill in the blank: When working in a Jupyter Notebook, before returning to a project, it's important to \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

1 point



Save your notebook.



Insert cells.



Insert to code.



Run cells.

8.

Question 8

Fill in the blank: Before running a notebook, it's a best practice to \_\_\_\_\_\_\_\_\_\_\_\_ to describe what the notebook does.

1 point



Insert a cell at the bottom of the notebook.



Delete notebook cells.



Insert a cell at the top of the notebook.



Refresh your page.

9.

Question 9

Fill in the blank: In the \_\_\_\_\_\_\_\_\_\_\_\_\_ tab you can define the hardware size and software configuration for the runtime associated with Watson Studio tools such as notebooks.

1 point



Settings



Assets



Overview



Environments

10.

Question 10

Fill in the blank: IBM Cloud uses \_\_\_\_\_\_\_\_\_\_\_\_\_\_ as a way for you to organize your account resources in customizable groupings so that you can quickly assign users access to more than one resource at a time.

1 point



Resource groups



Services



Projects



Catalogs

Honor Code Agreement

I, **Dr.SIREESHA NANDURI**, understand that submitting work that isn’t my own may result in permanent failure of this course or deactivation of my Coursera account. Learn more about Coursera’s Honor Code

Other IBM Tools for Data Science

In this video, we will look at

several other IBM tools that help

data scientists in their day to day work.

Watson Knowledge Catalog helps data scientists to

catalog and manage all their data resources.

Data refinery provides graphical tools

for analyzing and preparing data.

SPSS based products include

easy to use graphical interfaces for

wide varieties of

statistical and machine learning algorithms

and data transformations.

We will talk about approaches to model deployment,

including open standards and Watson Machine Learning.

Newer features of Watson Studio include

AutoAI that automatically computes

the best data pipeline and

Watson OpenScale which helps to ensure

fairness and explainability of the models.

#### IBM Watson Knowledge Catalog

Most organizations have huge amounts of data stored in many forms in various locations.

Finding relevant data quickly and connecting disparate data sources can be challenging

and time-consuming. Watson Knowledge Catalog unites all information assets into a single

metadata-rich catalog, based on Watson’s understanding of relationships between assets

and how they’re being used and socialized among users in existing projects. Let’s

have a look at the overview of different tool categories that we’ve previously discussed.

Watson Knowledge Catalog corresponds to the Data Asset Management, Code Asset Management,

Data Management, and Data Integration and Transformation. Watson Knowledge Catalog is

a data catalog that is integrated with an enterprise data governance platform. It also

merges the analytics capabilities of Watson Studio. The data catalog assists data scientists

to easily find, prepare, understand, and use the data as needed. Watson Knowledge Catalog

protects data from misuse and enables the sharing of assets with automated, dynamic

masking of sensitive data elements. Data-profile visualizations, built-in charts and statistics

help users to understand data assets. Seamless integration with Watson Studio helps data

citizens to drive production of their data in a suite of powerful data science, AI, machine-learning

and deep-learning tools. Joining with Watson Studio directs the building, training, and

deploying of models. Users can interactively discover, cleanse, and prepare data with a

built-in data refinery. Possible connections to more than 30 IBM and third-party data sources

help to catalog and use your data in the original locations. IBM Watson Knowledge Catalog has

various deployment choices on IBM Cloud™ and can be run anywhere with IBM Cloud Pak™

for Data. The latter is a fully-integrated data and AI platform built on Red Hat® OpenShift®

Container base. It can be deployed easily into any public or private cloud or other

enterprise platforms. A catalog contains metadata about the contents

of assets and how to access them. And a set of collaborators who need to use the assets

for data analysis. The metadata is stored in an encrypted IBM Cloud object storage instance.

Any data that you want to store in the Cloud, you can upload to the cloud object storage

of your choice, and then specify that object storage when you create the catalog. This

split between where the data's metadata is stored and the actual location of the data

is important. It means that you can keep your data where ever it is. You don't need to move

it into the catalog because the catalog only contains metadata. You can have the data in

unpremises data repositories in other IBM cloud services like Cloudant or Db2 on Cloud

and in non-IBM cloud services like Amazon or Azure, in streaming data services or even

dark data sources like PDFs. Included in the metadata is how to access the data asset.

In other words, the location and credentials. That means that anyone who is a member of

the catalog and has sufficient permissions can get to the data without knowing the credentials

or having to create their own connection to the data. Since the new catalog is empty,

let's take a look at an existing catalog. On the Browse Assets tab you can see "recommendations",

"highly rated assets", and "recently created assets", as well as a list of all the assets.

You can type a search term to find assets, and you can filter by asset type, such as

Data Asset or Notebook. Or filter by tags that were assigned to the asset when it was

added to the catalog. When you view an asset, you get a preview of the data and other information

like a description, ratings, tags, where the source is located, and any classifications.

On the Access tab, those with permission can add members to view this particular asset.

And the Review tab shows reviews and lets you contribute a review. When assets are added

to a catalog with Data Policies enabled, Watson Knowledge Catalog automatically profiles and

classifies the content of the asset based on the values in those columns. The Profile

tab contains more detailed information about the inferred classifications. You can see

the other possibilities for classifying each column and the confidence scores for those

other possibilities. On the Lineage tab, you'll see the various events that Watson Knowledge

Catalog has captured that occurred in the lifecycle of this data asset, allowing you

to trace what's happened to the asset since it was created. On the Access Control tab,

you can see the current list of catalog members. you can also add members which is pretty similar

to adding collaborators in a project. Most catalog members will likely have the editor

role. The viewer role is intentionally restricted and only a select few users will have the

admin role. Watson Knowledge Catalog includes capabilities

to automatically mask sensitive data according to your organization's governance policies.

For example, you can see in the diagram that the first name, last name, and gender data

in the data set have been masked. You’ve learned how IBM Watson Knowledge Catalog can

help organizations deal with their numerous data and other assets. In the next video we’ll

look at Data Refinery, a powerful tool for analyzing and preparing data.

#### Data Refinery

Hi, I'm Sonali Surange Dev.

Play video starting at 11 seconds and follow transcript0:11

Data scientists often end up spending a lot

of time doing mundane tasks like cleansing, shaping and preparing data.

Typically these tasks are roadblocks for starting the more enjoyable part of

analyzing the data sets or building and training machine learning models.

Play video starting at 28 seconds and follow transcript0:28

This is because data sets typically are not in a format that can be readily used.

Play video starting at 34 seconds and follow transcript0:34

They first need to be cleansed, refined before they are useable by a data scientist.

Play video starting at 40 seconds and follow transcript0:40

IBM Data Refinery addresses this issue and simplifies the task of refining data and

its workflows.

It provides a self-service data preparation environment where you

can quickly analyze, cleanse and prepare data sets.

Data refinery is available with

Watson Studio on public cloud, private cloud and desktop.

In the rest of the

video we will walk through a scenario and see Data Refinery in action.

In this

scenario we will use Data Refinery to find the best deals

using data about discounts offered over time.

We will then automate the

analysis to run on a regular schedule.

Before the Data Scientist starts, she looks at the data

distribution and notices that the inSale column is missing data.

She visualizes the offer column and notices that it contains valuable

information about discounts.

Many fields contain the percent of information,

some contain references to previous price indicating a new reduced price

being available.

She decides to derive sale from offer.

She uses a conditional decrease operation to derive if the product is on

sale.

Next she uses a filter operation

to eliminate deals that are not on sale

She then wants to pick up the bargains.

She uses the replace substring operation

and provides a pattern that extracts the discounts from the offer.

After

converting the discount values to a decimal

she can visually see the discounts that were available.

She needs to find the

months that offered the best deals.

She visualizes the dateUpdated and notices

that the date field has a variety of formats, some with dashes some with

slashes and some with months as text.

She hopes that Data Refinery can normalize the

data and extract a month.

She uses the convert column operation to convert to

date and selects ymd.

Next she extracts month and creates a derived column

called discountMonth.

The data now represents all brands and products

providing sales and the month the offer was available.

The data scientist is only

interested in her preferred brands.

Over time she has built a list of preferred

brands and has imported the data in her project.

Data Refinery provides

relational transformations such as left, inner right, full, semi and anti-join.

To ensure

that the data only contains her preferred brand she uses a semi-join

operation which narrows the brands to match her preferences.

She then selects the keys for the join and the resulting fields.

The visual results now confirms that the brands match the preferences.

To find the best

possible deals she needs to perform some aggregations.

Several features determine a good deal.

She is interested in the best offer and

duration when the discounts are active.

Aggregating the sale data will help

understand the deals.

She groups the columns by brand and discountMonth and

calculates the maximum discount.

Finally she sorts the result in descending order

Data refinery is now displaying the best deals by brand preferences and the

duration which the offer is available.

The last step is to execute the analysis on the full dataset.

She starts the full

analysis, which she can monitor for the completion status.

It's time to automate the analysis which runs on a regular basis.

The data in the

database can grow over time.

She uses a personalized runtime to match the larger

data volumes and sets a schedule for automation.

The hourly schedule reads

from updated data from the database and writes to the target table.

Data Refinery has helped her uncover deals in

the raw data through a small set of

operations and transformations with the bulk of the work done for her.

Thank you

for watching

#### Practice Quiz - Other IBM Tools

**TOTAL POINTS 12**

1.

Question 1

Which products (of those we covered) allow you to build data pipelines using graphical user interface and no coding?

1 point



Only IBM SPSS Statistics.



Only IBM SPSS Modeler.



OpenScale



IBM SPSS Modeler and Modeler Flows in Watson Studio.



All of the above.

2.

Question 2

Which features of Data Refinery help save hours and days of data preparation?

1 point



Flexibility of using Intuitive user interface and coding templates enabled with powerful operations to shape and clean data.



Data visualization and profiles to spot the difference and guide data preparation steps.



Incremental snapshots of the results allowing the user to gauge success with each iterative change.



Saving, editing and fixing the steps provide ability to iteratively fix the steps in the flow.



All of the above.

3.

Question 3

Watson Knowledge Catalog provides what functionality?

1 point



Catalog all books mentioning Doctor Watson and Sherlock Holmes.



Process data, build and deploy models.



Catalog data and ML assets, help to find relevant assets, keep track of asset lineage, enforce data governance.



Build data and water pipelines.



Create data and deploy models into production.

4.

Question 4

Fill in the blank: PMML, PFA, and ONNX are \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

1 point



Robots that are plotting to take over the planet.



Codes for getting rid of undesired data or models.



Passwords for some super-secret system.



Abbreviations for machine learning algorithm names.



Open standards for predictive model serialization, exchange, and deployment.

5.

Question 5

Which node must be used in Modeler flows before any modeling node?

1 point



Output node



Type node



Derive node



Auto Numeric node



All of the above

6.

Question 6

Fill in the blank: Auto Classification node can be used for data with \_\_\_\_\_\_\_\_\_\_\_\_\_\_.

1 point



No target variables.



A categorical target variable.



A continuous target variable.



Any target variable.



All of the above

7.

Question 7

Fill in the blank: Auto Numeric node can be used for data with \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

1 point



No target variables.



A categorical target variable.



A continuous target variable.



Any target variable.



All of the above.

8.

Question 8

IBM SPSS Modeler evolved from which product?

1 point



Netezza



Oracle



SPSS



IBM DB2



Clementine

9.

Question 9

Fill in the blank: IBM SPSS Statistics syntax can be created using \_\_\_\_\_\_\_\_\_\_\_.

1 point



AutoAI



OpenScale



IBM SPSS Modeler streams.



Watson Studio Modeler flows.



Graphical user interface of IBM SPSS Statistics product or syntax editor.

10.

Question 10

AutoAI provides which of the following services?

1 point



Monitoring for fairness, bias, and model drift.



Automatic finding of optimal data preparation steps, model selection, and hyperparameter optimization.



Cataloging data and model assets.



Creating SPSS syntax.



All of the above.

11.

Question 11

OpenScale provides which of the following services?

1 point



Creating SPSS syntax.



Automatic finding of optimal data preparation steps, model selection, and hyperparameter optimization.



Cataloging data and model assets.



Monitoring for fairness, bias, and model drift.



All of the above.

12.

Question 12

Predictive Model Markup Language (PMML) was created by which entity?

1 point



SPSS



The Data Mining Group



Oracle



Microsoft



IBM

Honor Code Agreement

I, **Dr.SIREESHA NANDURI**, understand that submitting work that isn’t my own may result in permanent failure of this course or deactivation of my Coursera account. Learn more about Coursera’s Honor Code

#### Graded Quiz

**TOTAL POINTS 10**

1.

Question 1

Which feature in Watson Studio helps to keep track of and discover relevant Machine Learning assets?

1 point



OpenScale



AutoAI



Modeler Flows



Watson Knowledge Catalog



All of the above

2.

Question 2

Data Refinery provides which of the following services?

1 point



Catalog the data assets.



Monitor for bias and model drift.



Visualize and prepare data.



Automatically build models.



All of the above.

3.

Question 3

How does Data Refinery help build repeatable Data Pipelines for workloads of almost any size?

1 point



Only a fixed workload size is supported.



Create a scheduled Job and use a custom environment to run the data flow/pipeline on different workloads.



Not supported.



Manually write APIs to provide automation.



Feature is available only in the UI, not API.

4.

Question 4

Modeler flows in Watson Studio always begin with which type of node?

1 point



A modeling node



A type node



A data source node



An output node



All of the above

5.

Question 5

IBM SPSS Modeler includes what kind of models?

1 point



Classification models (for data with a categorical target).



Regression models (for data with a continuous target).



Clustering models (for data with no target variables).



Other kinds of models.



All of the above.

6.

Question 6

What feature of IBM SPSS Statistics allows easy saving and modifying of previous tasks?

1 point



Charts



Graphical user interface



SPSS syntax



SPSS Modeler streams



All of the above

7.

Question 7

Open Neural Network eXchange (ONNX) was originally created for what models?

1 point



Deep learning models.



Clustering models



Decision trees



Support Vector Machines (SVM).



Regression models

8.

Question 8

Fill in the blank: If you'd like to schedule a notebook in Watson Studio to run at a different time, you can create a(n) \_\_\_\_\_\_\_\_.

1 point



markdown cell



job



API



asset

9.

Question 9

Fill in the blank: In the \_\_\_\_\_\_\_\_\_\_ tab you can define the hardware size and software configuration for the runtime associated with Watson Studio tools such as Notebook.

1 point



settings



assets



overview



environments

10.

Question 10

Fill in the blank: It's a best practice to remove or replace \_\_\_\_\_\_\_\_\_\_\_\_\_ before publishing to GitHub.

1 point



code cells



markdown text



charts



credentials

Honor Code Agreement

I, **Dr.SIREESHA NANDURI**, understand that submitting work that isn’t my own may result in permanent failure of this course or deactivation of my Coursera account. Learn more about Coursera’s Honor Code