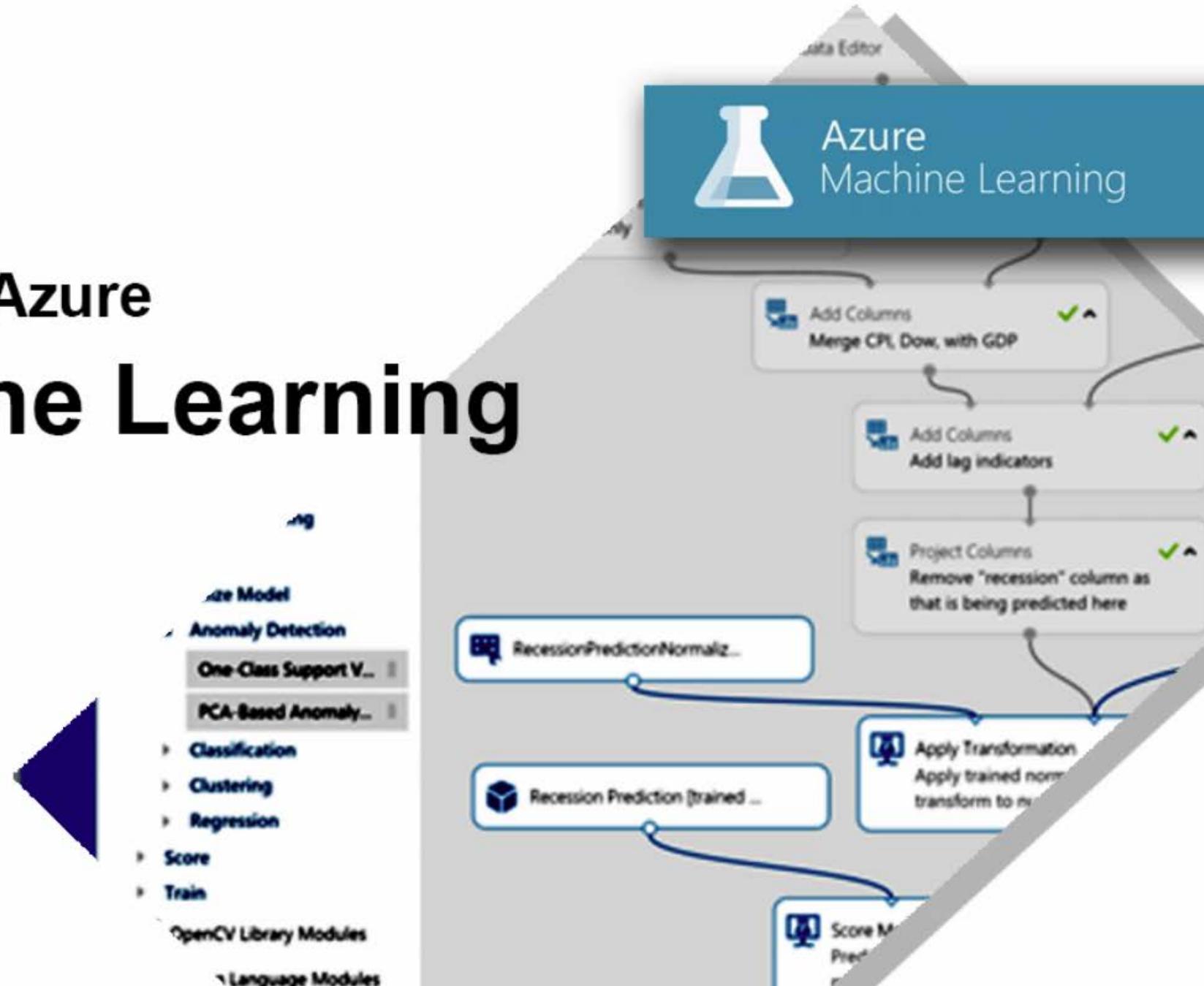
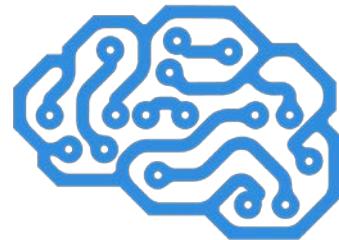


Essential Microsoft Azure Machine Learning



INTRODUCTION TO COURSE



INSTRUCTOR INFORMATION

Loy Vanich
084 007 5544
Line ID: laployv
laploy@gmail.com
www.laploy.com



TIME FRAME

- Course duration
 - Start time
 - Coffee break
 - Lunch break
 - Course end

REPOSITORY AND NOTE SHARE

Repository

github.com/laploy/ML

Note Share

gist.github.com/laploy

(Azure ML Note for student)

Hand out Links and text

notebooks.azure.com/laploy/libraries/loym1

SOFTWARE REQUIREMENT

- Windows 10
- Visual Studio 2017
- R Interpreter
 - R Studio
 - Anaconda
 - Pycharm
- Microsoft Azure subscription
- Microsoft Azure Machine Learning subscription
 - Microsoft Power BI subscription

COURSE OUTLINE

Day 1: ML Basic

- How to: ML (Machine Learning)
- How to: DS (Data Science)
- How to: FE (Feature Engineering)
- Practice: Handle Missing Value (HMV) in Azure ML
- Create: first ML
- Create: ML in C#

Day 2: R Script

- How to: R and DS
- How to: FE in R
- Practice: HMV in R
- Make: Integration to Azure ML

Day 3: Anaconda

- How to: Anaconda and DS
- How to: FE in Anaconda
- Practice: HMV in Anaconda
- Make: Integration to Azure ML

Day 4: ML workshop

- Workshop: DS of data ingestion
- Workshop: Liner regression
- Workshop: train/evaluate ML model
- Workshop: batch execution in C#

Day 5: ML + BI

- How to: Business Intelligence basic
- How to: ML.BI integration (Power BI)
- Create: ML.BI job
- Publish: ML.BI on web

Course scope

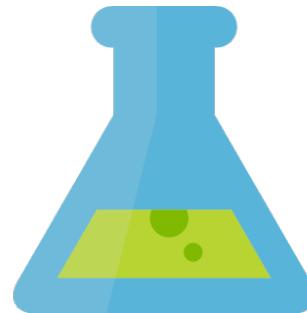
In scope

- Introductory of Machine learning
- Introductory of data science
- Introductory of R and Python for DS
- Basic ML programming

Out of scope

- Advance Machine Learning
- Advance R or Python programming
- Deep learning
- Computer vision
- Complete data science training

INTRODUCTION TO AZURE ML



In this session

- What Bill have to say?
- What is Machine Learning?
- What is Azure ML?
- What is Azure ML Studio?
- Artificial Intelligence VS AI
- ML paradigm
- Everyday examples of predictive analytics
- Machine Learning work flow
- Machine learning language
- Cortana Intelligence Suite (CIS)
- Google Machine Learning
- Amazon Machine Learning

What Bill have to say?

“A breakthrough in machine learning would be worth ten Microsofts,”

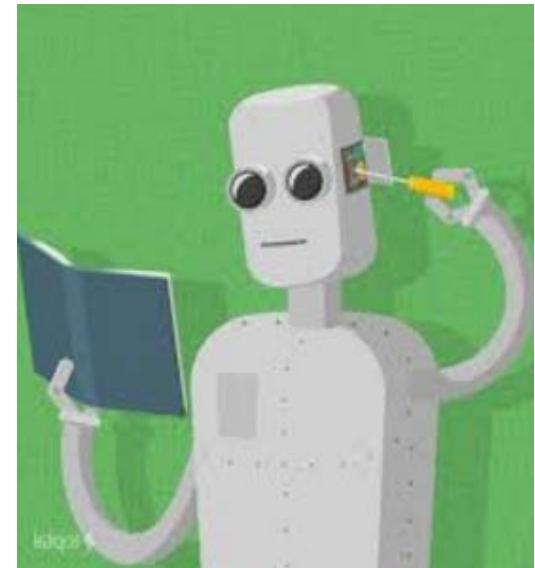
Bill Gates



What is Machine Learning?

What is Machine Learning?

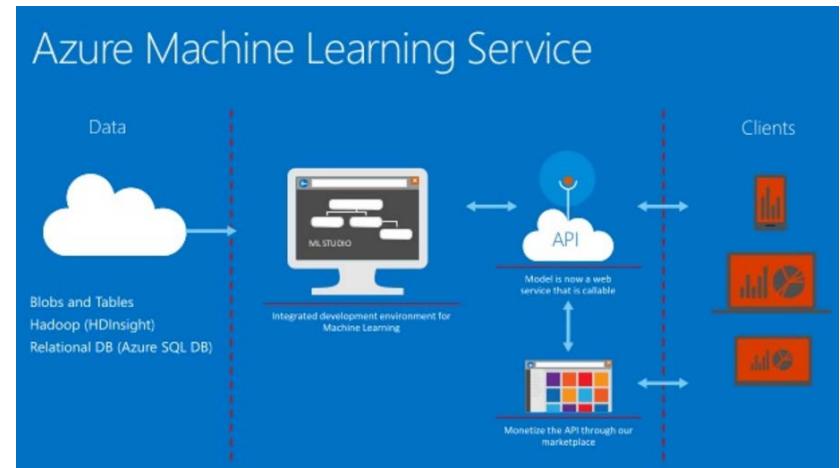
- Subfield of computer science
- Ability to learn without being explicitly programmed
- Algorithms that can learn from and make predictions on data
- Closely related to computational statistics
- Focuses on prediction-making
- Sometimes mixed with Data Mining (DM)
- Used complex models and algorithms
- Predictive analytics



What is Azure ML?

What is Azure ML?

- Pronounce = Air-Cher
- Part of Azure Could platform
- Part of Cortana Intelligence Suite
- Platform as a service (PaaS)
- Create systems that improve with experience
- Help turning data into software
- Help training with huge volumes of data
- Used to predict certain patterns, trends, and outcomes
- Predictive analytics is the underlying technology
- Use the past to predict the future



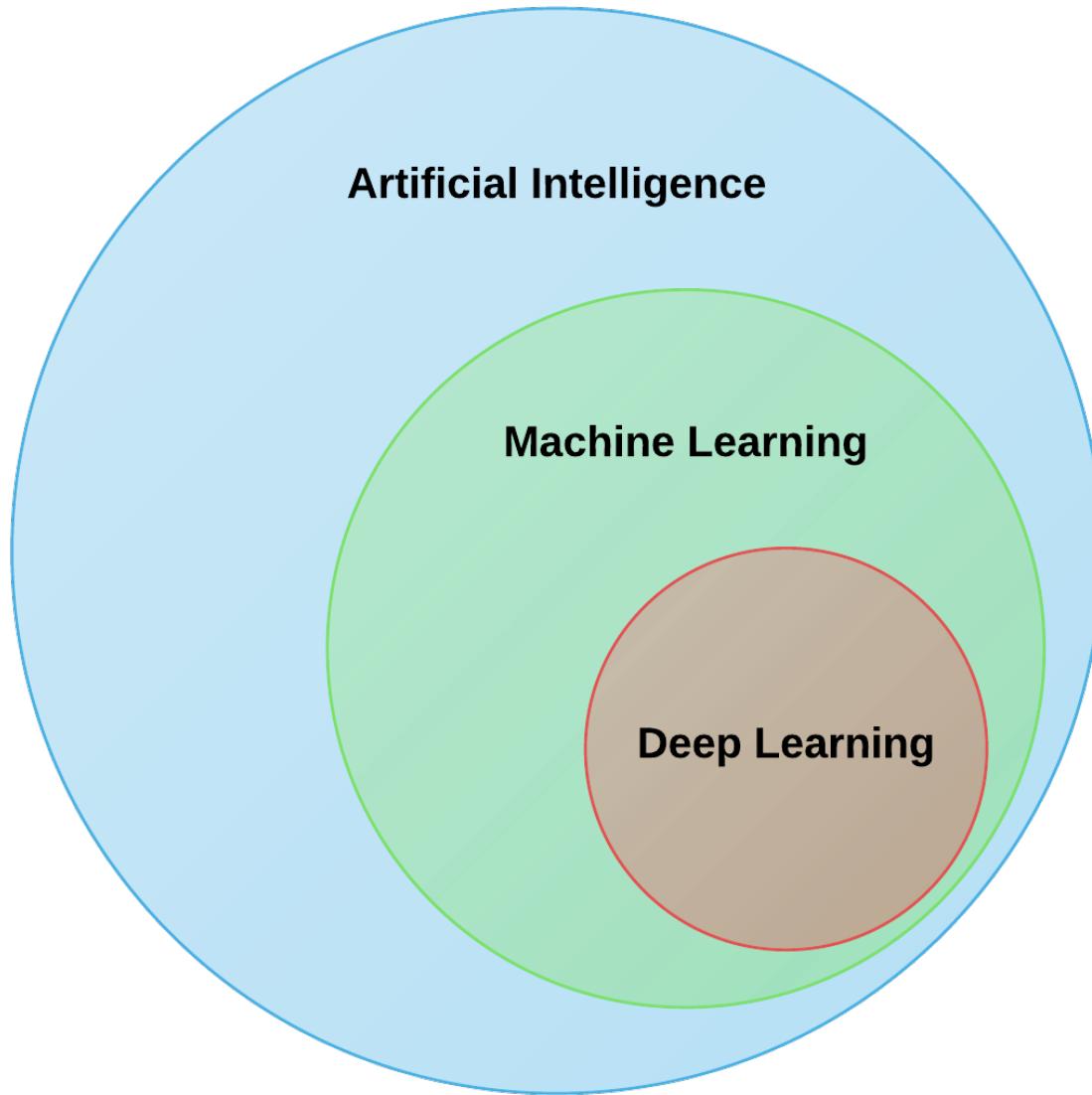
What is Azure ML Studio?

Azure Machine Learning Studio

What is Azure ML Studio?

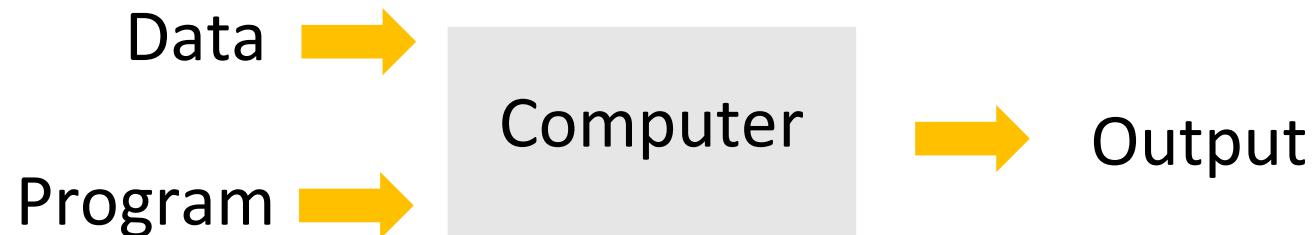
- Is a collaborative tool
- Drag-and-drop
- Use to build, test, and deploy predictive analytics solutions
- Publishes models as web services
- Easily be consumed by custom apps or BI tools
- Is where data science, predictive analytics, cloud resources, and your data meet.

Artificial Intelligence VS AI

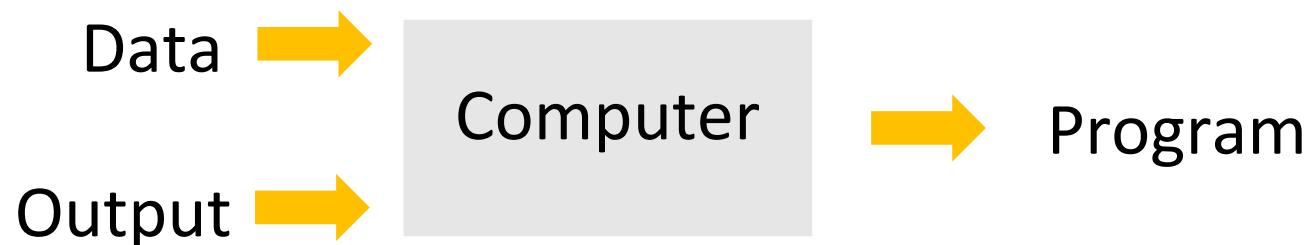


ML paradigm

Traditional Programming



Machine Learning



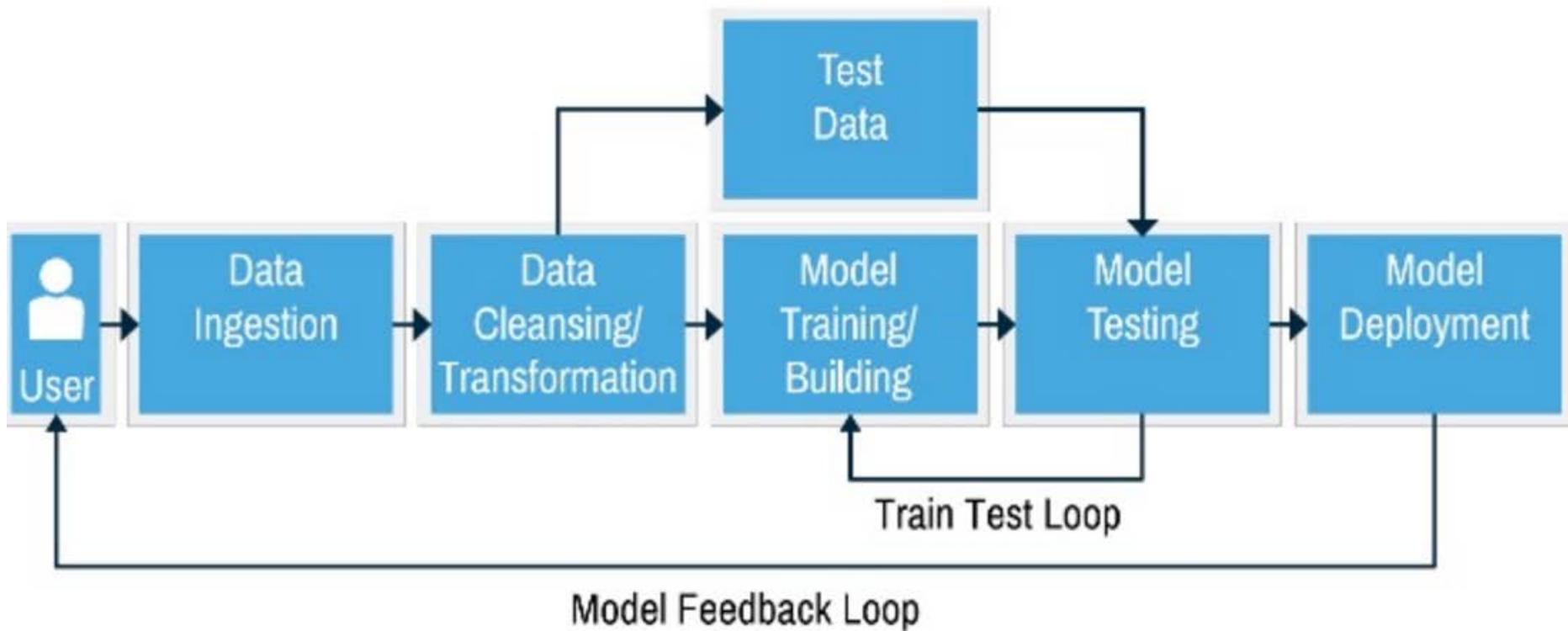
Everyday examples of predictive analytics

Everyday examples of predictive analytics

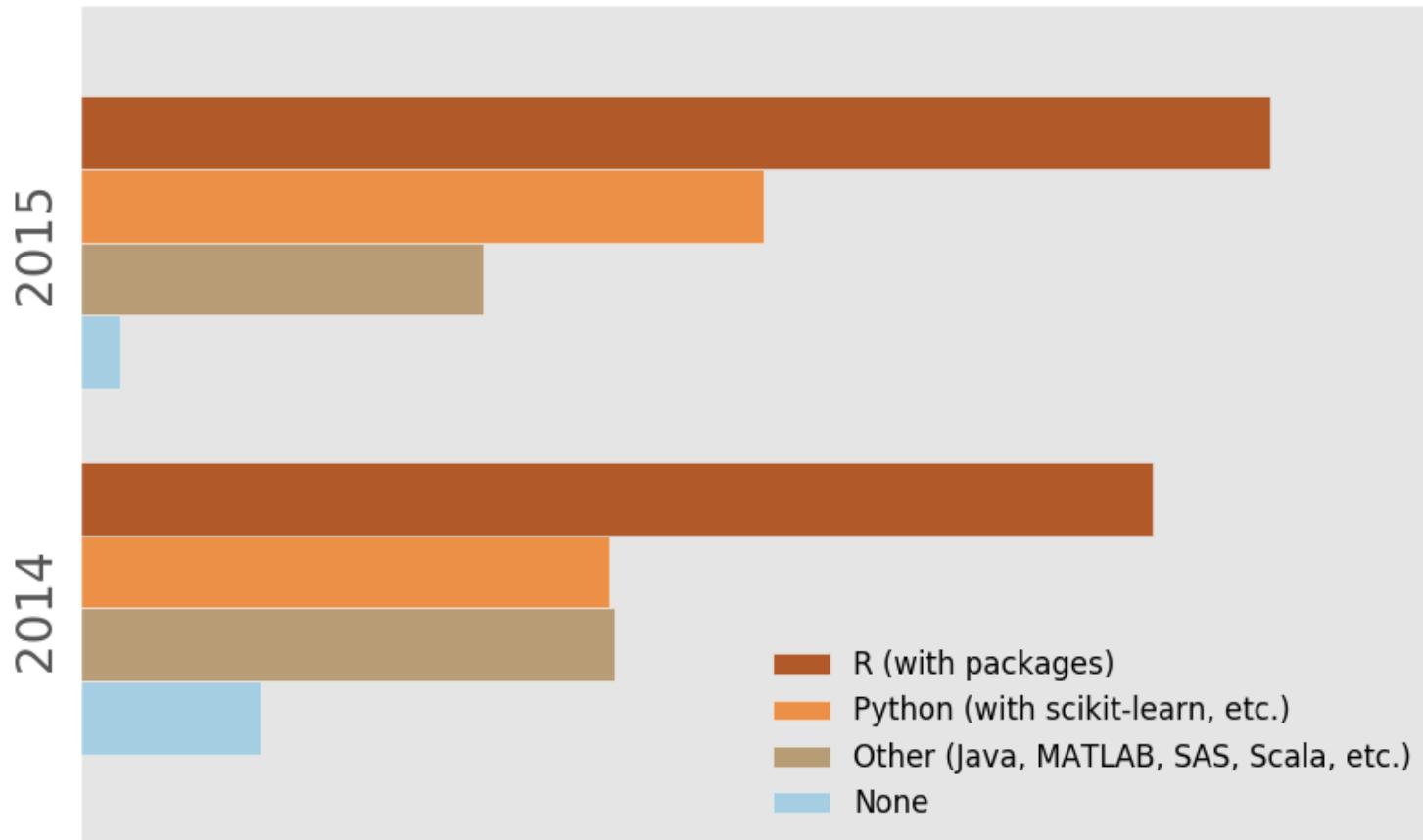
- Spam/junk email filters
- Mortgage applications
- Pattern recognition
- Life insurance
- Medical insurance
- Liability/property insurance
- Credit card fraud detection
- Airline flights
- Web search
- Predictive maintenance
- Health care



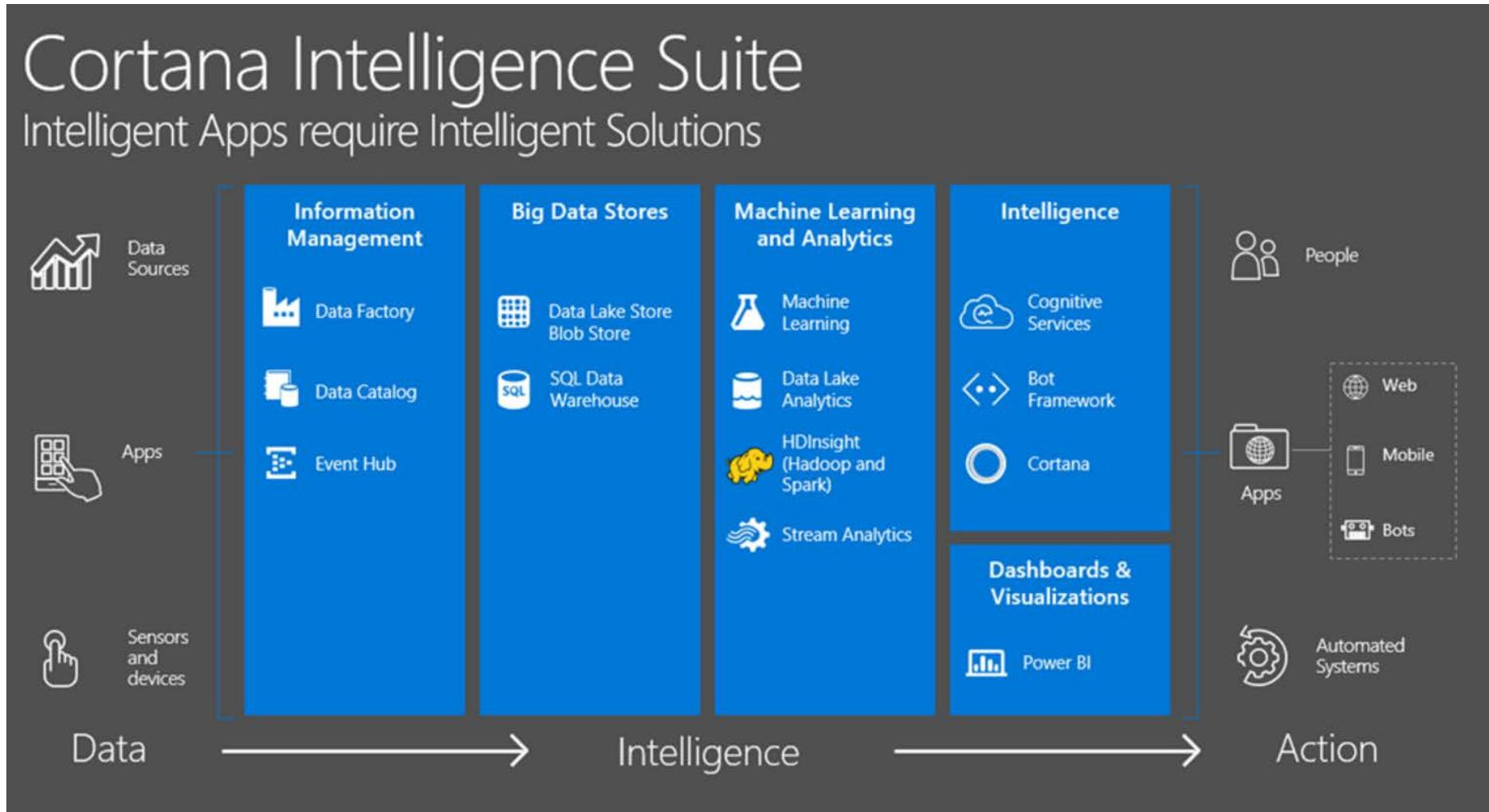
Machine Learning work flow



Machine learning language



Cortana Intelligence Suite (CIS)



Google Machine Learning

Use your own data to train models



TensorFlow



Cloud Machine
Learning Engine

Ready to use Machine Learning models



Cloud
Vision API



Cloud
Speech API



Cloud
Jobs API



Cloud
Translation
API



Cloud Natural
Language API



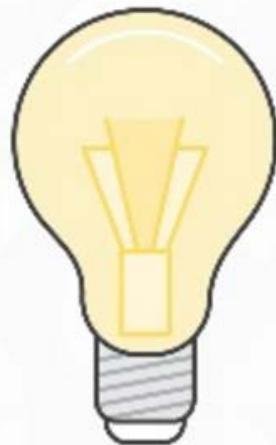
Cloud Video
Intelligence API



Coming
soon

Amazon Machine Learning

Introducing Amazon ML



Easy to use, managed machine learning service built for developers

Robust, powerful machine learning technology based on Amazon's internal systems

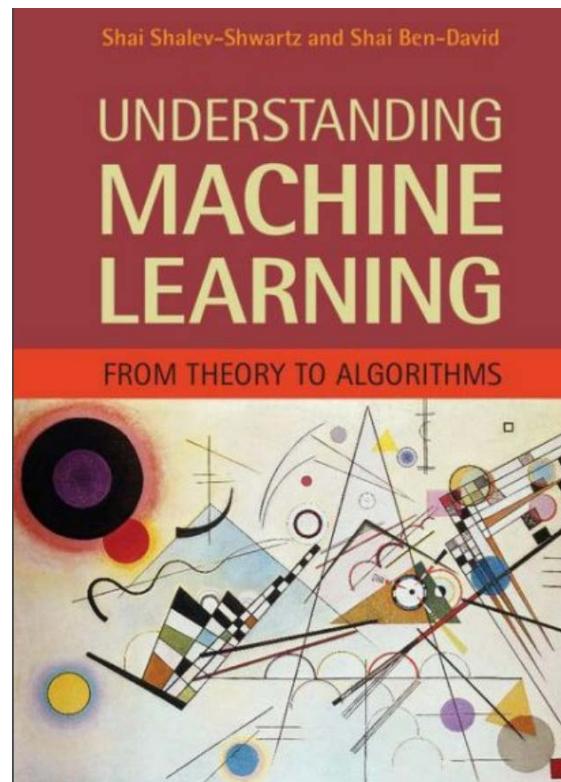
Create models using your data already stored in the AWS cloud

Deploy models to production in seconds



More information

Understanding Machine Learning: From Theory to Algorithms
<http://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/copy.html>



GET FREE AZURE



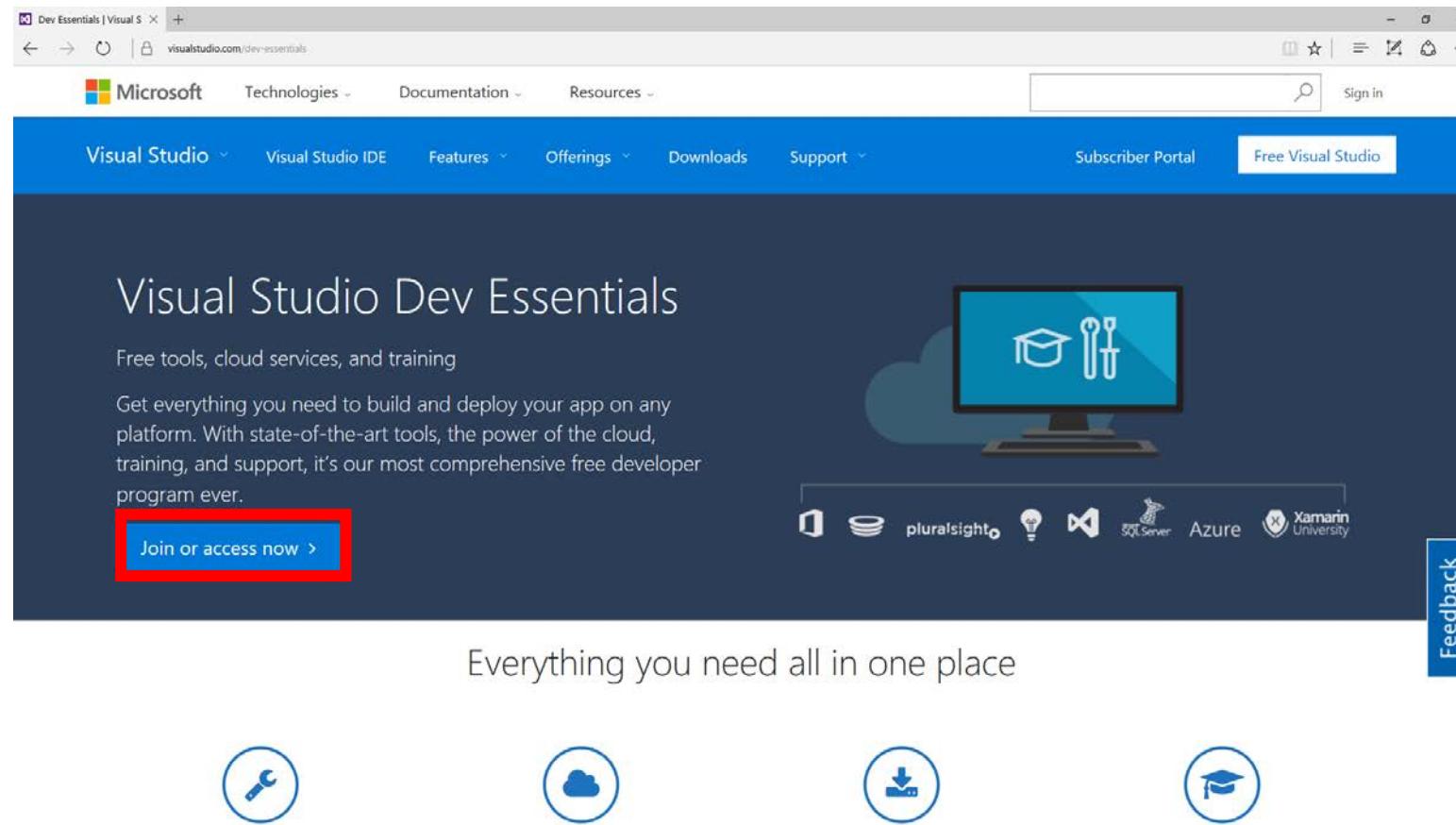
Microsoft
Azure

In this session

- Go to Visual Studio Dev Essentials
- Sign-in: If you already have Microsoft Account
- Create new: If you don't have Microsoft Account
- Check email and get the security code
- Login to Microsoft Live
- Verify security info
- Enter name, country and email address
- Confirm to join Dev Essentials
- Azure \$25 monthly credit for 1 year
- Sign up for Developer Program Benefit
- Identity verification by phone
- Payment information
- Go to Try Microsoft Azure
- Sing-in account select

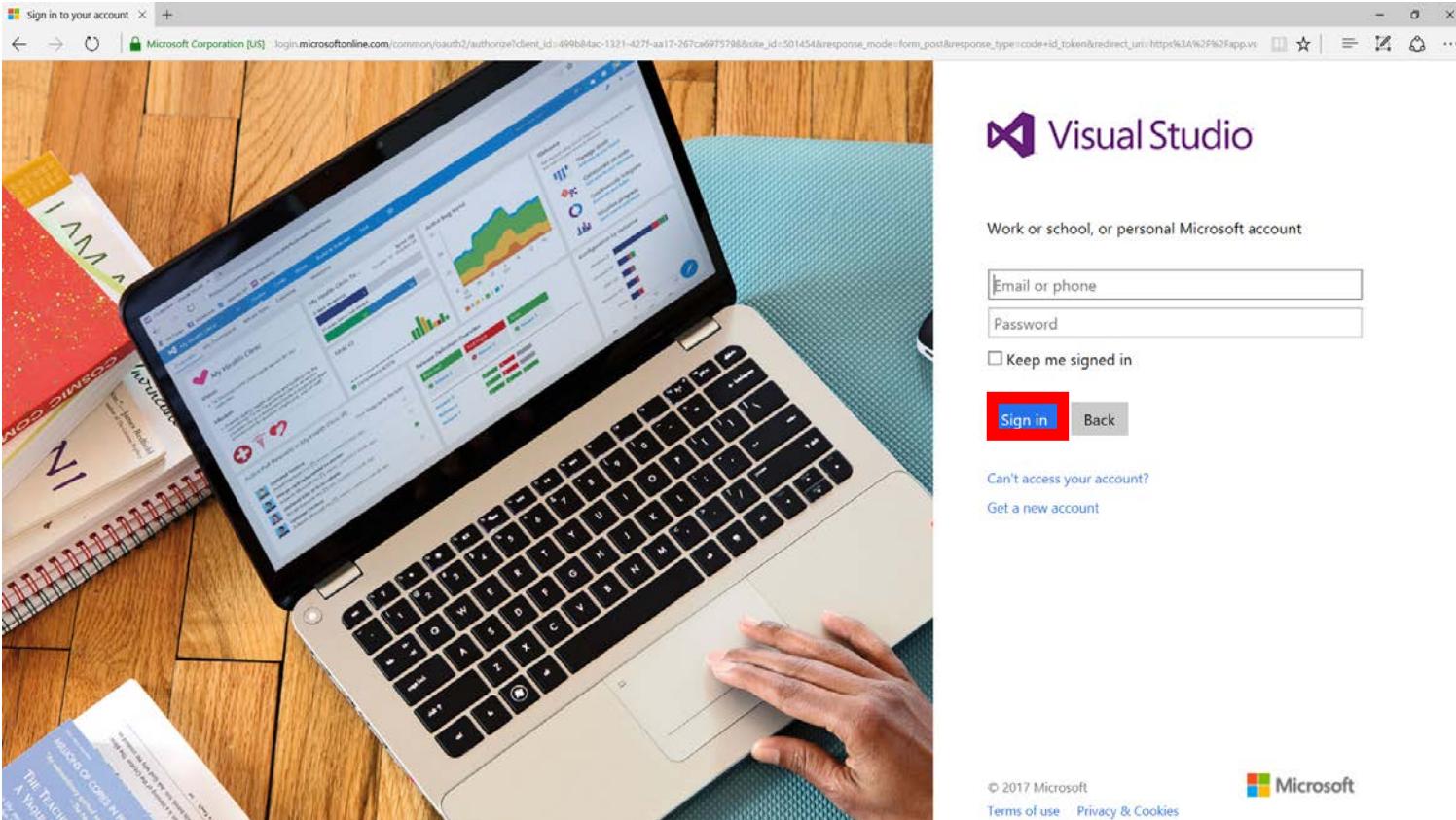
Go to Visual Studio Dev Essentials

<https://www.visualstudio.com/dev-essentials/>



If you already have Microsoft Account

Enter email address & password / Click Sign in



 Visual Studio

Work or school, or personal Microsoft account

Keep me signed in

Sign in

Back

[Can't access your account?](#)

[Get a new account](#)

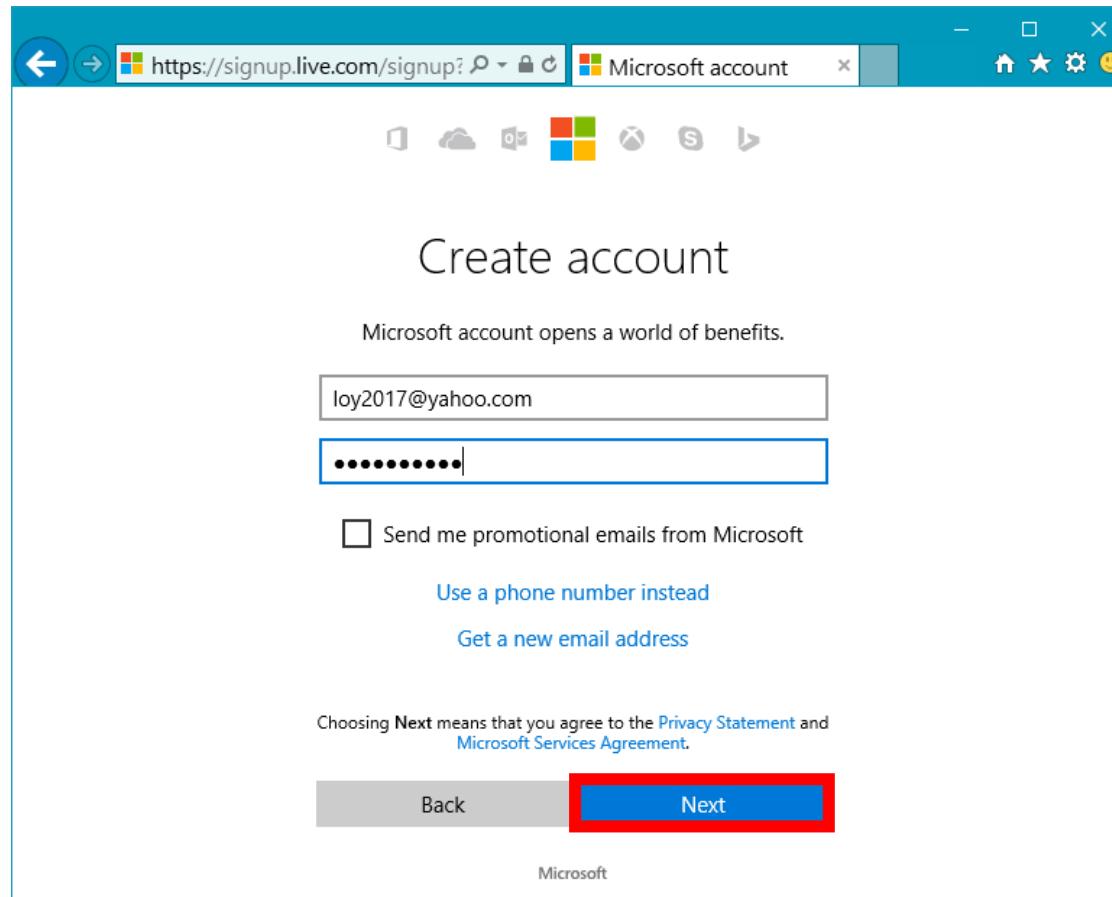
© 2017 Microsoft
[Terms of use](#) [Privacy & Cookies](#)

 Microsoft

If you don't already have Microsoft Account

Signup for a new Microsoft Account

<https://signup.live.com>



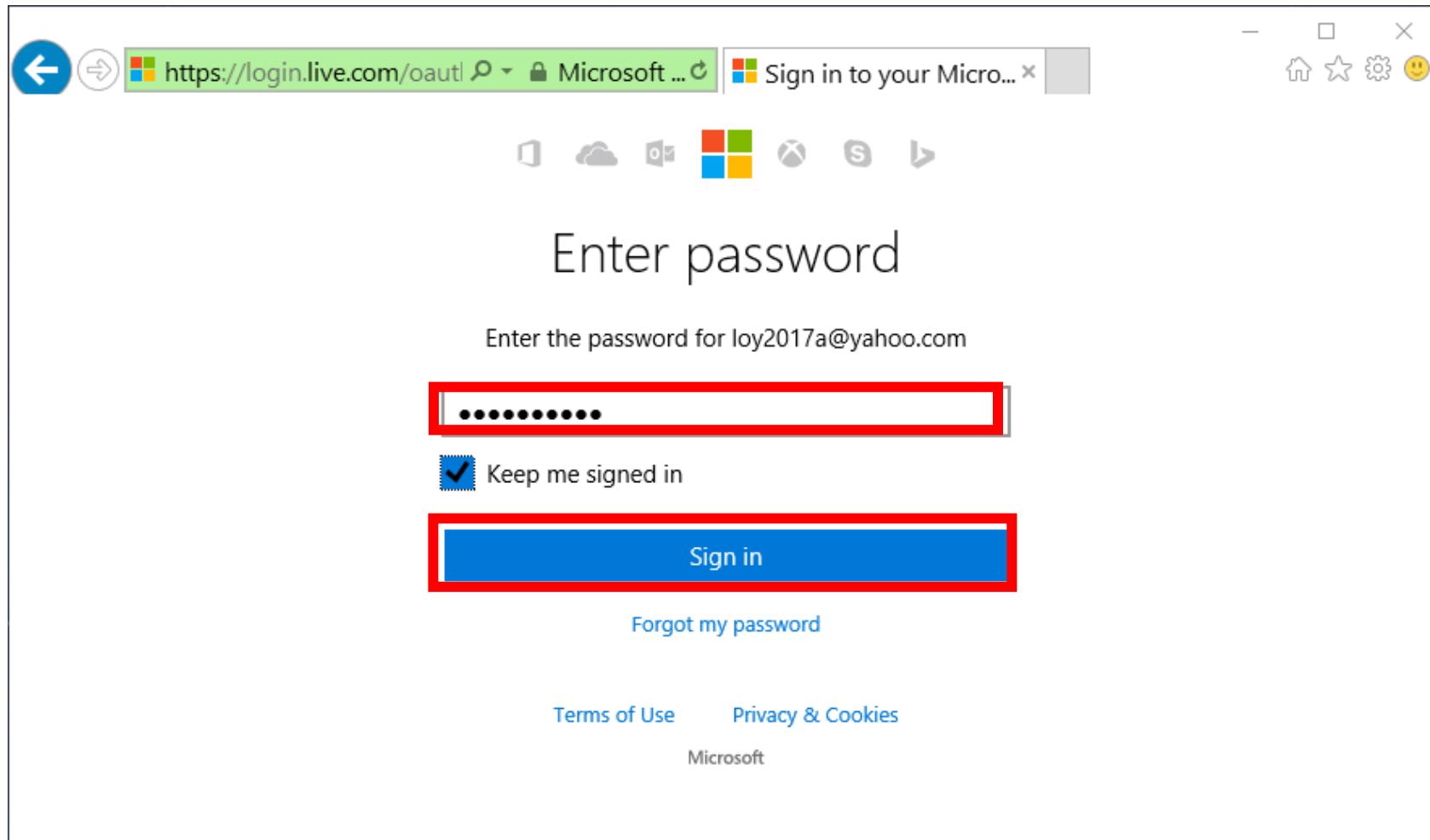
Verify email address

Check email and get the security code

The screenshot shows a Yahoo! Mail inbox. On the left, there's a sidebar with links like Home, Mail, Search, News, Sports, Finance, Celebrity, Weather, Answers, Flickr, Mobile, More, Compose, Add Gmail, Outlook, AOL and more, Inbox, Drafts, Sent, Archive, Spam, Trash, Smart Views (Unread, Starred, People, Social, Shopping, Travel, Finance), Folders, and Recent. The main area shows an email from "Microsoft account team <account-security-noreply@account.microsoft.com>" to "loy2017a@yahoo.com" received "Today at 6:16 AM". The subject is "Verify your email address". The body of the email contains the text: "To finish setting up your Microsoft account, we just need to make sure this email address is yours." Below this, it says "To verify your email address use this security code : **1174**". A red box highlights the code "1174". It also states: "If you didn't request this code, you can safely ignore this email. Someone else might have typed your email address by mistake." At the bottom, there are reply, forward, and more options.

Login to Microsoft Live

<https://login.live.com>



Verify security info

Click Looks good!

The screenshot shows a Microsoft account verification page titled "Is your security info still accurate?". The URL in the address bar is <https://account.live.com/proofs/remind>. The page content includes a message about keeping security info up-to-date, the email address "loy2017a@yahoo.com", and two buttons: "Looks good!" (highlighted with a red box) and "Update now". At the bottom, there's a "Remind me later" link and a footer with links to "© 2017 Microsoft", "Terms of Use", "Privacy & Cookies", and "Developers".

We need a few more details

Enter name, country and email address

The screenshot shows a web browser window with the URL <https://app.vsaex.visualstudio.com/profile/> in the address bar. The page title is "Review Information |...". The top navigation bar includes "Visual Studio" and "My Benefits" on the left, and "loy2017a@yahoo.com" and "Sign out" on the right.

The main content area displays the message "We need a few more details". It contains three input fields:

- "Your name:" input field containing "loy2017a"
- "From:" dropdown menu showing "Thailand"
- "We'll reach you at:" input field containing "loy2017a@yahoo.com"

Below these fields is a checkbox with the text: "Microsoft may use your contact information to provide updates and special offers about Visual Studio. You can unsubscribe at any time." A red box highlights the "Continue" button.

At the bottom, there is a note: "To keep our lawyers happy: By continuing, you agree to the [Terms of Service](#) and the [Privacy Statement](#).
A red box highlights the "Continue" button.

Accept the terms of the program

Click Accept

The screenshot shows the 'My Benefits' section of the Visual Studio website. At the top, there's a navigation bar with links for 'Benefits', 'Downloads', 'Product Keys', 'Subscriptions', 'Support', and 'Marketplace'. A user profile 'loy2017a' is shown along with a 'Sign out' link and a search icon. Below the navigation, it says 'Showing: Visual Studio Dev Essentials'. A message at the top states: 'To access the following Visual Studio Dev Essentials benefits, please accept the terms of the program.' A blue 'Accept' button is highlighted with a red box. The main content area features a large banner for 'Visual Studio Dev Essentials' with icons for iOS, Android, Windows, Linux, and various cloud services. Below the banner, there are sections for 'Featured' and 'Tools'. The 'Featured' section contains six items: 'Visual Studio Community' (Full-featured, extensible IDE), 'Visual Studio Code' (Modern lightweight editor), 'Visual Studio for Mac' (Preview), 'Visual Studio Team Services' (Basic level), 'Azure' (\$25 monthly credit for 1 year), and 'Xamarin University Training' (Free on-demand access). Each item has a 'Download' or 'Get Code' button. The 'Tools' section contains six items: 'Microsoft R Server' (Developer Edition), 'Microsoft SQL Server' (Developer Edition), 'Azure App Service' (Free plan), 'Application Insights' (Free plan), 'HockeyApp' (Free plan), and 'Universal Windows Platform' (60 day trial VM). Each tool has a 'Download', 'Use it free', or 'Try it free' button.

Confirm to join Dev Essentials

Click Confirm

×

Welcome to Visual Studio Dev Essentials

We're glad you're here!

By joining Visual Studio Dev Essentials, you get a wide range of free benefits from development tools to online training to help you build and deploy your apps on any platform.

Here are some of the great benefits:

- \$25 monthly Azure credit for 12 months
 - On-demand access to Xamarin University
 - 3-month subscription to Pluralsight
- Periodic email communications with latest trends, news, benefit and product announcements

By confirming to join, you accept these [Terms & Conditions](#).

You can leave the program any time to stop receiving communications and access to your benefits by going to the Subscriptions tab.

Review our [Privacy Statement](#).

Confirm

Cancel

Azure \$25 monthly credit for 1 year

Click Activate

The screenshot shows the 'Visual Studio | My Benefits' interface. At the top, there are tabs for Benefits, Downloads, Product Keys, Subscriptions, Support, and Marketplace. The user is signed in as 'loy2017a'. A search bar is on the right.

The main area displays a welcome message for 'Visual Studio Dev Essentials' and a graphic showing various development platforms like iOS, Android, Windows, and Linux integrated with cloud services.

Featured (6)

- Visual Studio Community**: Full-featured, extensible IDE. Free for individuals, open source or small teams. Create apps for Windows, iOS, macOS, and Linux. [Download](#) [Activate](#)
- Visual Studio Code**: Modern lightweight editor. A powerful, streamlined code editor for your favorite platform - Linux, Mac OS X, and Windows. [Download](#)
- Visual Studio for Mac**: Preview. Build apps for the cloud, iOS, Android, macOS, and wearables. [Download](#)
- Visual Studio Team Services**: Basic level. Free Git repos, Agile planning tools and hosted builds, for any language – it's the perfect team service. [Get started](#)
- Azure**: \$25 monthly credit for 1 year. Your own personal sandbox for dev/test VMs, cloud services, and more. Credit cannot be transferred. [Activate](#)
- Xamarin University Training**: Free on-demand access. Build native iOS and Android apps in C# with expert getting-started videos (subset of class). [Get Code](#)

Tools (10)

- Microsoft R Server**: Developer Edition. Build Advanced Analytics solutions in R on Windows, Hadoop, Terradata and Linux. [Download](#)
- Microsoft SQL Server**: Developer Edition. Build mission-critical data solutions with unparalleled security, mobile BI, and more. [Download](#)
- Azure App Service**: Free plan. Everything you need to quickly and easily build web and mobile apps for any platform. [Use it free](#)
- Application Insights**: Free plan. Gain deep insight into the health and performance of your web app no matter where it runs. [Use it free](#)
- HockeyApp**: Free plan. Beta test mobile apps, track crashes, gather feedback and drive improvements with one tool. [Use it free](#)
- Universal Windows Platform**: 60 day trial VM to help you get started developing on Windows hassle-free. Comes with a developer license for Windows 10 Pro. [Try it free](#)

Sign up for Developer Program Benefit

Enter information and click Next

Microsoft Azure Sign up Loy2017a@yahoo.com | Sign Out

Developer Program Benefit

Learn more ▾

1 About you

* Country/Region ⓘ
Thailand

* First Name
manha

* Last Name
vanich

* Email address for important notifications ⓘ
loy2017a@yahoo.com

* Work Phone
8 4007 5544

Organization
- Optional -

Next

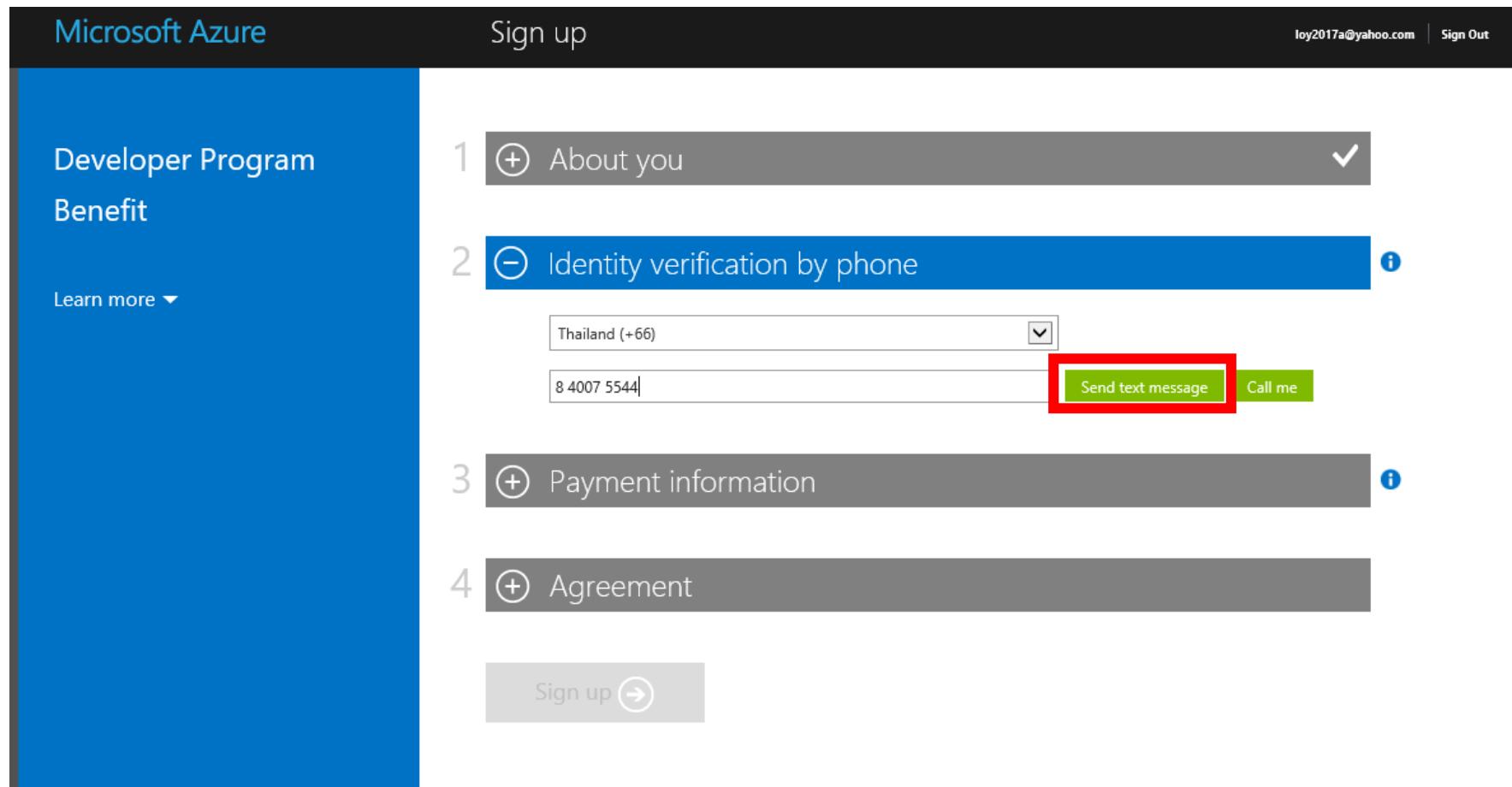
2 Identity verification by phone ⓘ

3 Payment information ⓘ

4 Agreement ⓘ

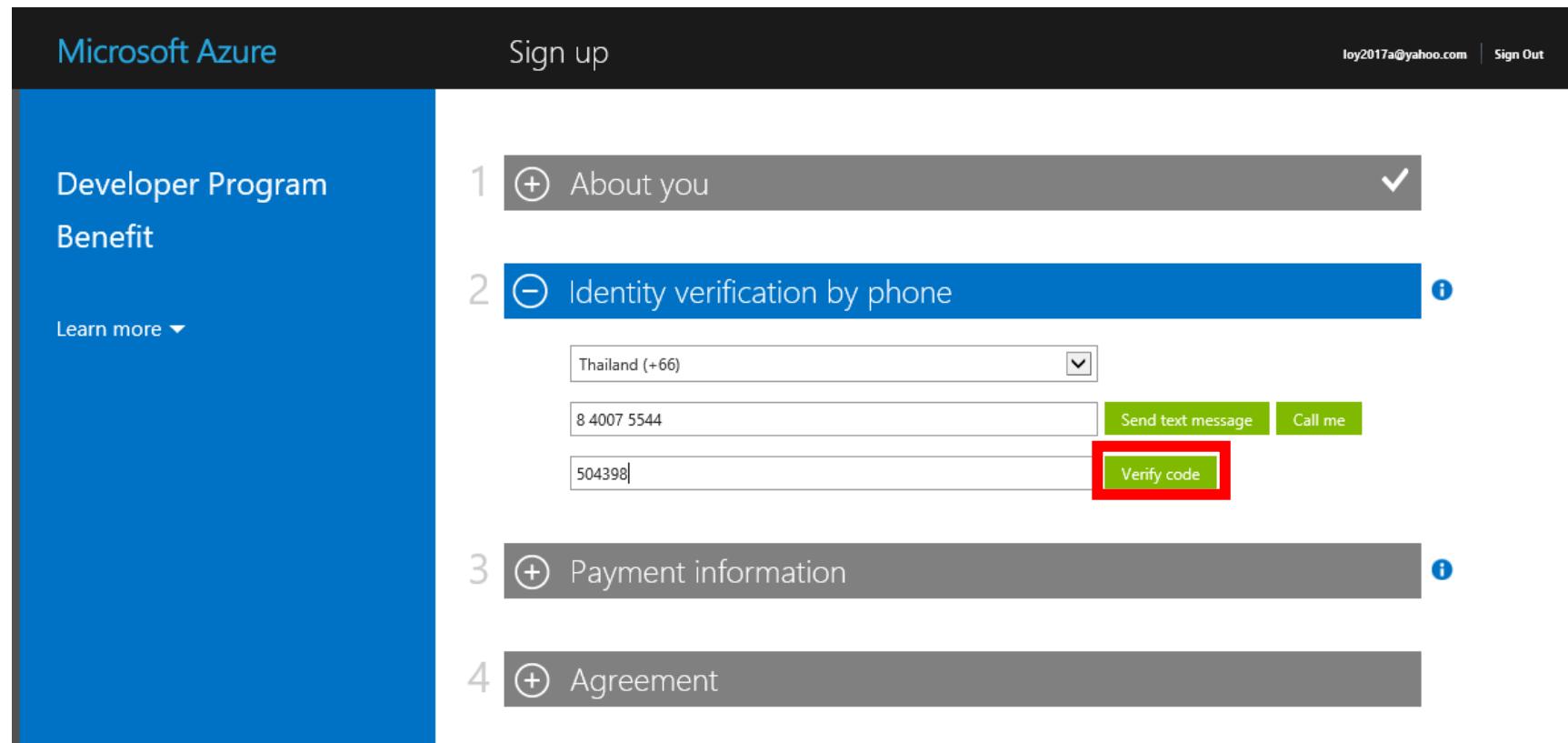
Identity verification by phone

Select country and enter phone number / Click Send text message



Identify verification by phone

Check phone SMS message / enter code / Click Verify code



Payment information

Enter credit or debit card information / For identity (no charge)

3  Payment information

Payment method

 
* Card number

* Expiration date * CVV 
 MM YYYY

* Name on card

* Address line 1

Address line 2
 - Optional -

* City

Province * Postal Code - 10330 -

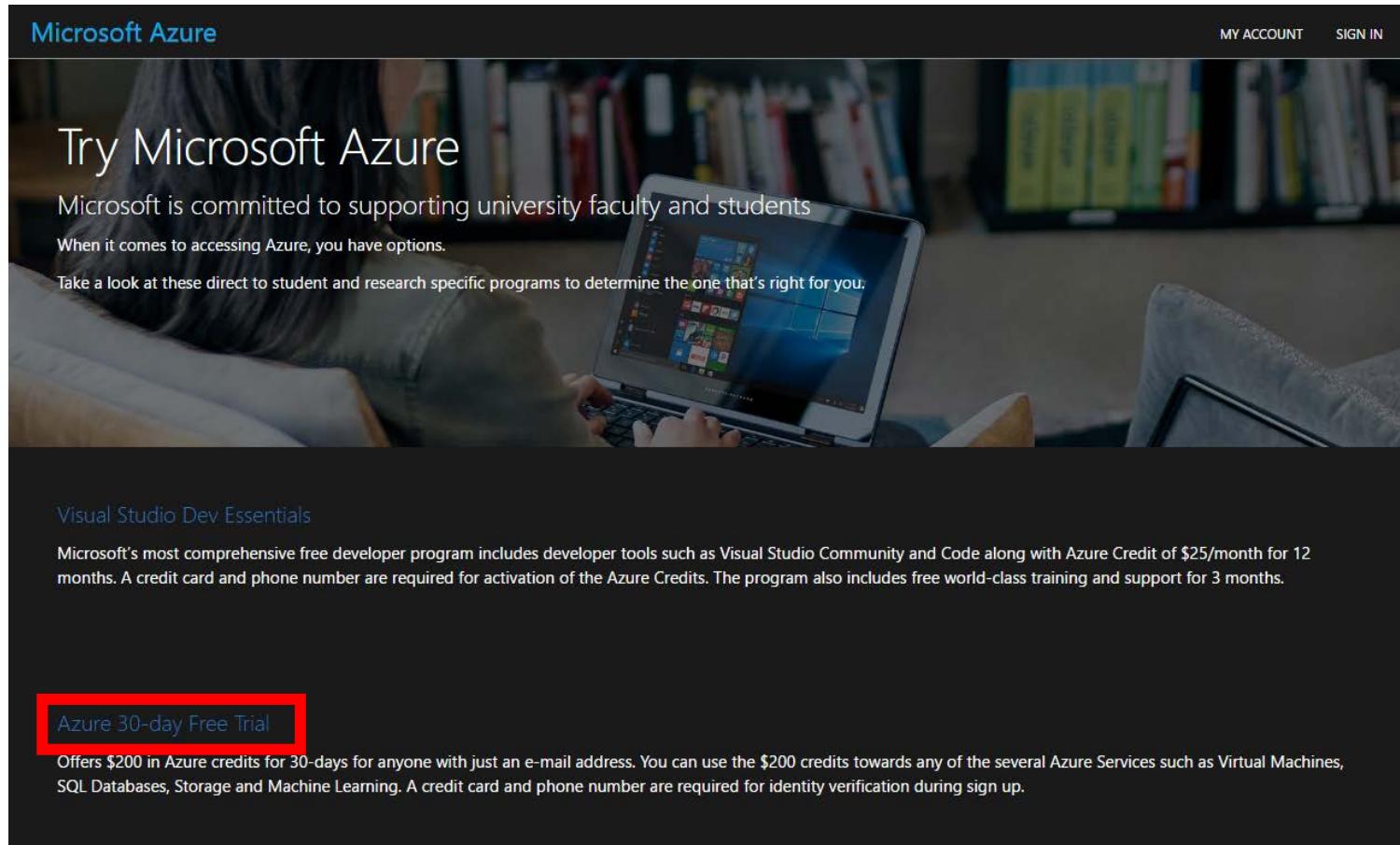
Phone number
 - Prefix - - Number -



Go to Try Microsoft Azure

<https://www.microsoftazurepass.com/azureu>

Click Azure 30-day Free Trial



Create your free Azure account today

Click Start free >

The screenshot shows the Microsoft Azure homepage with a dark header bar. The header includes links for Sales (1-800-867-1389), My Account, Portal, and a search bar. Below the header, there's a navigation bar with links for Why Azure, Solutions, Products, Documentation, Pricing, Partners, Blog, Resources, and Support.

The main content area features a large blue banner with the text "Create your free Azure account today". Below the banner, there are three sections:

- Get \$200 free credit**: Includes a gift icon and text: "Start free with \$200 in credit, and keep going with free options."
- Try any Azure services**: Includes a cloud icon and text: "Explore our cloud by trying out any combination of Azure services for 30 days."
- Pay nothing at the end**: Includes a credit card icon and text: "We use your credit card information for identity verification, but you'll never be charged unless you choose to subscribe."

At the bottom of this section is a green button labeled "Start free >" which is highlighted with a red box.

Below these sections are links for "Or buy now >" and "Frequently asked questions >". There's also a phone icon followed by the text "Call sales 1-800-867-1389".

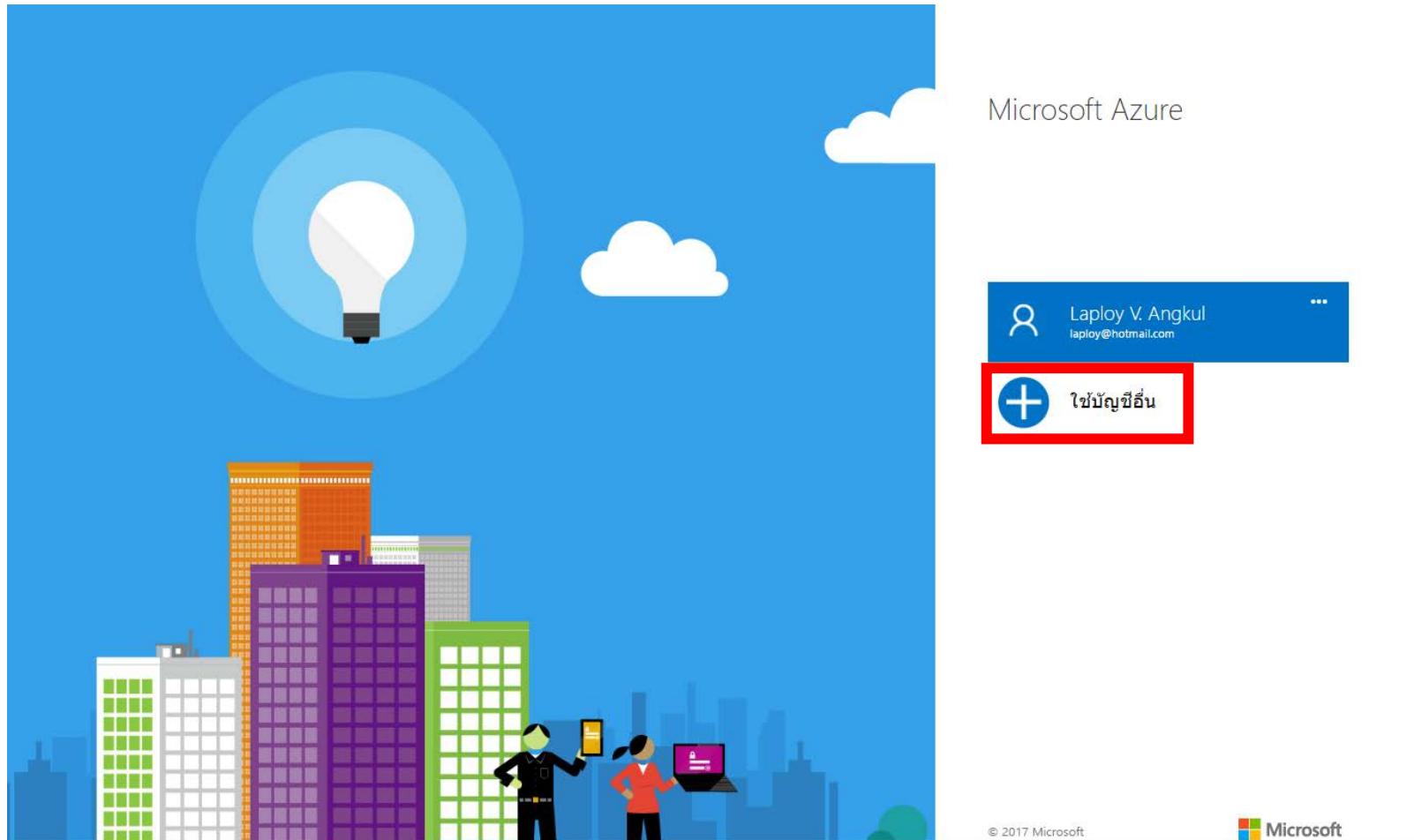
A modal window titled "Web + Mobile" is overlaid on the page. It shows a list of "FEATURED APPS" including "Web App", "Mobile App", "API App", and "Logic App (preview)". On the right side of the modal, there's a form for creating a "Web App". The form fields include:

- * App Service Name: Enter a name for your App. (Example: azurewebsites.net)
- * Subscription: Microsoft Azure Internal Consumption
- * Resource Group: Default
- * App Service plan/Location: ServicePlan(East US)

At the bottom of the modal is a "Create" button.

Sing-in account select

Click at the account to sign-in



More information

More information on Azure subscription

Azure subscription and service limits, quotas, and constraints

<https://docs.microsoft.com/en-us/azure/azure-subscription-service-limits>

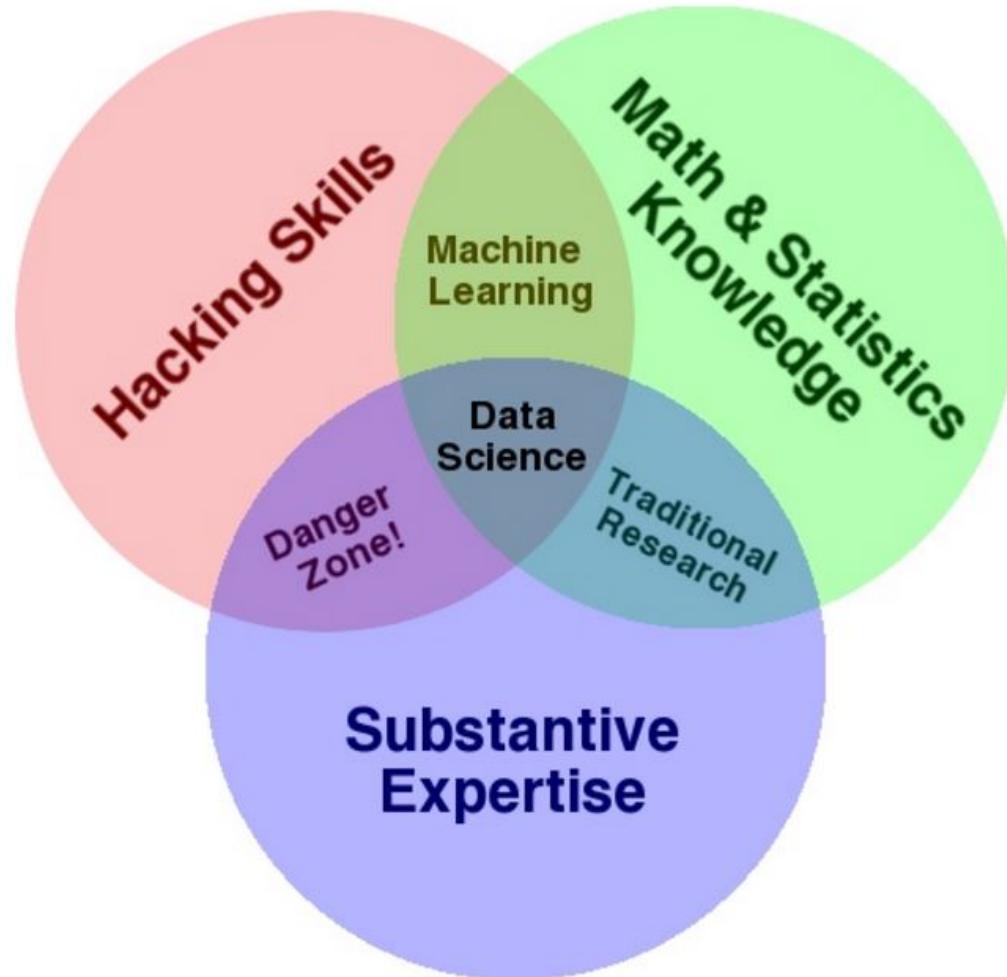
DATA SCIENCE INTRODUCTION



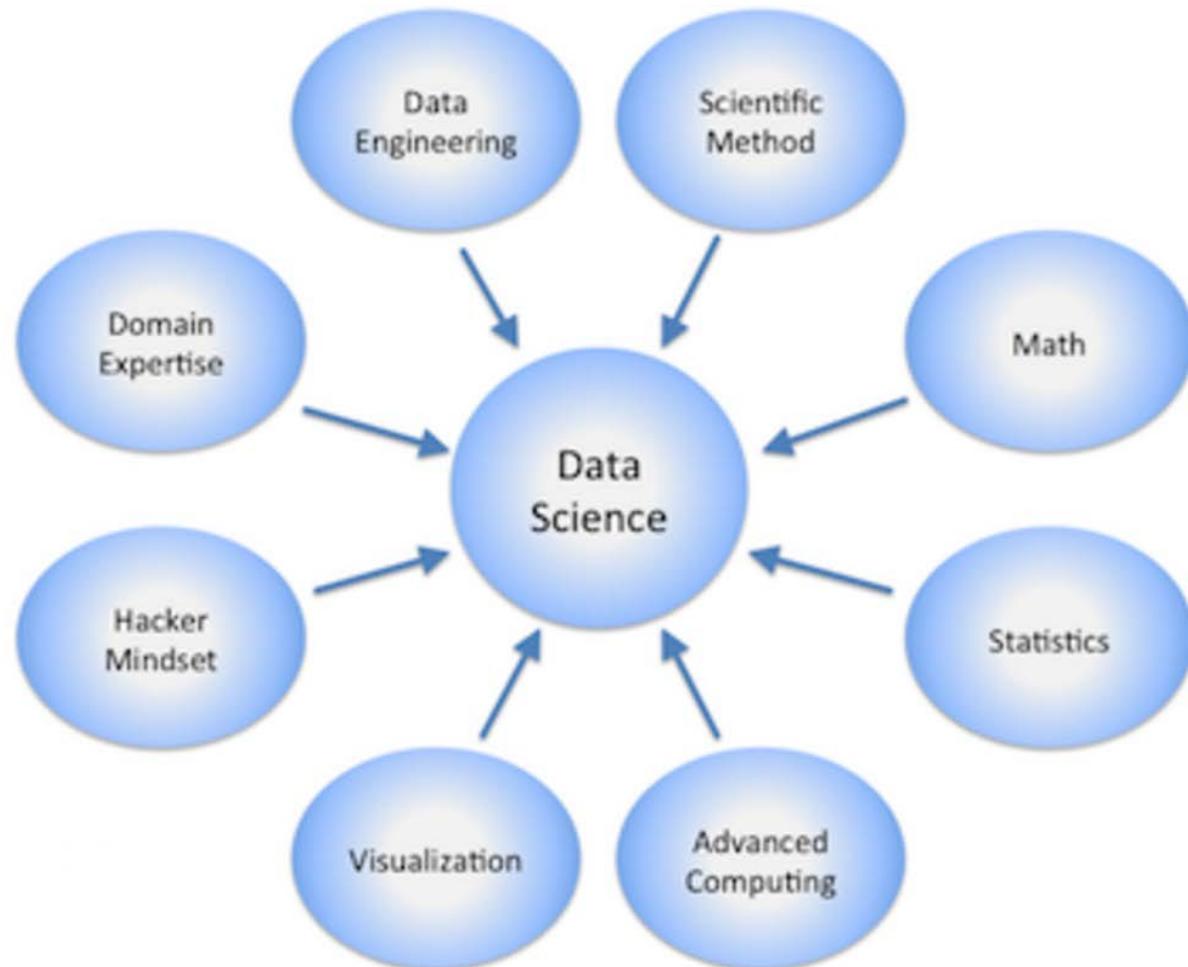
In this session

- Venn diagram of data science
- What is data science?
- Data scientist
- Glassdoor best job in 2016 - 2017
- Data science job trend
- Data science job
- Data Scientist education levels
- Data science backgrounds
- Key topic to learn
- Learn Python library stack
- Go kaggle
- Get your degree
- Investigate the team
- Interview question type
- Take-home machine learning task
- Whiteboard coding
- Whiteboard SQL
- Bayes' theorem
- Machine learning evaluation metrics
- Data Science job facts
- DS compared to ML engineer
- More information on Data Science

What is Data Science



What is data science?



Data scientist

MATH & STATISTICS

- ★ Machine learning
- ★ Statistical modeling
- ★ Experiment design
- ★ Bayesian inference
- ★ Supervised learning: decision trees, random forests, logistic regression
- ★ Unsupervised learning: clustering, dimensionality reduction
- ★ Optimization: gradient descent and variants

DOMAIN KNOWLEDGE & SOFT SKILLS

- ★ Passionate about the business
- ★ Curious about data
- ★ Influence without authority
- ★ Hacker mindset
- ★ Problem solver
- ★ Strategic, proactive, creative, innovative and collaborative

PROGRAMMING & DATABASE

- ★ Computer science fundamentals
- ★ Scripting language e.g. Python
- ★ Statistical computing packages, e.g., R
- ★ Databases: SQL and NoSQL
- ★ Relational algebra
- ★ Parallel databases and parallel query processing
- ★ MapReduce concepts
- ★ Hadoop and Hive/Pig
- ★ Custom reducers
- ★ Experience with xaaS like AWS

COMMUNICATION & VISUALIZATION

- ★ Able to engage with senior management
- ★ Story telling skills
- ★ Translate data-driven insights into decisions and actions
- ★ Visual art design
- ★ R packages like ggplot or lattice
- ★ Knowledge of any of visualization tools e.g. Flare, D3.js, Tableau



Glassdoor best job in 2016 - 2017

2016

Data Scientist (#1), Tax Manager (#2) and Solutions Architect (#3) stand out as the three Best Jobs in America for 2016. But which other jobs made the cut?

<https://www.glassdoor.com/blog/25-jobs-america-2016/>

2017

1 Data Scientist



4.8 / 5
Job Score

\$110,000
Median Base Salary

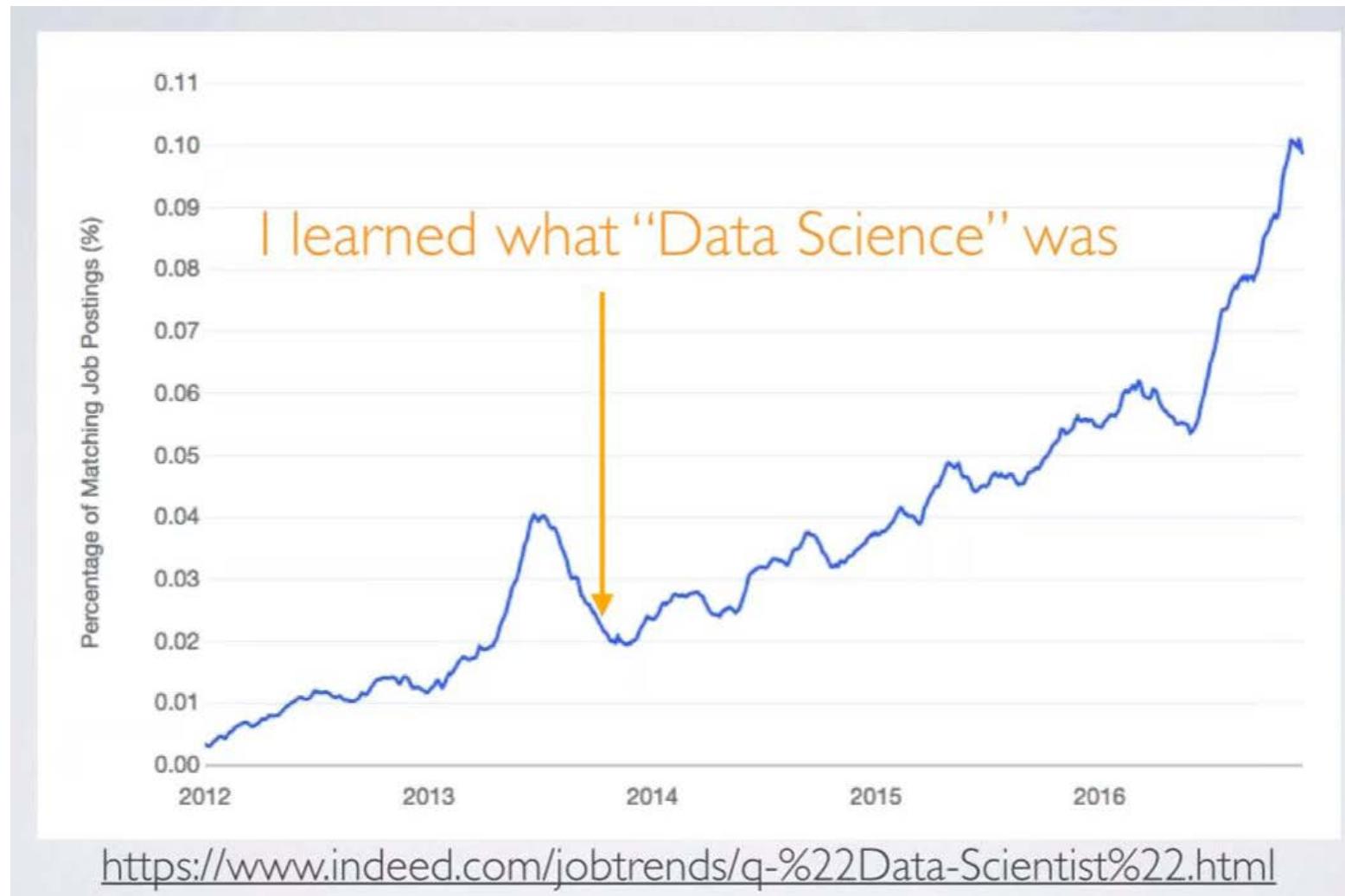
4.4 / 5
Job Satisfaction

4,184
Job Openings

[View Jobs](#)

https://www.glassdoor.com/List/Best-Jobs-in-America-LST_KQ0,20.htm

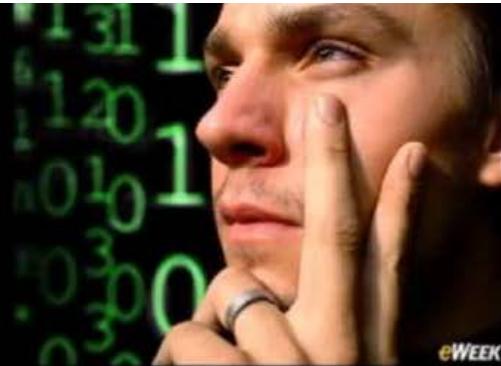
Data science job trend



Data science job



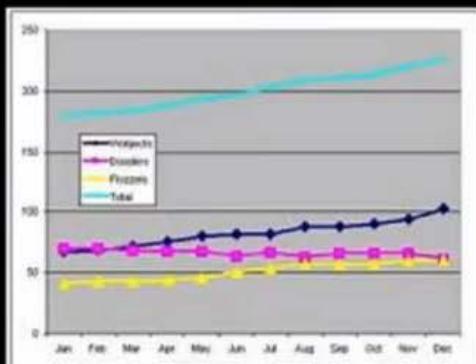
What my friends think I do



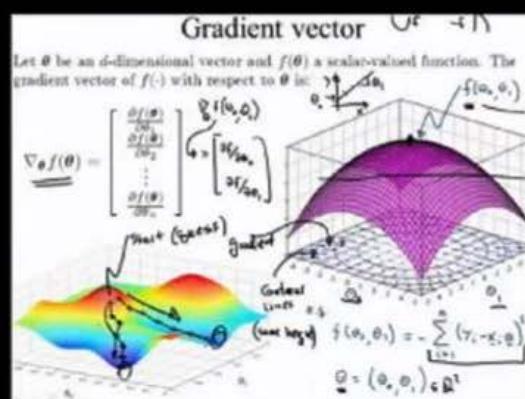
What my mom thinks I do



What society thinks I do



What my boss thinks I do

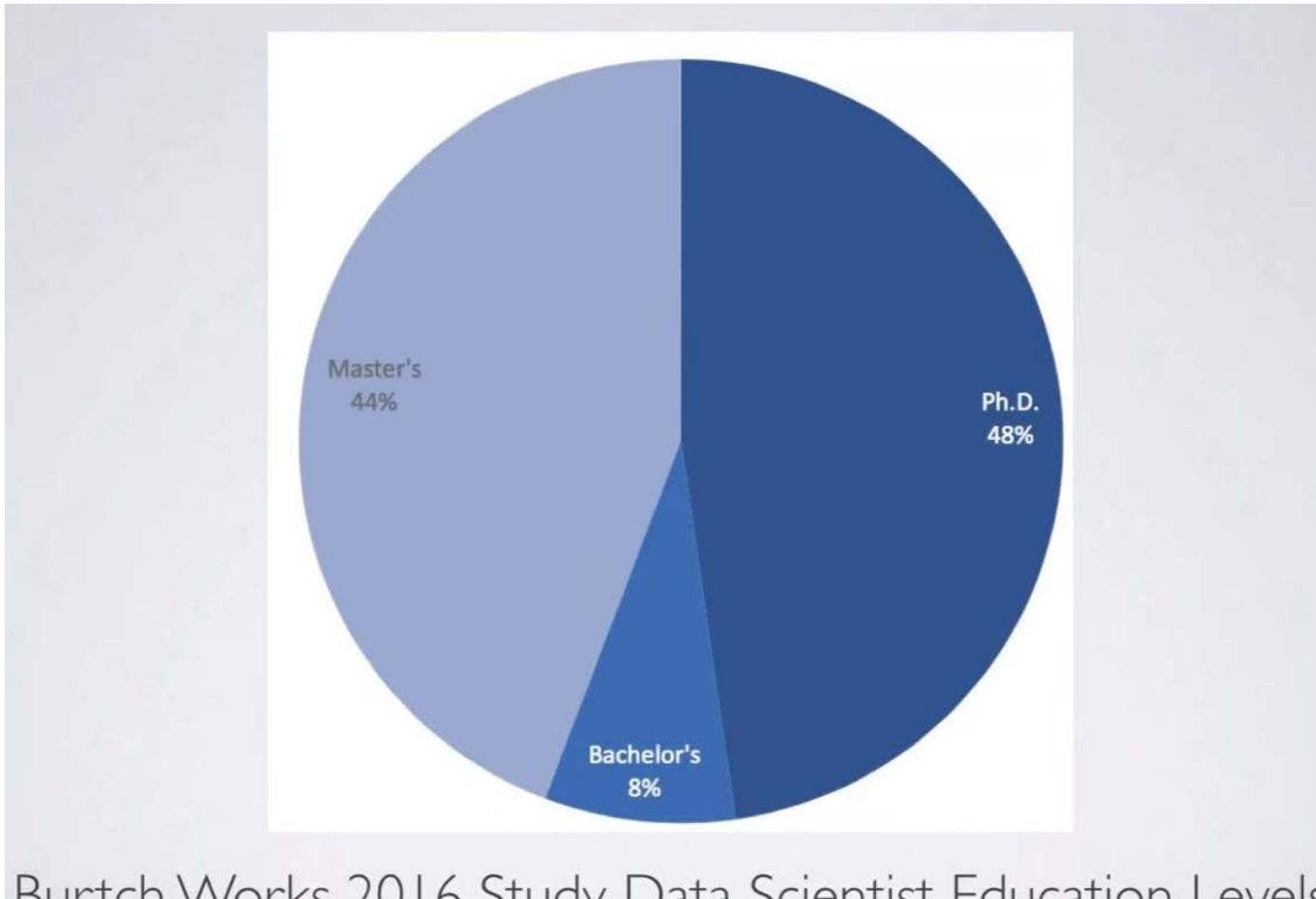


What I think I do

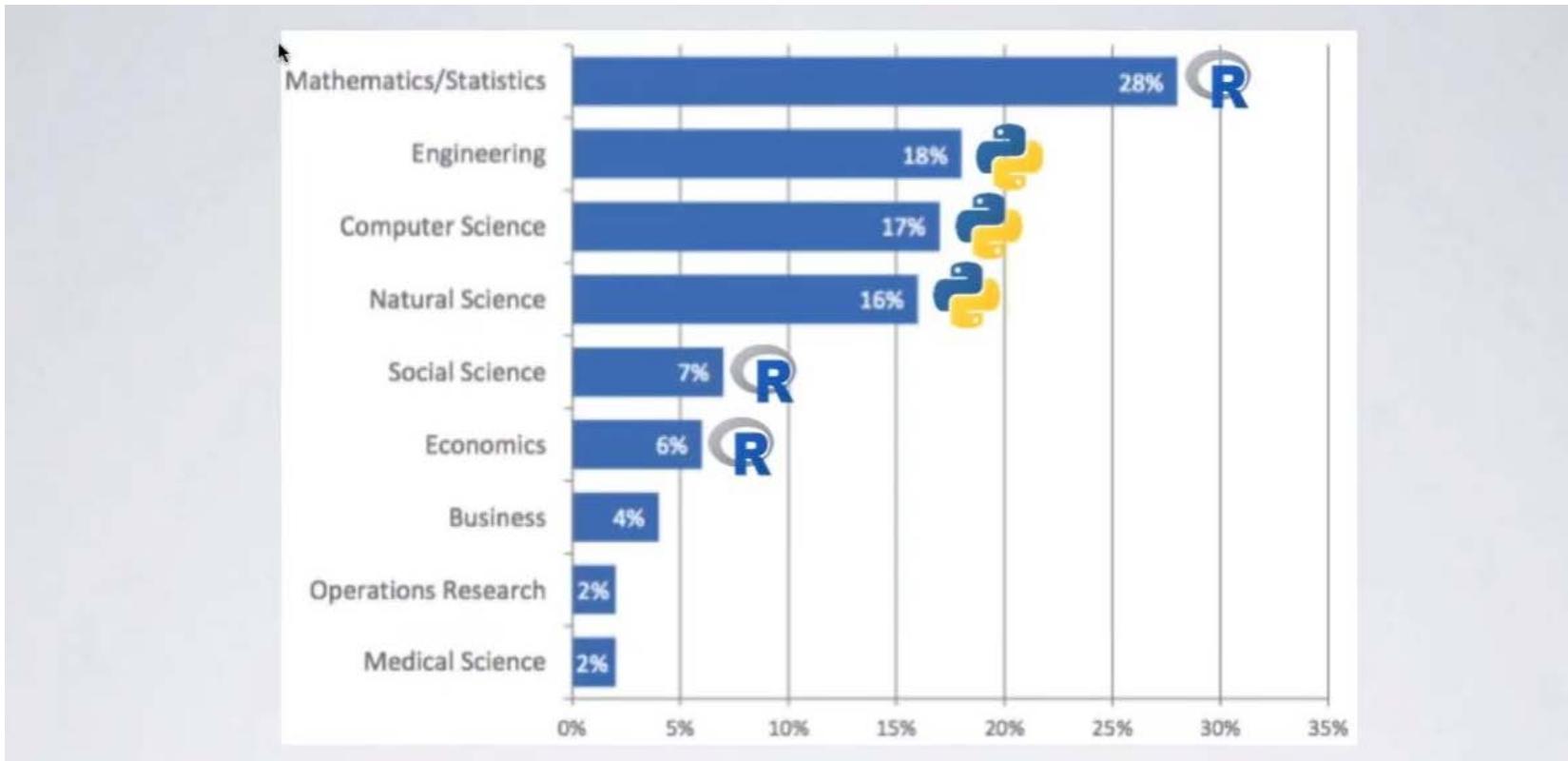


What I actually do

Data Scientist education levels



Data science backgrounds



Burtsch Works 2016 Study, Data Scientist Backgrounds

http://www.burtschworks.com/files/2016/04/Burtsch-Works-Study_DS-2016-final.pdf

Key topic to learn

I. Pick an **open-source** language well-designed for Data Science



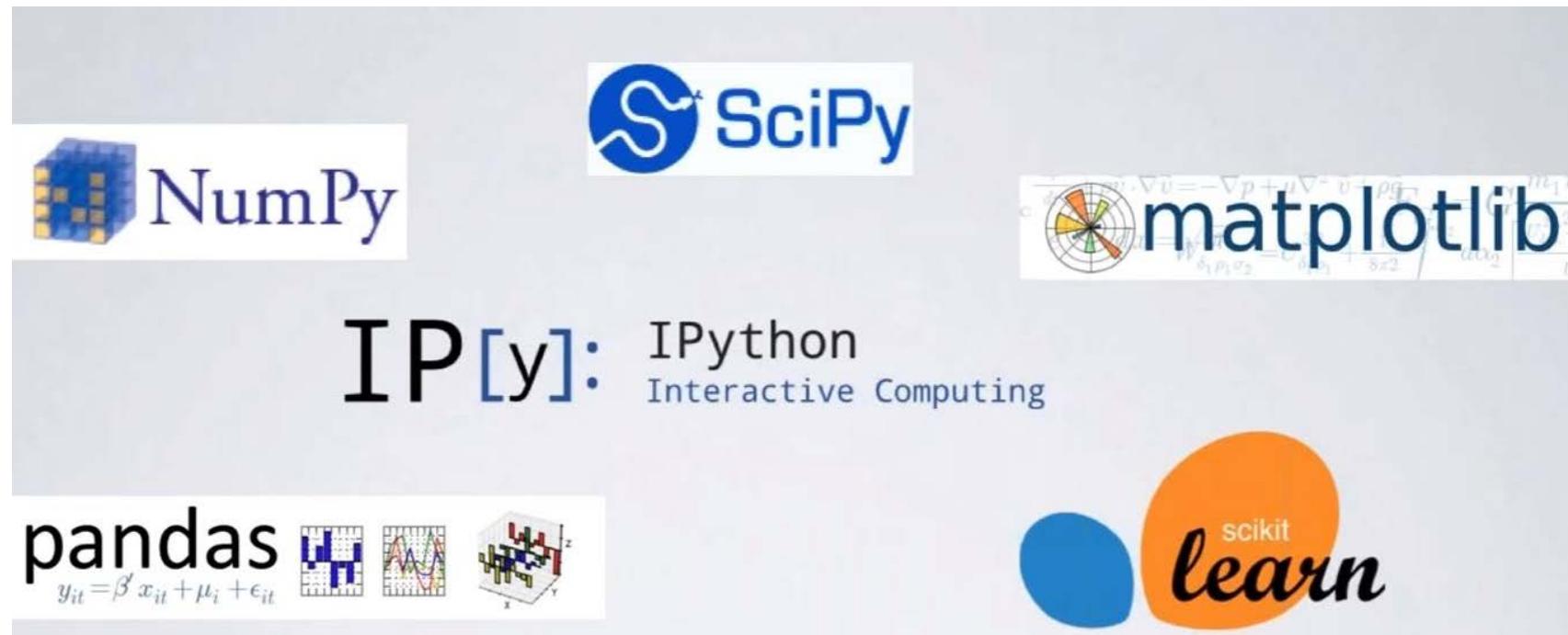
or



Python (my recommendation)

R (**if** you already know it well)

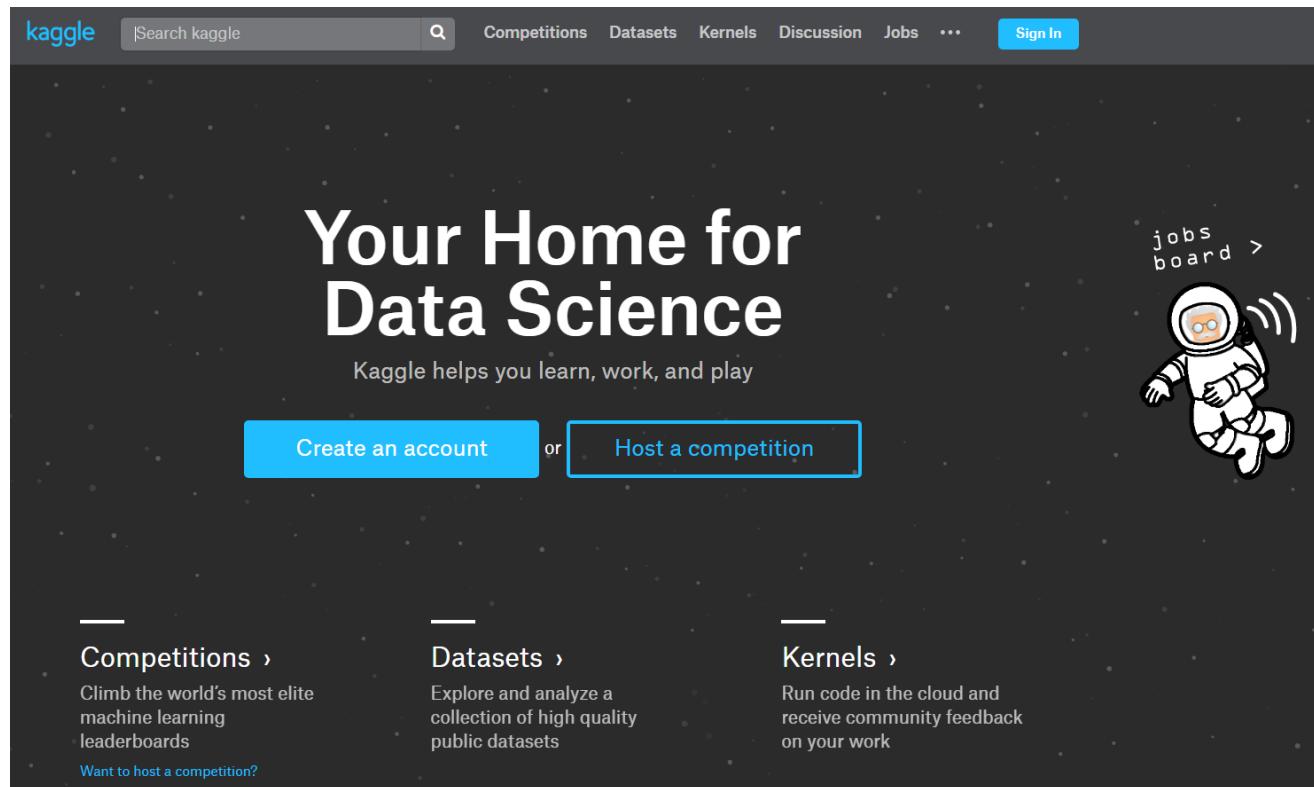
Learn Python library stack



Go kaggle

- There are countless strategies that can be applied to any predictive modelling
- It is impossible to know at the outset which technique or analyst will be most effective
- Compete to produce the best models

kaggle™



The screenshot shows the homepage of Kaggle. At the top, there is a navigation bar with the 'kaggle' logo, a search bar containing 'Search kaggle', and links for 'Competitions', 'Datasets', 'Kernels', 'Discussion', 'Jobs', and 'Sign In'. Below the navigation bar, the main heading reads 'Your Home for Data Science' in large white text. A subtext below it says 'Kaggle helps you learn, work, and play'. There are two prominent blue buttons: 'Create an account' and 'Host a competition'. To the right of the main heading, there is a cartoon illustration of an astronaut floating in space with a 'jobs board' sign above them. At the bottom of the page, there are three sections: 'Competitions', 'Datasets', and 'Kernels', each with a brief description and a 'Want to host a competition?' link.

Competitions ›
Climb the world's most elite machine learning leaderboards
[Want to host a competition?](#)

Datasets ›
Explore and analyze a collection of high quality public datasets

Kernels ›
Run code in the cloud and receive community feedback on your work



A nine-course introduction to data science, developed and taught by leading professors.

Johns Hopkins University (commonly referred to as **Johns Hopkins, JHU**, or simply **Hopkins**) is an American [private research university](#) in Baltimore, Maryland. Founded in 1876,

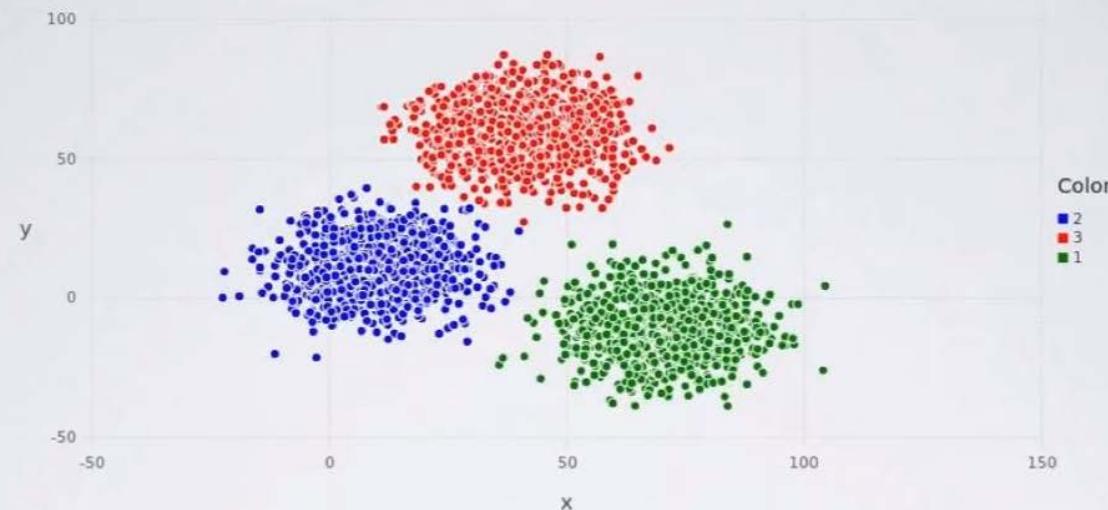
Ask the right questions, manipulate data sets, and create visualizations to communicate results.

This Specialization covers the concepts and tools you'll need throughout the entire data science pipeline, from asking the right kinds of questions to making inferences and publishing results. In the final Capstone Project, you'll apply the skills learned by building a data product using real-world data. At completion, students will have a portfolio demonstrating their mastery of the material.

Created by:
 JOHNS
HOPKINS
UNIVERSITY

Investigate the teams

Teams **tend to** like hiring people similar to themselves.

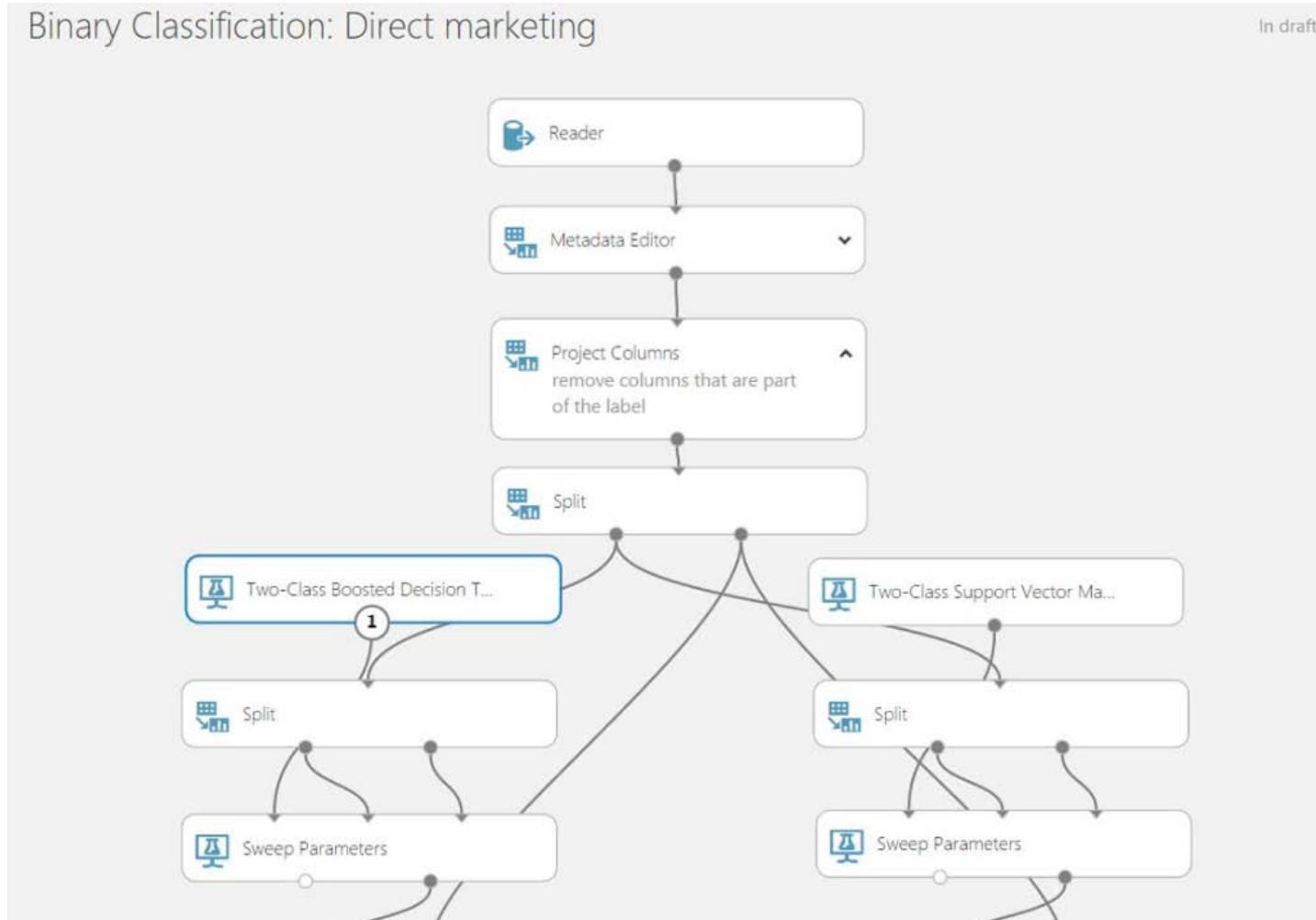


- No Ph.D. on the team? They probably don't want one.
- All team members have a Ph.D? You probably need one.
- Are most of them computer scientists? Physical scientists? Social scientists?
 - Do they seem to prefer Python, R, or a mix?

Interview question type

- Take-home machine learning task
- “Whiteboard” coding (focus on Data Structures/Algorithms)
- “Whiteboard” SQL
- Bayes' Theorem probability questions
- Machine learning evaluation metrics

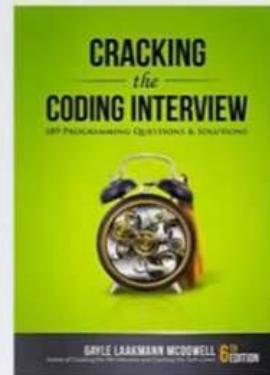
Take-home machine learning task



Whiteboard coding

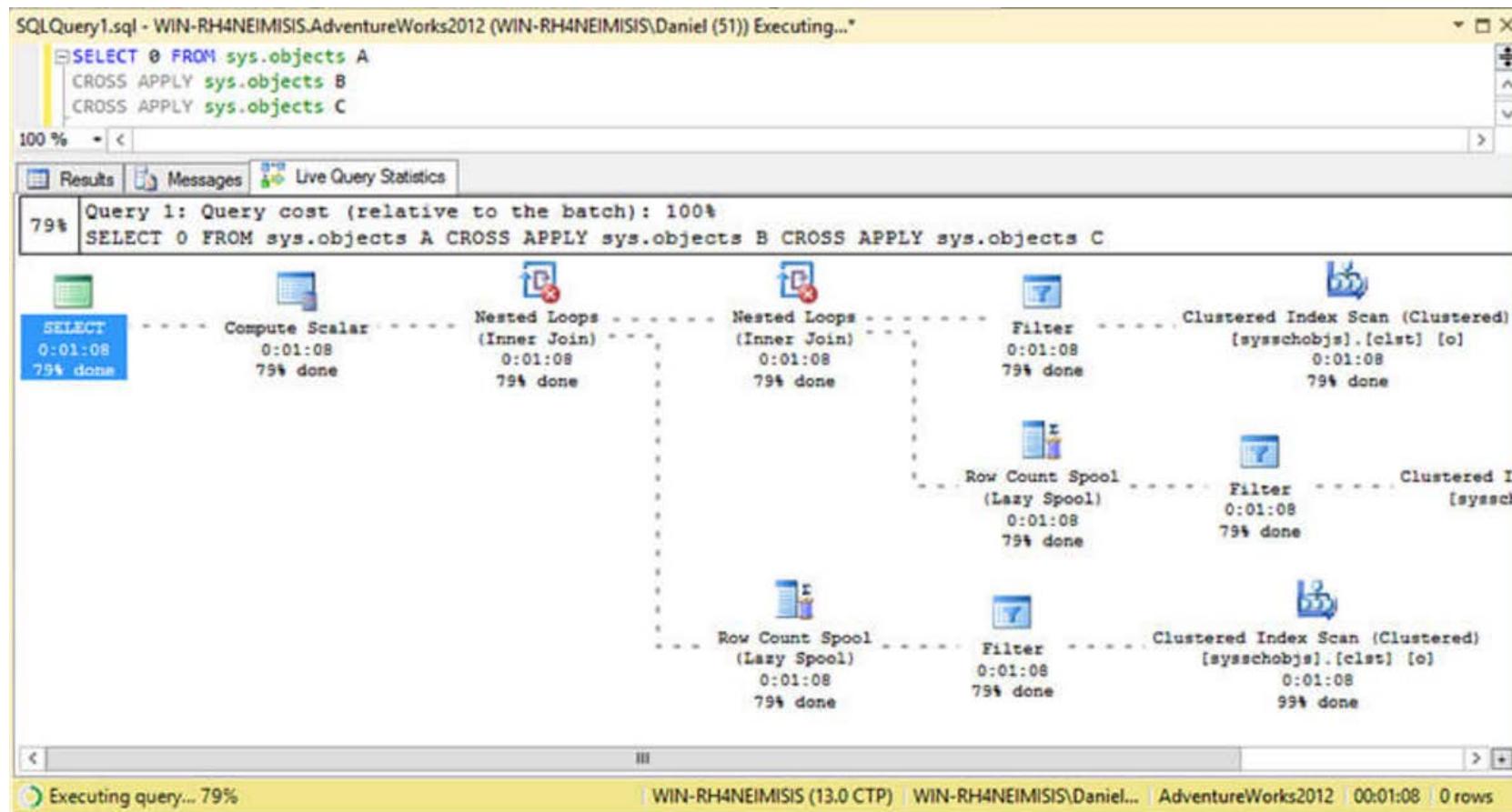
Tends to be similar to software engineer interviews, but focuses most on data structures/algorithms

Practice with:



<https://www.amazon.com/Cracking-Coding-Interview-Programming-Questions/dp/0984782850>

Whiteboard SQL



Bayes' theorem

- Memorize this formula

$$P(A | B) = \frac{P(B | A) P(A)}{P(B)},$$

where A and B are events and $P(B) \neq 0$.

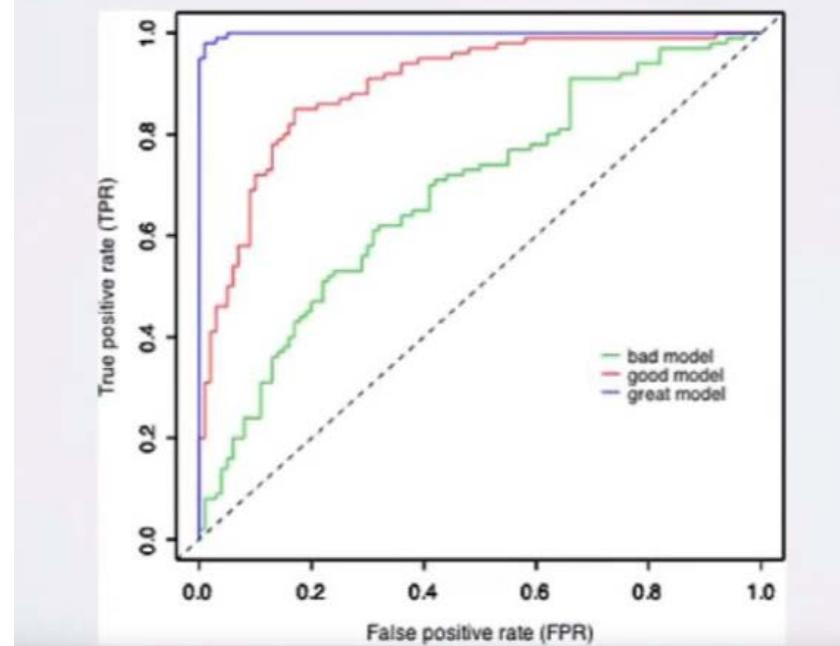
- $P(A)$ and $P(B)$ are the probabilities of observing A and B without regard to each other.
- $P(A | B)$, a conditional probability, is the probability of observing event A given that B is true.
- $P(B | A)$ is the probability of observing event B given that A is true.

- Understand terms

- Bayes' theorem describes the probability of an event
- based on prior knowledge of conditions that might be related to the event
- For example, if cancer is related to age, age should be included as input parameter

Machine learning evaluation metrics

- ROC curves
- cross-validation
- metrics for classification



Data Science job fact #1

Most of Data Science is fine-tuning models to get the highest performance possible

REALITY:



You are going to spend most of your time cleaning/merging data

Data Science job fact #2

Big Data is EVERYWHERE! You will need Hadoop and Spark all the time to solve every problem!

REALITY:



With exceptions, most problems can be handled on a single machine

Data Science job fact #3

Deep Learning solves EVERYTHING! Other methods are obsolete.

REALITY:



You probably don't need it, unless you are working with images and want to maximize performance

DS compared to ML engineer

How is a Machine Learning Engineer different from a Data Scientist?

Data Scientist

- Trained to be strong in Data
- R, Python, MATLAB
- Data treatment
- Evaluate ML algorithm
- Evaluate ML module

ML Engineer

- Trained to be strong in Coding
- C++, Java, C#
- Coding
- Change algorithm to code
- Create ML module

More information

More information on Data Science

Doing Data Science by Cathy O'Neil, Rachel Schutt:
Chapter 1. Introduction: What Is Data Science?

<https://www.safaribooksonline.com/library/view/doing-data-science/9781449363871/ch01.html>

DATA SCIENCE

BASIC



In this session

- The 5 questions data science answers
- Is your data ready for data science
- Ask a question you can answer with data
- Predict an answer with a simple model

The 5 questions data science answers

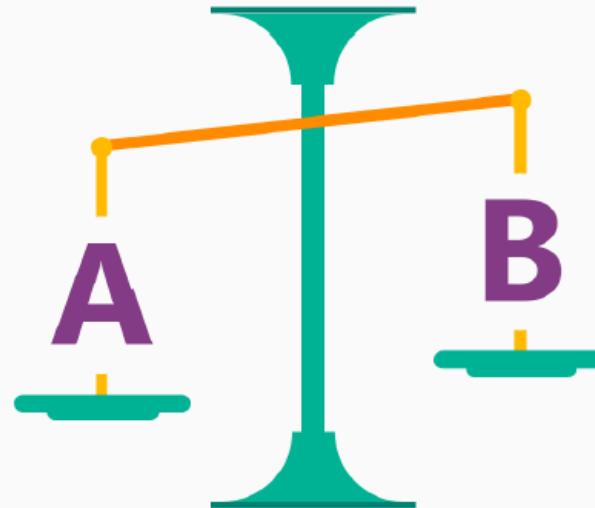
The 5 questions data science answers

- Is this A or B?
- Is this weird?
- How much – or – How many?
- How is this organized?
- What should I do next?

The 5 questions data science answers

Is this A or B?

Classification algorithms



- Will this tire fail in the next 1,000 miles: Yes or no?
- Which brings in more customers: a \$5 coupon or a 25% discount?
- Can also be more than two options: Is this A or B or C or D, etc.?
- Classification algorithms: helps choosing the most likely one.

The 5 questions data science answers

Is this weird?

Anomaly detection algorithms

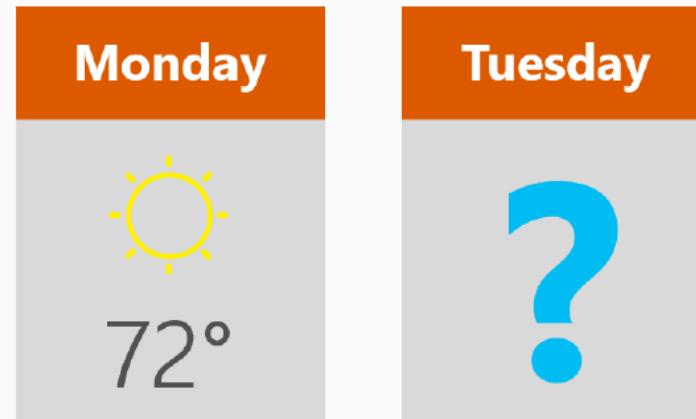


- Is this pressure gauge reading normal?
- Is this message from the internet typical?
- Credit card purchase pattern normal?
- Anomaly algorithms: Detect unexpected or unusual events or behaviors

The 5 questions data science answers

How much? How many?

Regression algorithms



- What will the temperature be next Tuesday?
- What will my fourth quarter sales be?
- Regression algorithms: Good for question that asks for a number

The 5 questions data science answers

How is this organized?

Clustering Algorithms



- Which viewers like the same types of movies?
- Which printer models fail the same way?
- Clustering algorithms: helps arranging data into groups
- Understanding how data is organized, helps predict behaviors and events.

The 5 questions data science answers

What should I do now?

Reinforcement Learning Algorithms



- Adjust the temperature or leave it where it is?
- At a yellow light, brake or accelerate?
- Keep vacuuming, or go back to the charging station?
- Reinforcement learning algorithms: the brains of rats and humans respond to punishment and rewards. learning from trial and error.

Is your data ready for data science

We need data that is:

- Relevant
- Connected
- Accurate
- Enough to work with

Is your data ready for data science
Relevant

Irrelevant Data

Price of milk (\$/gal)	Red Sox batting avg.	Blood alcohol content (%)
3.79	.304	.03
3.45	.320	.09
4.06	.259	.01
3.89	.298	.05
4.12	.332	.13
3.92	.270	.06
3.23	.294	.10

Relevant Data

Body mass (kg)	Margaritas	Blood alcohol content (%)
103	3	.03
67	5	.09
87	1	.01
52	2	.05
73	5	.13
79	3	.06
110	7	.10

- We need to know Blood alcohol content %
- Price of milk and Red Sox are irrelevant

Is your data ready for data science Connected

Disconnected Data			Connected Data		
Grill temp. (Fahrenheit)	Weight of beef patty (lb)	Burger rating (out of 10)	Grill temp. (Fahrenheit)	Weight of beef patty (lb)	Burger rating (out of 10)
	.33	8.2	575	.33	8.2
	.24	5.6	550	.24	5.6
550		7.8	550	.69	7.8
725	.45	9.4	725	.45	9.4
600		8.2	600	.57	8.2
625		6.8	625	.36	6.8
	.49	4.2	550	.49	4.2

- Quality of hamburgers
- But notice the gaps in the table on the left
- It's common to have holes like this

Is your data ready for data science

Accurate



- Top left: precise=yes/accurate=no
- Bottom left: precise=no/accurate=no
- Top right: precise=yes/accurate=yes
- Bottom right: precise=no/accurate=yes

Is your data ready for data science
Enough to work with



We need enough data to work with

1. Not enough data: can not make decision
2. Barely enough data: can make basic decision (Is it somewhere I might want to visit? It looks bright, that looks like clean water – yes, that's where I'm going on vacation.)
3. Enough data: can make detailed decision (Now I can look at the three hotels on the left bank. You know, I really like the architectural features of the one in the foreground. I'll stay there, on the third floor.)

Ask a question you can answer with data

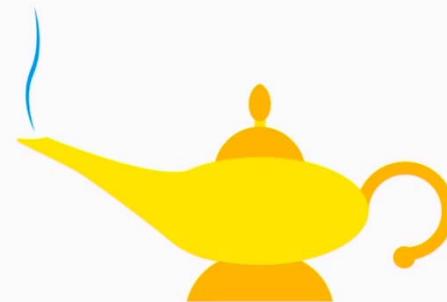
Ask a question you can answer with data

- Sharp question is the Key
- Know Target Data
- Reword the question

Ask a question you can answer with data

Sharp question is the Key

Ask a sharp question



- Asking a sharp question is the most important
- ML is a mischievous genie
- "What's going to happen with my stock?", the genie might answer, "The price will change"
- "What will my stock's sale price be next week?", the genie can't help but give you a specific answer and predict a sale price

Ask a question you can answer with data

Know Target Data

Examples of the answer: Target data

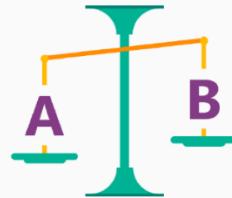


- Target data = what we are trying to predict
- Must have examples of the target in data.
- Question = "What will my stock's sale price be next week?" Target = stock price history.
- Question = "Which car in my fleet is going to fail first?" Target = previous failures data.

Ask a question you can answer with data

Reword the question

Reformulate your question



- Question dictates the algorithm
- "Is this data point A or B?" = classification
- "How much?" or "How many?" = regression
- "Which news story is the most interesting to this reader?"
- Algorithm = classification A or B or C or D; difficult
- Reword = "How interesting is each story on this list to this reader?"
- Give each article a numerical score
- Identify the highest-scoring article; easy
- Above example change classification question into a regression question

Predict an answer with a simple model

Predict an answer with a simple model

- Collect relevant, accurate, connected, enough data
- Ask a sharp question
- Plot the existing data
- Draw the model through the data points
- Use the model to find the answer
- Create a confidence interval

Predict an answer with a simple model

Collect relevant, accurate, connected, enough data

- I want to know how much 1.35 carat diamond will cost
- Go to jewelry store
- Write down the price of all of the diamonds
- List has two columns
- Each column has a different attribute
- Weight in carats and price
- Each row is a single data point
- Data that represents a single diamond.
- This is a small data set; a table

<u>Carats</u>	<u>Price</u>
1.01	7,366
.49	985
.31	544
1.51	9,140
.37	493
.73	3,011
1.53	11,413
.56	1,814
.41	876
.74	2,690
.63	1,190
.6	4,172
2.06	11,764
1.1	4,682
1.31	6,171

Predict an answer with a simple model

This data set meets our criteria for quality:

- Relevant: weight is definitely related to price
- Accurate: we double-checked the prices that we write down
- Connected: there are no blank spaces in either of these columns
- Enough data: to answer our question



Predict an answer with a simple model

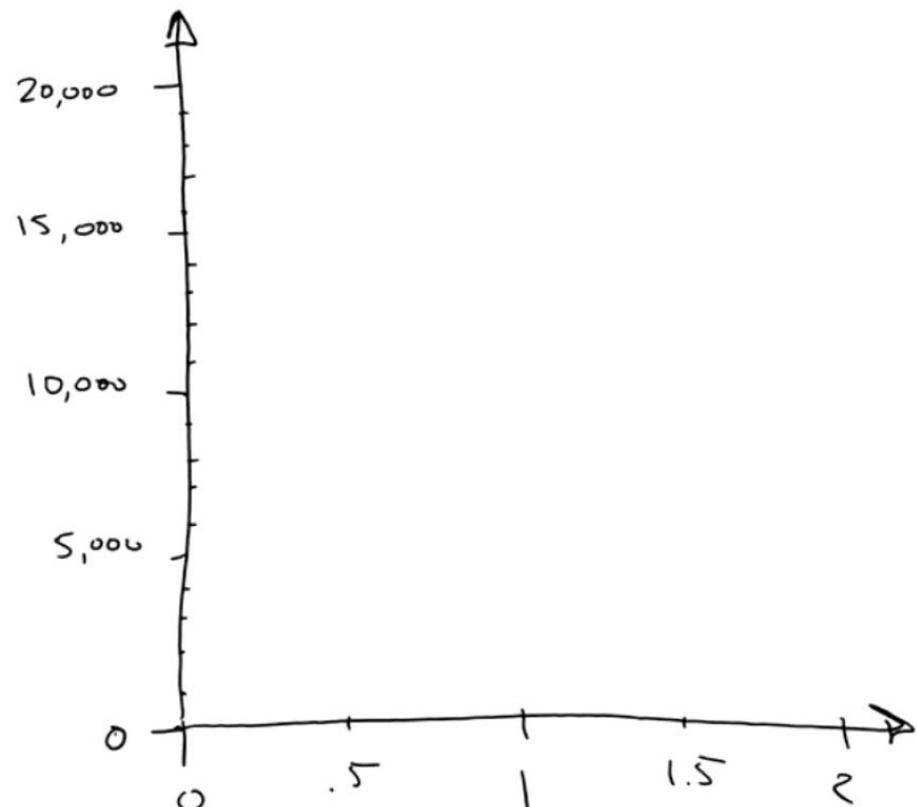
Ask a sharp question

- How much will it cost to buy a 1.35 carat diamond?
- Our list doesn't have a 1.35 carat diamond
- Use the rest of our data to get an answer to the question

Predict an answer with a simple model

Draw axis

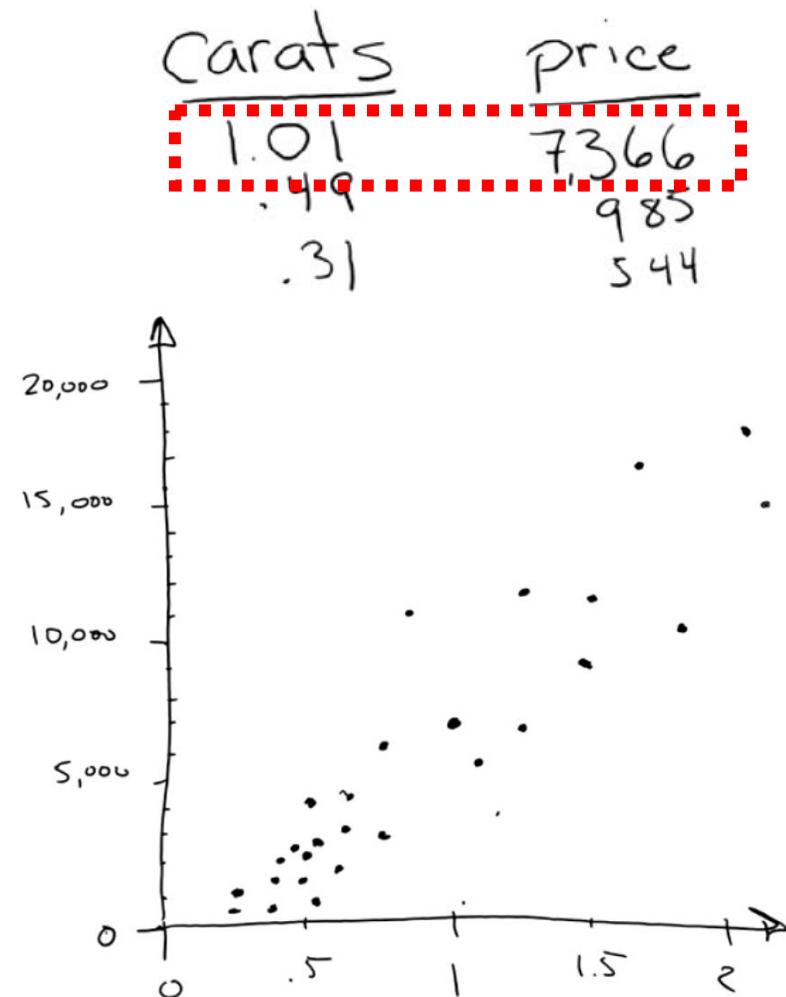
- Draw a horizontal number line, called an axis, to chart the weights
- The range of the weights is 0 to 2
- Line covers that range and put ticks for each half carat
- Draw a vertical axis to record the price and connect it to the horizontal weight axis
- This will be in units of dollars
- Now we have a set of coordinate axes.



Predict an answer with a simple model

Plot the existing data

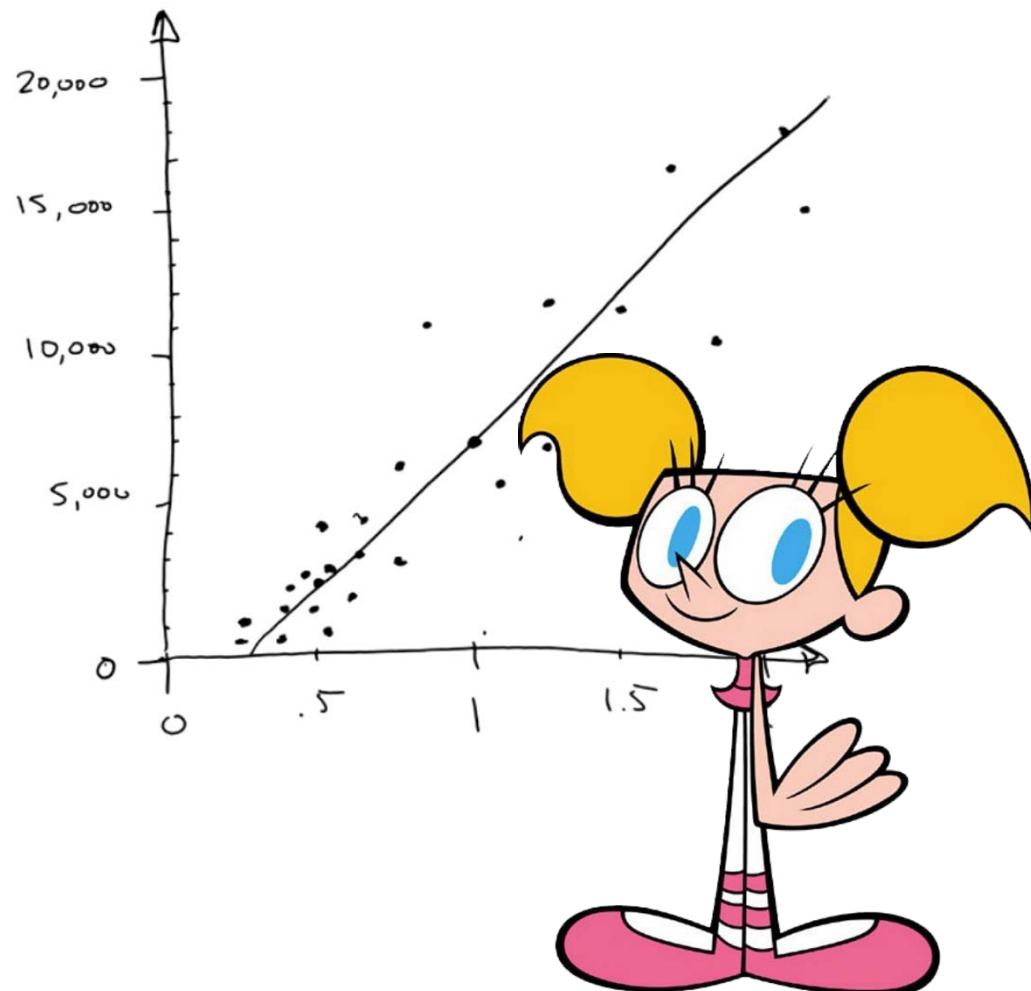
- Make a scatter plot
- Great way to visualize numerical data sets
- For the first data point, we eyeball a vertical line at 1.01 carats. Then, we eyeball a horizontal line at \$7,366. Where they meet, we draw a dot
- This represents our first diamond.
- Now we go through each diamond on this list and do the same thing.
- We get a bunch of dots, one for each diamond



Predict an answer with a simple model

Draw the model through the data points

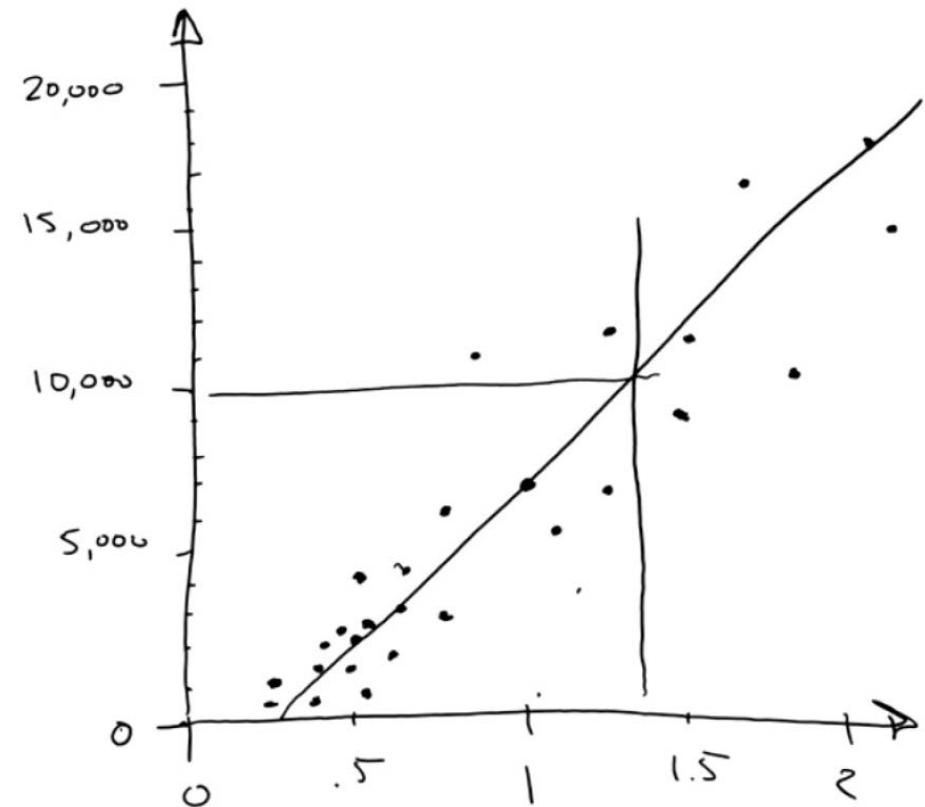
- Look at the dots and squint, the collection looks like a fat, fuzzy line
- Draw a straight line through it
- This a model
- Model = cartoon
- The cartoon is wrong
- But, it's a useful simplification
- The line doesn't go through all the data points.
- It has some noise or variance
- But, it's a useful simplification
- Question = How much? regression
- we're using a straight line, linear regression



Predict an answer with a simple model

Use the model to find the answer

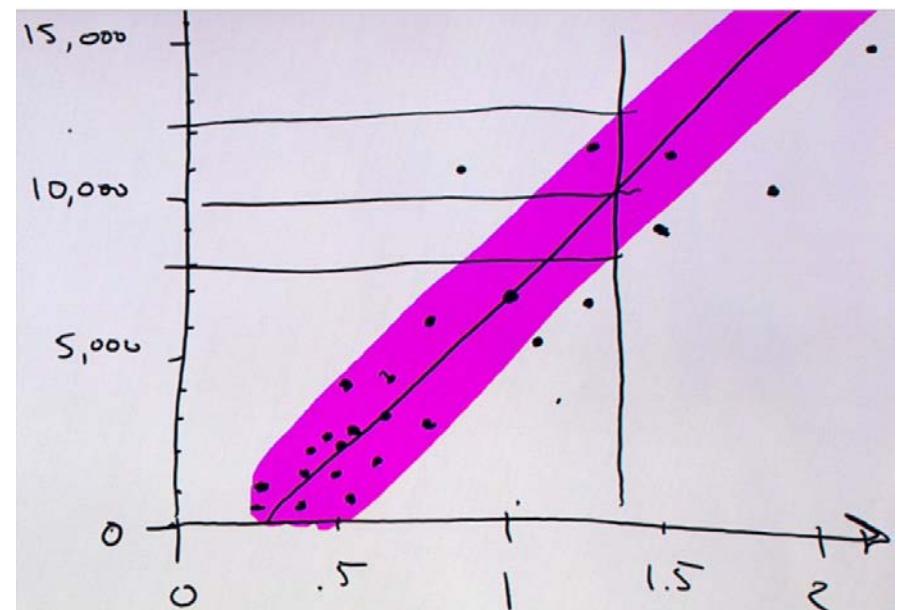
- How much will a 1.35 carat diamond cost?
- Look at 1.35 carats
- Draw a vertical line
- Draw a horizontal line to the dollar axis
- It hits right at 10,000
- Answer = about \$10,000



Predict an answer with a simple model

Create a confidence interval

- How precise this prediction is?
- Is it a lot higher or lower?
- Draw an envelope around the regression line
- that includes most of the dots.
- This envelope is called our confidence interval
- We're pretty confident that prices fall within this envelope, because in the past most of them have.
- We can draw two more horizontal lines from where the 1.35 carat line crosses the top and the bottom of that envelope.
- The price of a 1.35 carat diamond is about \$10,000 - but it might be as low as \$8,000 and it might be as high as \$12,000



Predict an answer with a simple model

We're done, with no math or computers!!

- We did what data scientists get paid to do, and we did it just by drawing:
- We asked a question that we could answer with data
- We built a model using linear regression
- We made a prediction, complete with a confidence interval

And we didn't use math or computers to do it.



Predict an answer with a simple model

Now if we'd had more information, like...

- the cut of the diamond
- color variations (how close the diamond is to being white)
- the number of inclusions in the diamond

...then we would have had more columns. In that case, math becomes helpful. If you have more than two columns, it's hard to draw dots on paper. The math lets you fit that line or that plane to your data very nicely.

Also, if instead of just a handful of diamonds, we had two thousand or two million, then you can do that work much faster with a computer.

$$a = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2}$$

$$b = \frac{1}{n} (\sum_{i=1}^n y_i - a \sum_{i=1}^n x_i)$$

More information

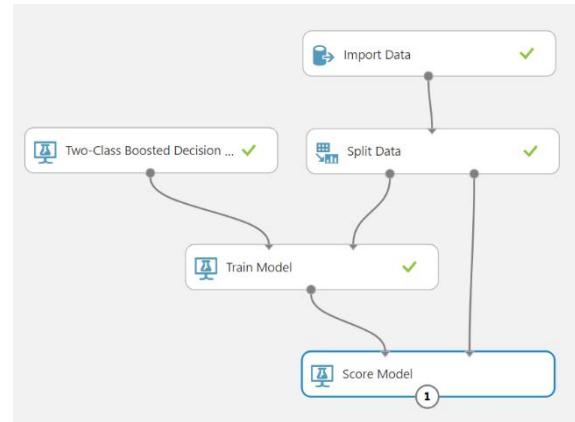
More information on Data Science Basic

An introduction to Data Science: Jeffrey Stanton

https://ischool.syr.edu/media/documents/2012/3/DataScienceBook1_1.pdf



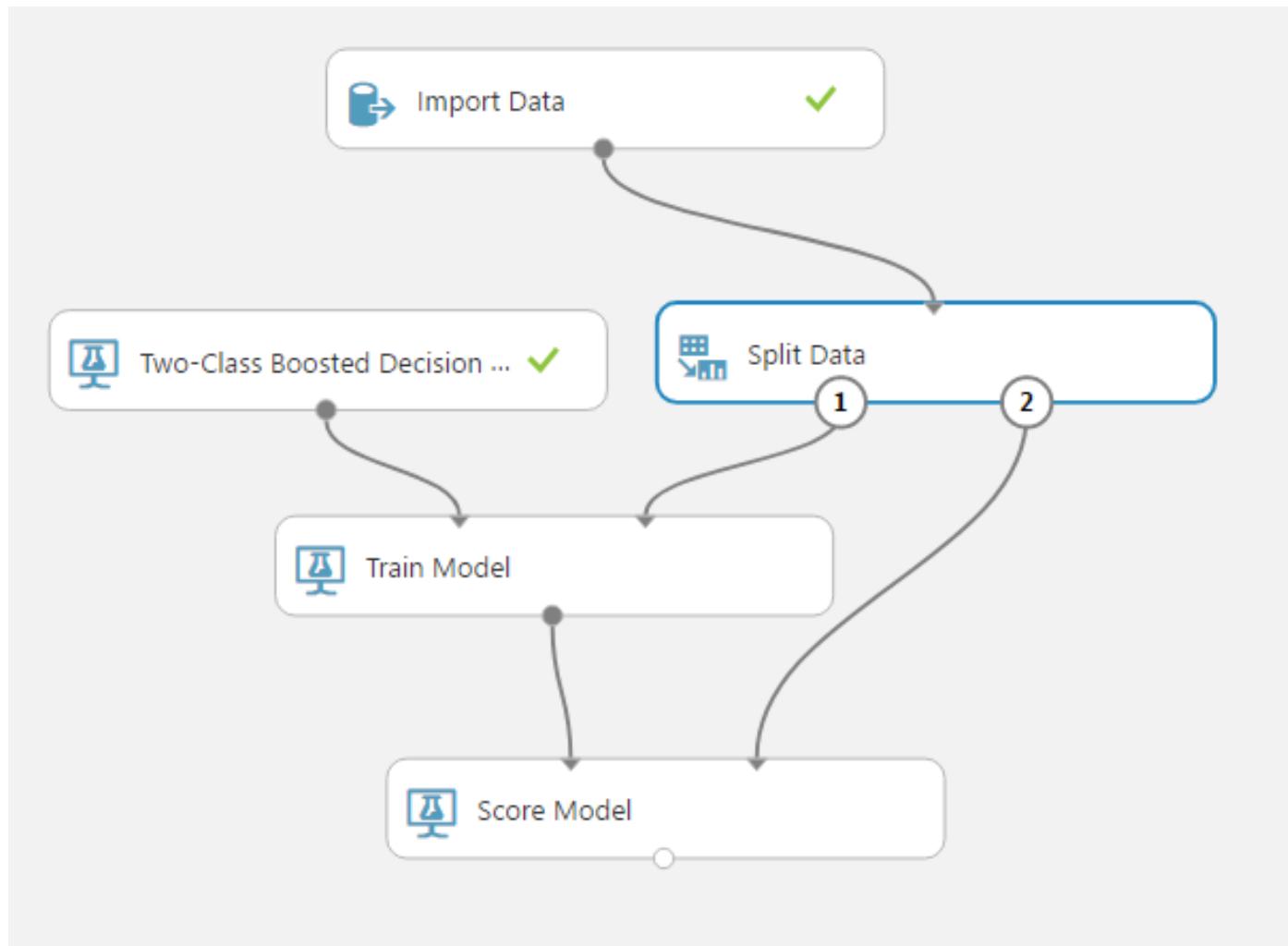
FIRST EXPERIMENT



In this session

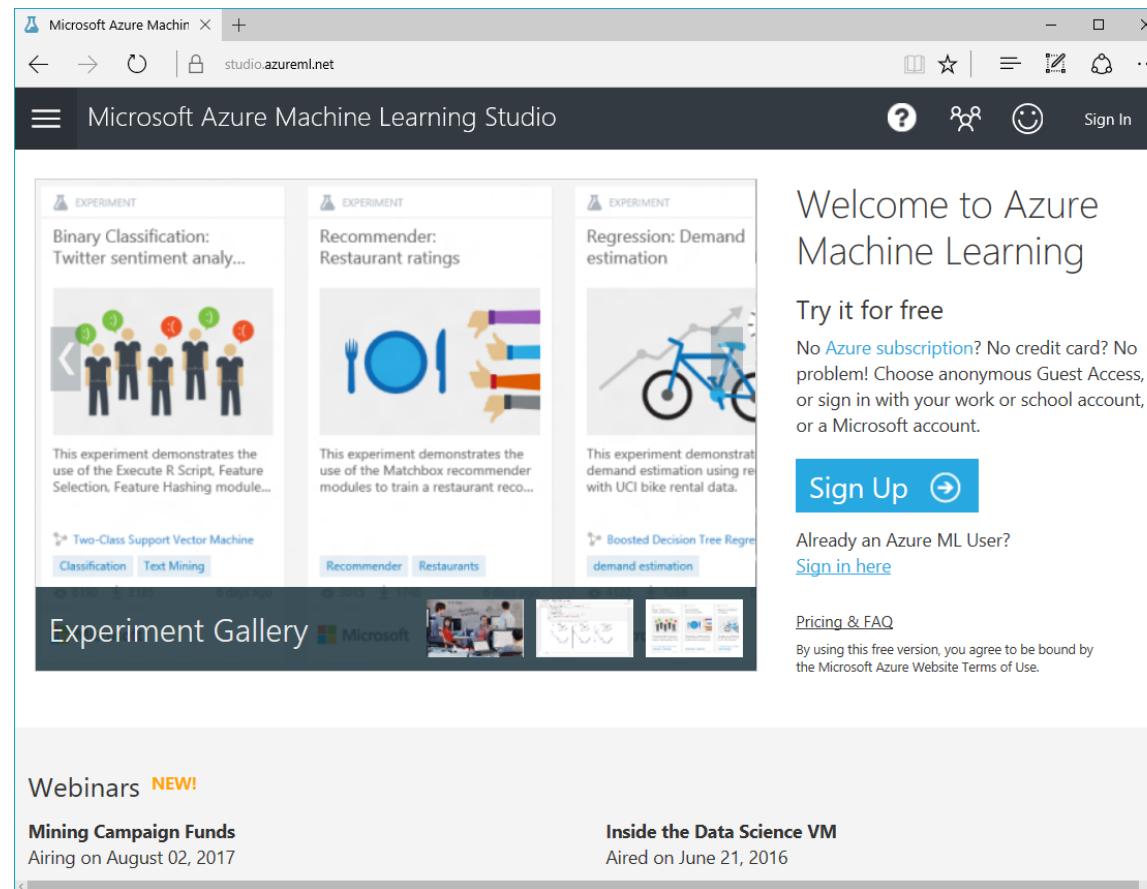
- Sing Up FREE Azure ML Studio Subscription
- Create Azure ML Studio workspace
- Train, Test, Evaluate for Binary Classification
- Import census income dataset
- Create a new Azure Machine Learning experiment
- Train and evaluate a prediction model
- Type of datasets

First experiment model



Sing Up FREE Azure ML Studio Subscription

<https://studio.azureml.net/>



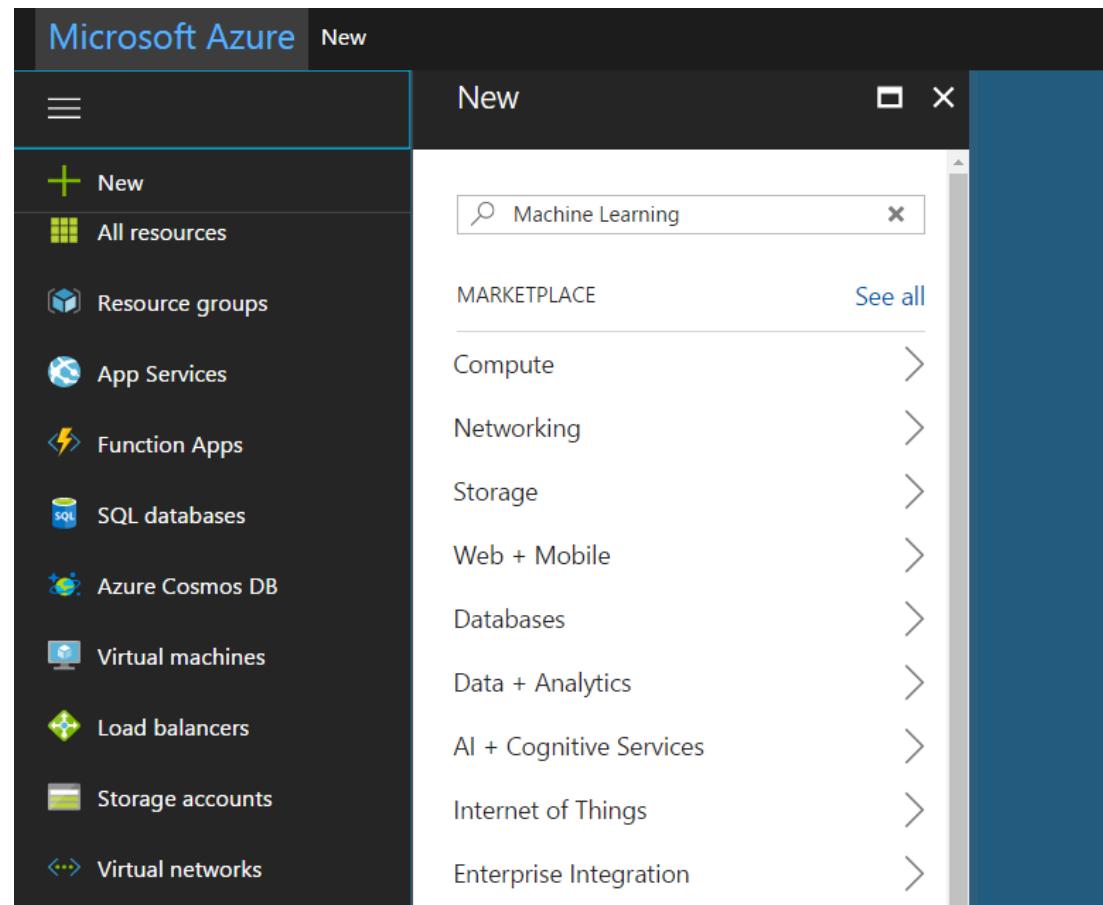
Sing Up FREE Azure ML Studio Subscription

Free Workspace -> sign up here

Quick Evaluation	Most Popular	Enterprise Grade
<p>Guest Workspace</p> <p>8-hour trial</p> <p>No sign-in required.</p> <p>Enter</p>	<p>Free Workspace</p> <p>\$0/month</p> <p>Don't already have a Microsoft account? Simply sign up here.</p> <p>Sign In</p>	<p>Standard Workspace</p> <p>\$9.99/month</p> <p><small>Azure subscription required Other charges may apply. Read more.</small></p> <p>Create Workspace</p>
<ul style="list-style-type: none">▪ No hassle instant access▪ Stock sample datasets▪ ML models built in minutes▪ Full range of ML algorithms	<ul style="list-style-type: none">▪ Free access that never expires▪ 10 GB storage on us▪ R and Python scripts support▪ Predictive web services	<ul style="list-style-type: none">▪ Full SLA Support▪ Bring your own Azure storage▪ Parallel graph execution▪ Elastic Web Service endpoints

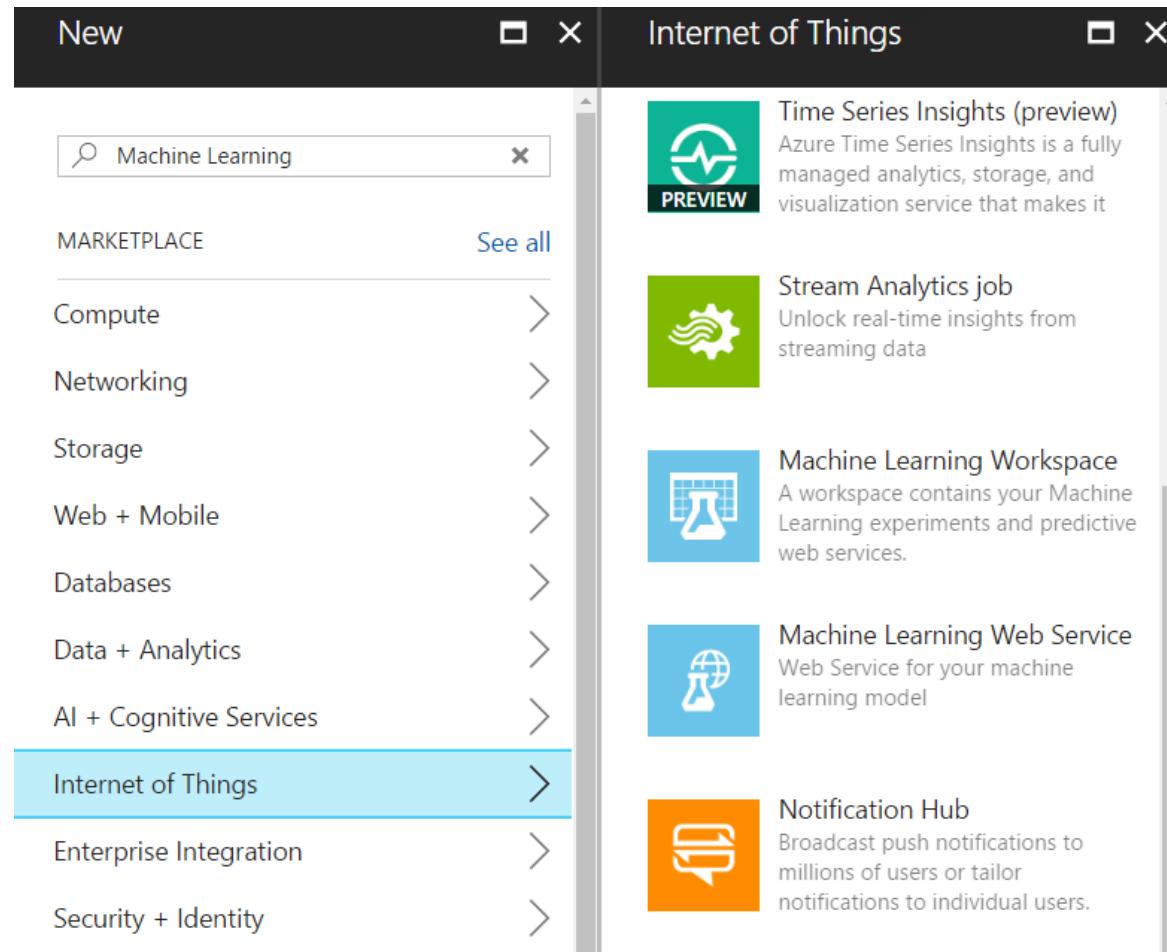
Create Azure ML Studio workspace

1. Go to the Azure portal <https://portal.azure.com>
2. Click +New



Create Azure ML Studio workspace

3. Select Internet of Things, click Machine Learning Workspace, then click Create



Create Azure ML Studio workspace

4. Workspace name = ws1
5. Subscription = default
6. Resource group = Create new: rs1
7. Location = Southeast Asia
8. Storage account = Create new: names1
9. Workspace pricing tier = Standard
10. Web service plan = Create new: ws1Plan

Machine Learning workspace X

Machine Learning workspace

* Workspace name: ws1

* Subscription: Loy2017a

* Resource group: Create new Use existing
r1

* Location: Southeast Asia

* Storage account: Create new Use existing
ws123

Workspace pricing tier: Standard

* Web service plan: Create new Use existing
ws1Plan

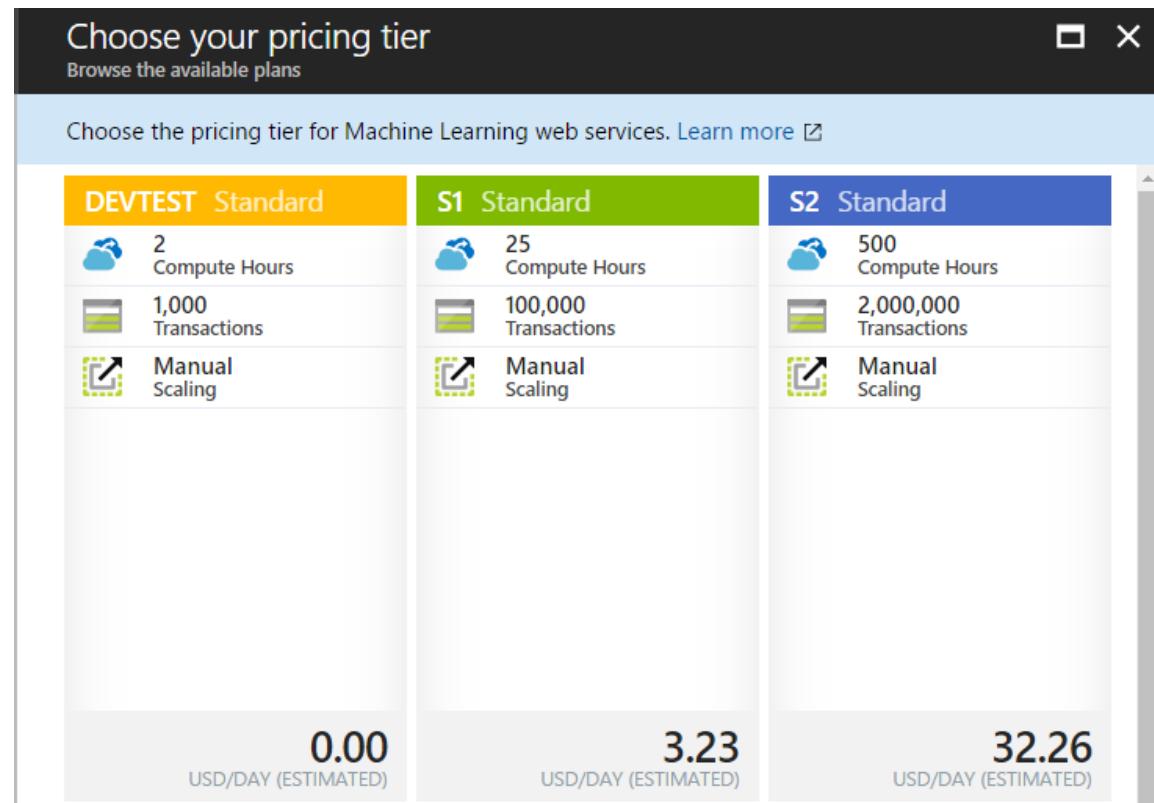
* Web service plan pricing tier: No pricing tier selected

Pin to dashboard

Create Automation options

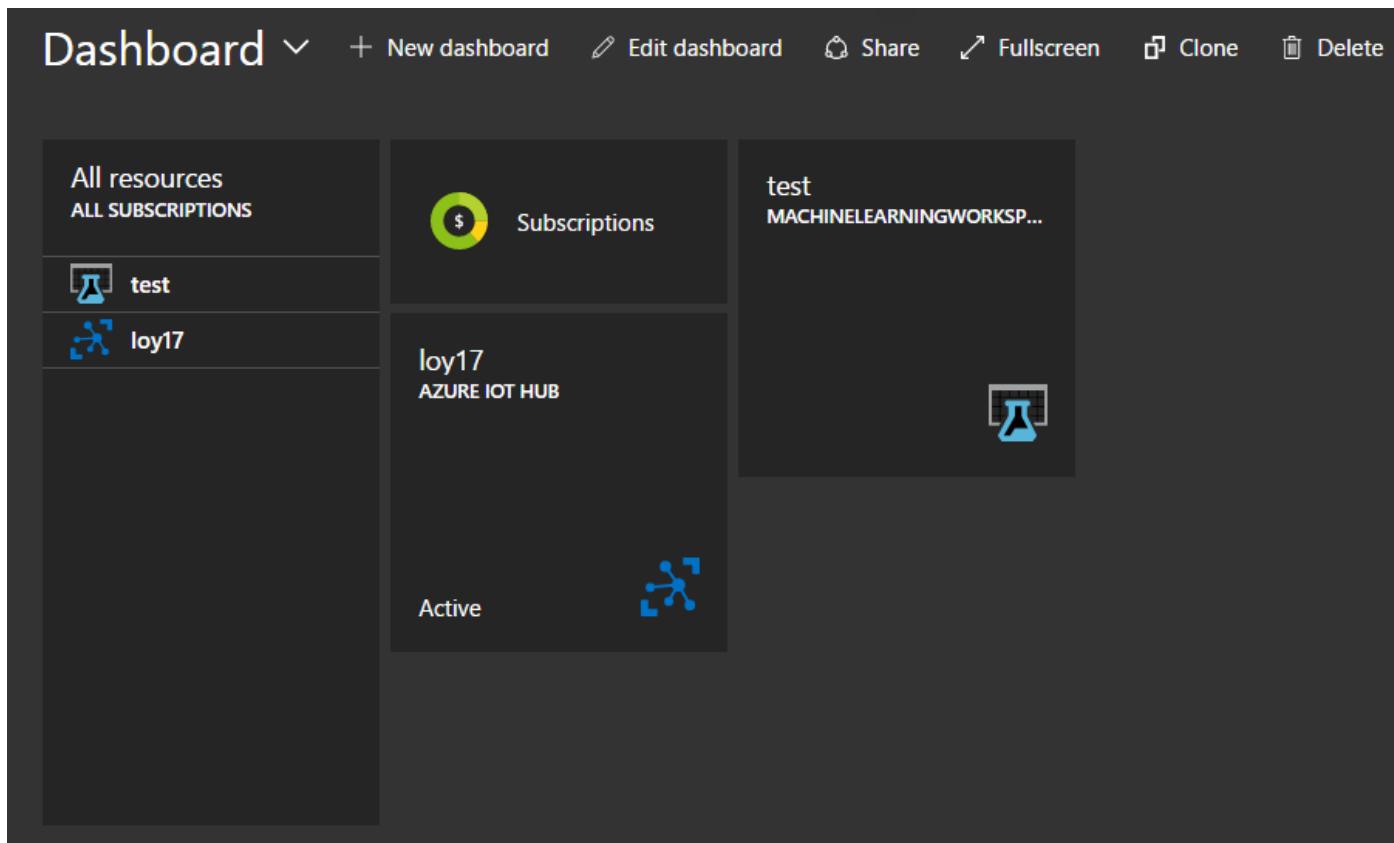
Create Azure ML Studio workspace

11. Click No pricing tier selected
12. Click DEVTEST
13. Click Pin to dashboard
14. Click Create



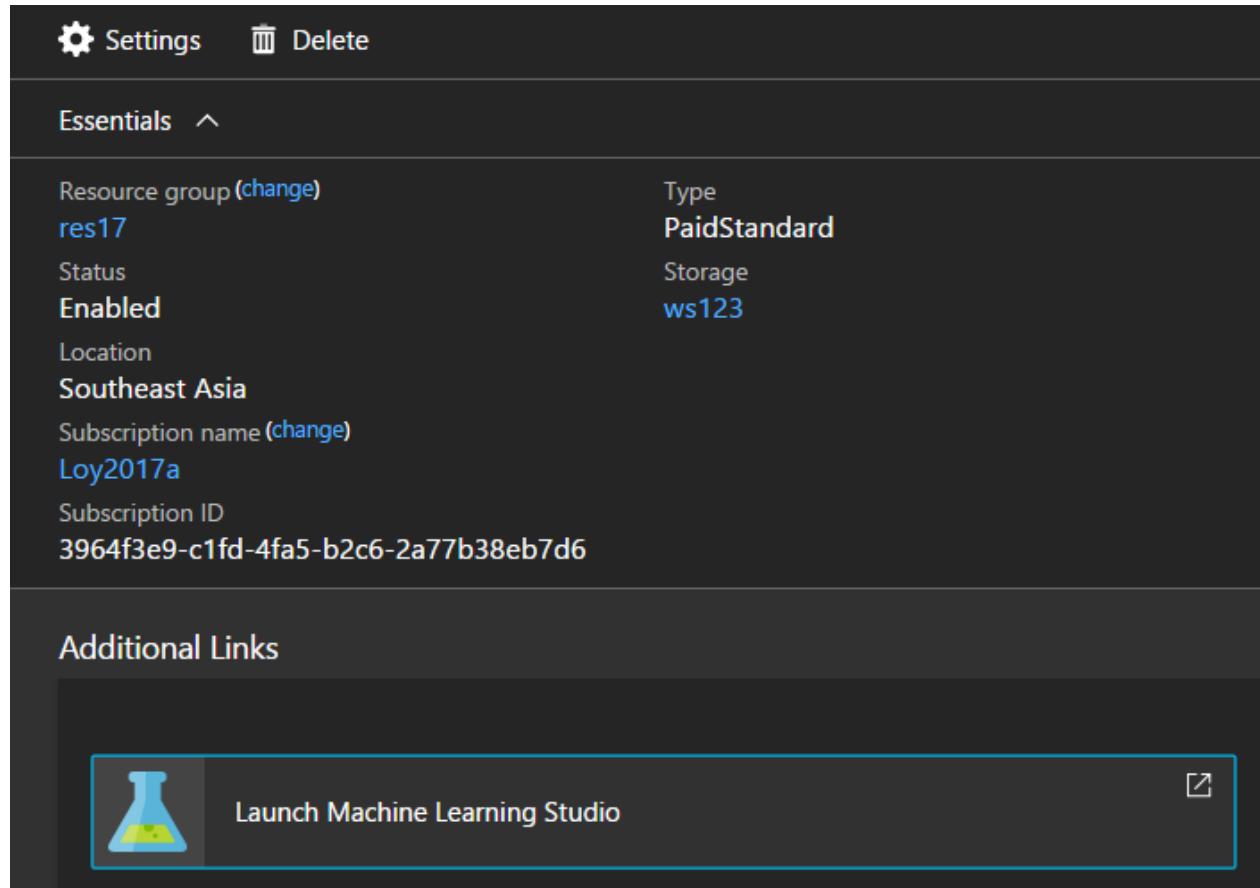
Create Azure ML Studio workspace

15. Click at Machine Learning workgroup on dashboard



Create Azure ML Studio workspace

16. Click Launch Machine Learning Studio



Create Azure ML Studio workspace

Blank, new ML Studio workspace

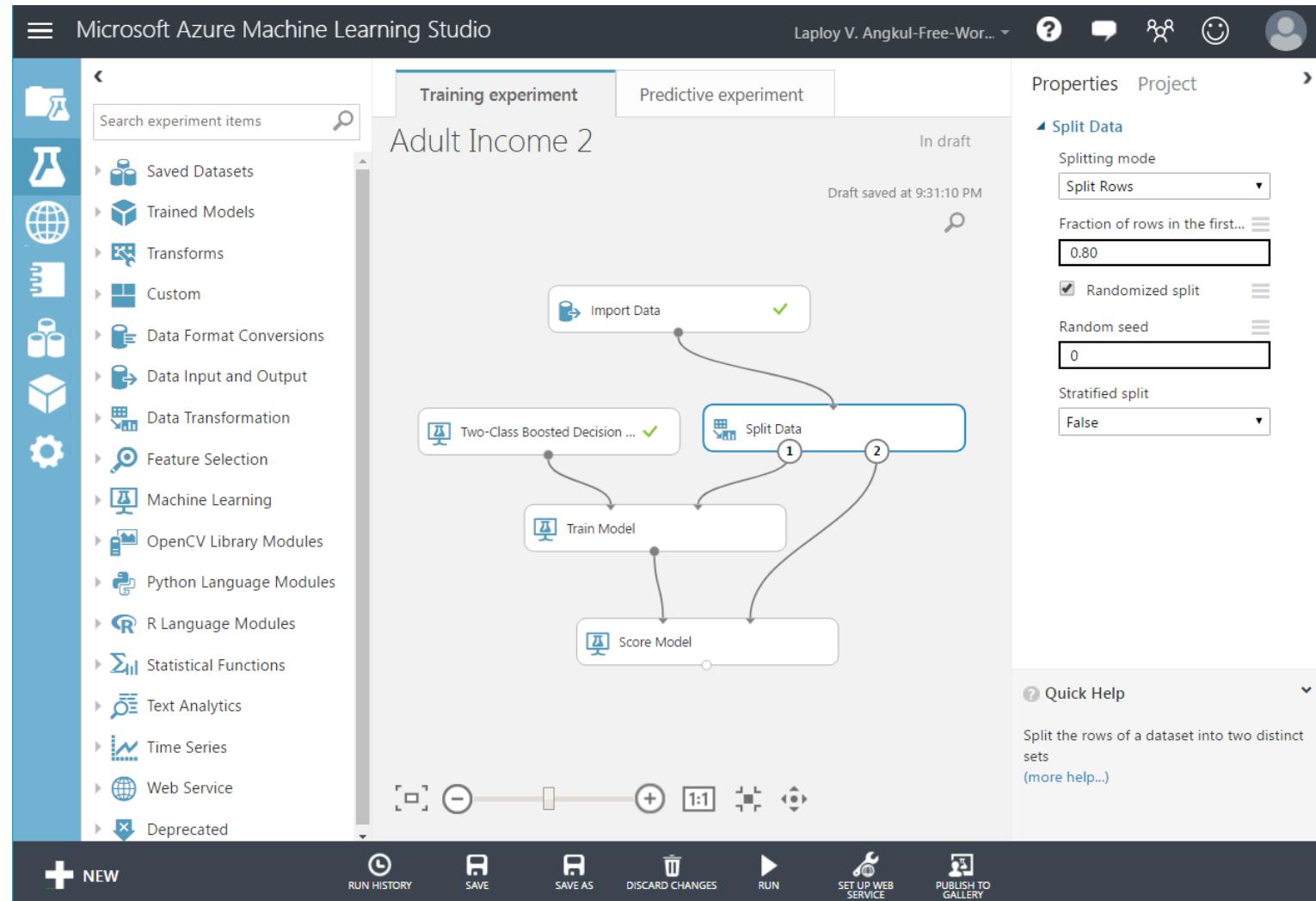
The screenshot shows the Microsoft Azure Machine Learning Studio interface. The top navigation bar displays the title "Microsoft Azure Machine Learning Studio" and the user "Laploy V. Angkul-Free-Wor...". The left sidebar menu includes options like PROJECTS, EXPERIMENTS (which is selected), WEB SERVICES, NOTEBOOKS, DATASETS, TRAINED MODELS, and SETTINGS. The main workspace is titled "experiments" and contains tabs for "MY EXPERIMENTS" and "SAMPLES". A table header with columns NAME, AUTHOR, STATUS, LAST ED., PROJ..., and a delete icon is visible. The message "No experiments found" is displayed. On the right side, a status bar indicates "0 items selected". At the bottom, there are buttons for "+ NEW", "DELETE", and "ADD TO PROJECT".

Train, Test, Evaluate for Binary Classification

Predicting whether a person's income exceeds \$50,000 per year based on his demographics or census data

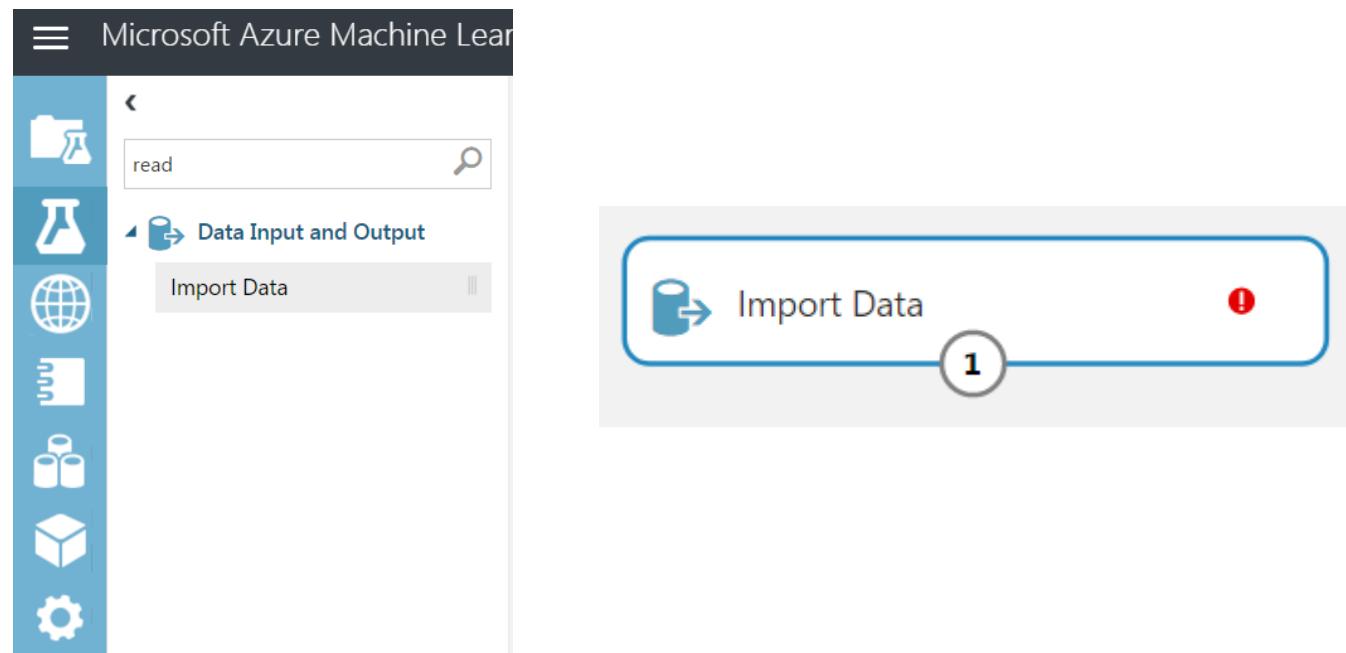
1. Download, prepare, and upload a census income dataset.
2. Create a new Azure Machine Learning experiment.
3. Train and evaluate a prediction model.

The overall workflow of the experiment



Train, Test, Evaluate for Binary Classification

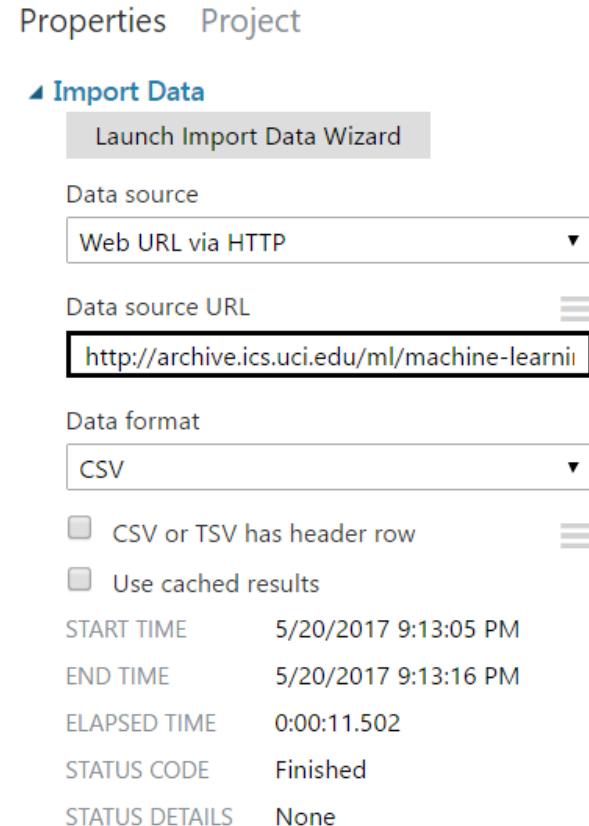
- Create New blank experiment. Name = Adult Income 1
- Click Data Input and Output
- Drag & drop Import Data



Train, Test, Evaluate for Binary Classification

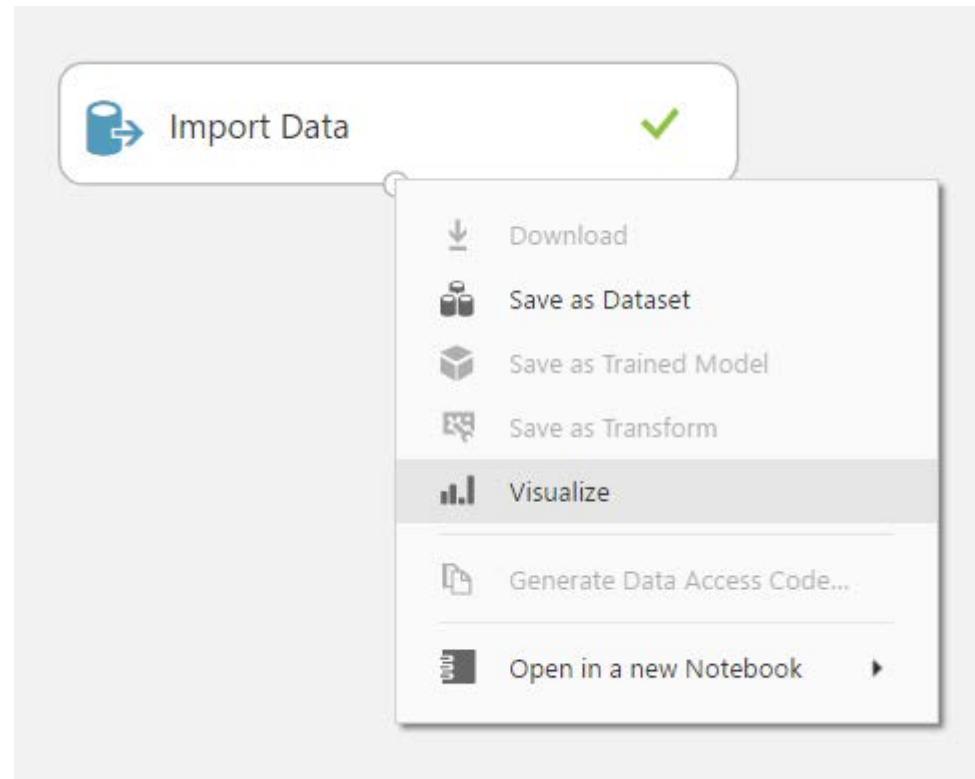
Configure Import data module:

- Data source = Web URL via HTTP
- Data source URL = `http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data`
- Data format = CSV
- Run experiment



Train, Test, Evaluate for Binary Classification

- Right click at the output of Import Data
- Click Visualize



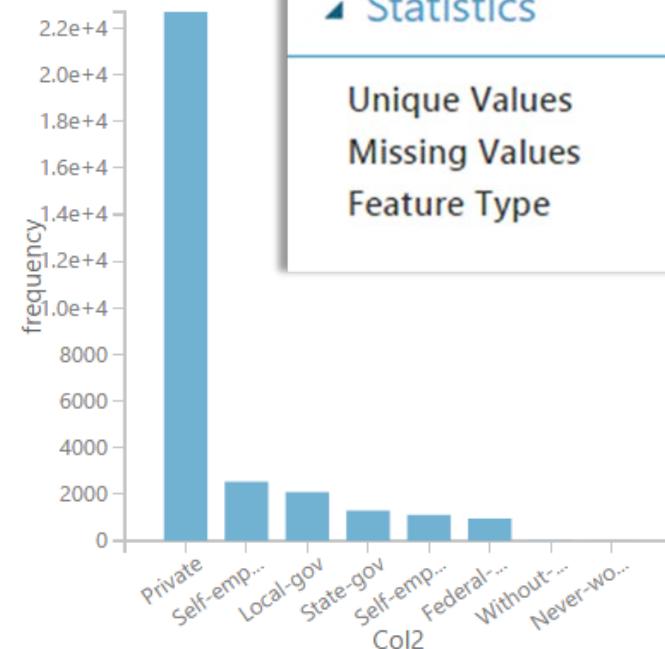
Train, Test, Evaluate for Binary Classification

- Click on Col2
- Look at Statistics and Histogram

Adult Income > Import Data > Results dataset

rows	columns					
32562	15					
view as						
	Col1	Col2	Col3	Col4	Col5	Col6
39		State-gov	77516	Bachelors	13	Never-married
50		Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse
38		Private	215646	HS-grad	9	Divorced
53		Private	234721	11th	7	Married-civ-spouse
28		Private	338409	Bachelors	13	Married-civ-spouse
37		Private	284582	Masters	14	Married-civ-spouse

Col2
Histogram



Statistics	
Unique Values	8
Missing Values	1837
Feature Type	String Feature

Train, Test, Evaluate for Binary Classification

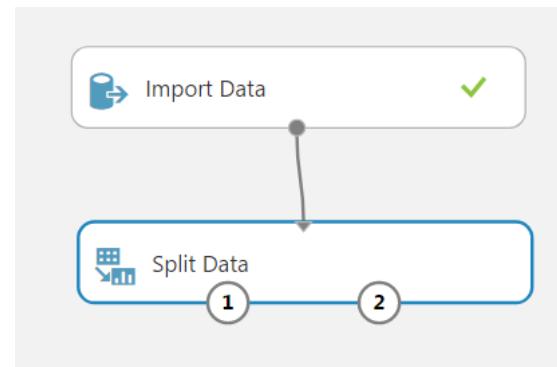
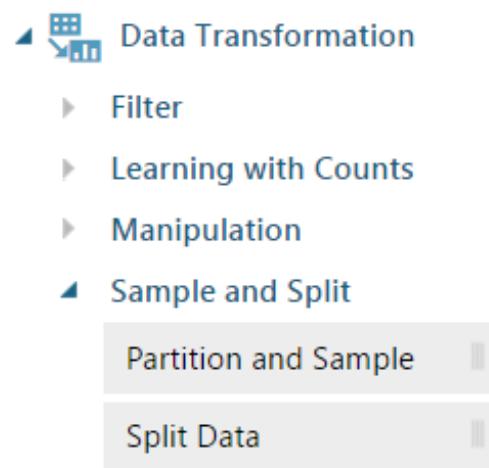
Split up the dataset

- Training data This grouping is used for creating our new predictive model based on the inherent patterns found in the historical data via the ML algorithm we use for the solution.
- Validation data This grouping is used for testing the new predictive model against known outcomes to determine accuracy and probabilities.

Train, Test, Evaluate for Binary Classification

Add Split Data:

- Click Data Transformation
- Click Sample and Split
- Drag & drop Split Data module into canvas
- Connect Import Data to Split Data
- Set properties Fraction of row to 0.80



Properties Project

▲ Split Data

Splitting mode: Split Rows

Fraction of rows in the first...: 0.80

Randomized split

Random seed: 0

Stratified split: False

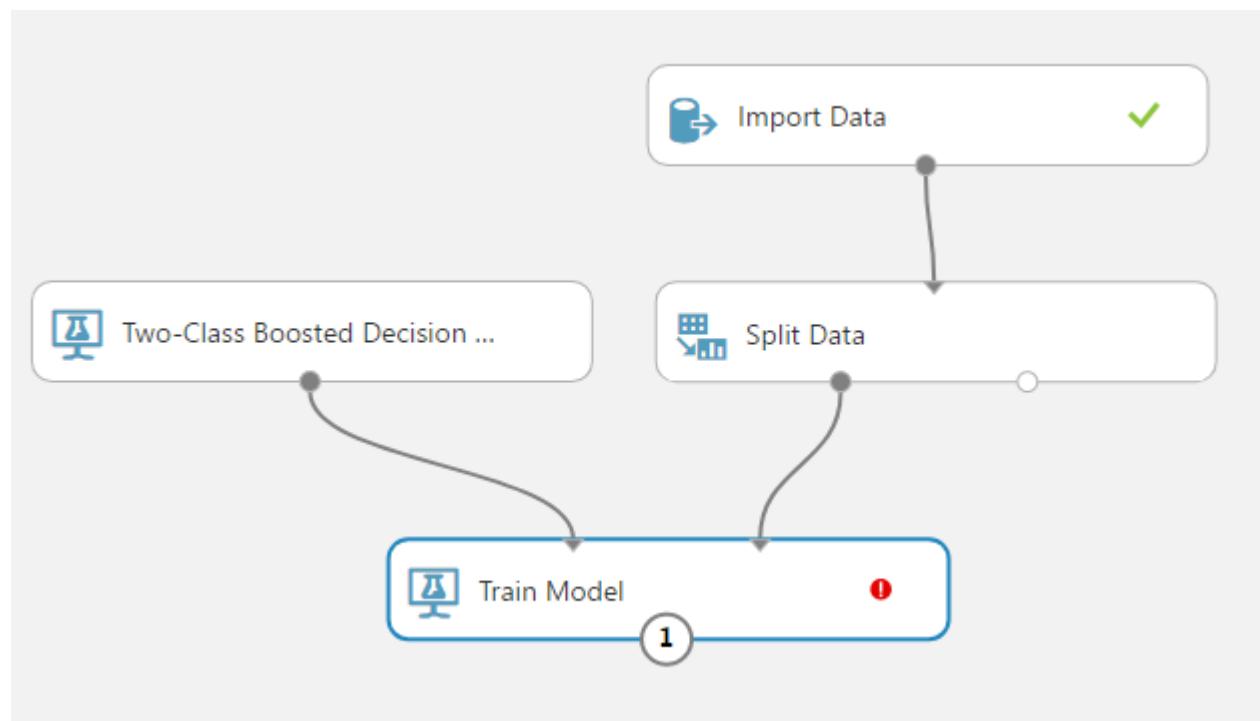
This screenshot shows the properties pane for the 'Split Data' module. It is set to 'Split Rows' mode with 80% of the rows in the first output. The 'Randomized split' checkbox is checked. A random seed of 0 is specified, and the 'Stratified split' option is disabled.

Train, Test, Evaluate for Binary Classification

Add Two-Class Boosted Decision Tree and Train Model

Connect Two-Class Boosted Decision Tree to Train Model

Connect Split Data to Train Model



Two-Class Boosted Decision T..

Create trainer mode

Single Parameter ▾

Maximum number of le...

20

Minimum number of sa...

10

Learning rate

0.2

Number of trees constru...

100

Random number seed

Allow unknown cate...

Train, Test, Evaluate for Binary Classification

Click Train Model

Click Launch column selector

Include col15

Click 

Save

Run



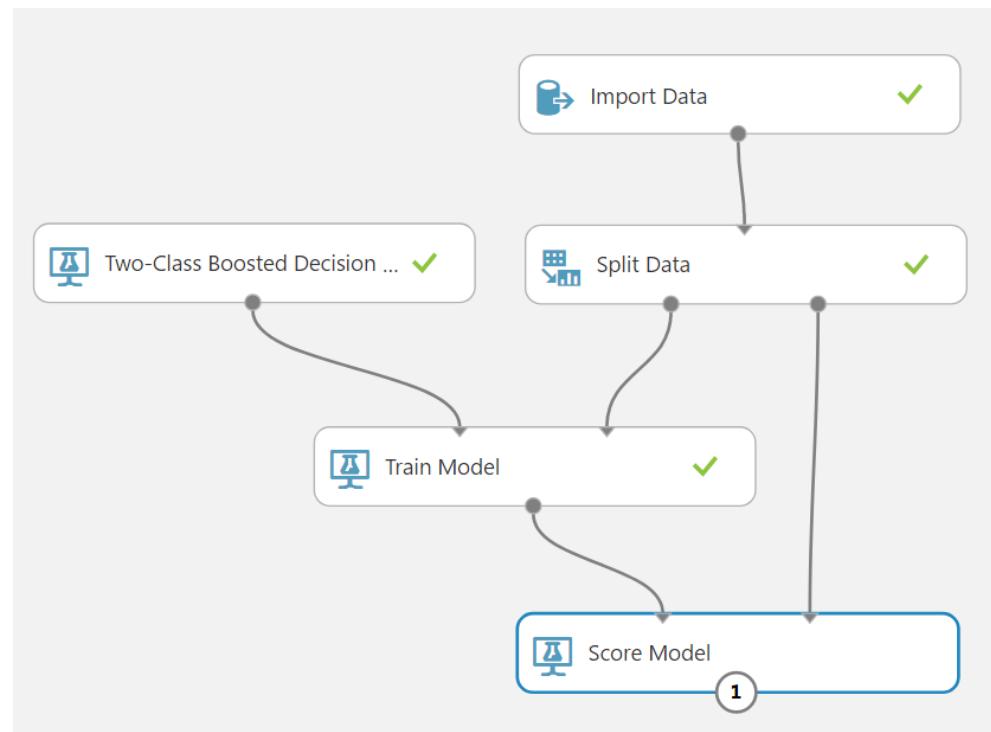
Score the model

Score the model:

Add Score Model to canvas

Connect Score Model to Train and Split model

Run



Visualize the model results

Col11	Col12	Col13	Col14	Col15	Scored Labels	Scored Probabilities
					<=50K	0.425173
0	0	50	United-States	<=50K	<=50K	0.425173
0	0	40	Puerto-Rico	<=50K	<=50K	0.008254
0	0	35	United-States	<=50K	<=50K	0.002206

Visualize the model results:

Visualize output of Score Model

Scored Labels This column denotes the model's prediction for this row of the dataset.

Scored Probabilities This column denotes the numerical probability (or the likelihood) of whether the income level for this row exceeds \$50,000.

Type of datasets

Training set

- A set of examples used for learning
- Where the answer value is known.

Validation set

- A set of examples data
- Used to tune the architecture of a classifier
- And estimate the error

Test set

- Use to test the performances of a classifier
- Never used during the training process
- Give estimate of error

More Information

Two-Class Boosted Decision Tree

<https://msdn.microsoft.com/en-us/library/azure/dn906025.aspx>

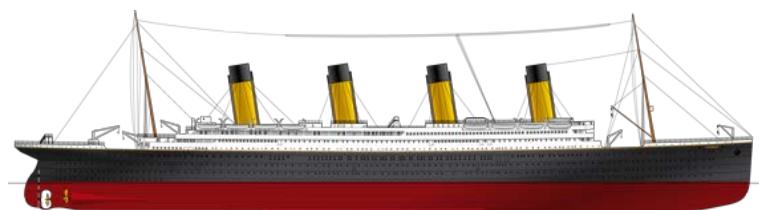
Score Model

<https://msdn.microsoft.com/en-us/library/azure/dn905995.aspx>

Published Experiment

<https://gallery.cortanaintelligence.com/Experiment/Adult-Income-1>

BUILDING A CLASSIFICATION MODEL



In this session

- Download Data set
- Data Dictionary
- View data set in Microsoft Excel
- Import Data set
- Create New Experiment
- Prepare Data
- Drop the columns
- Make categorical values
- Replace missing value with median
- Drop rows with missing data
- Create Label
- Split data
- Select Algorithm
- Train
- Score
- Create web service
- Test web service

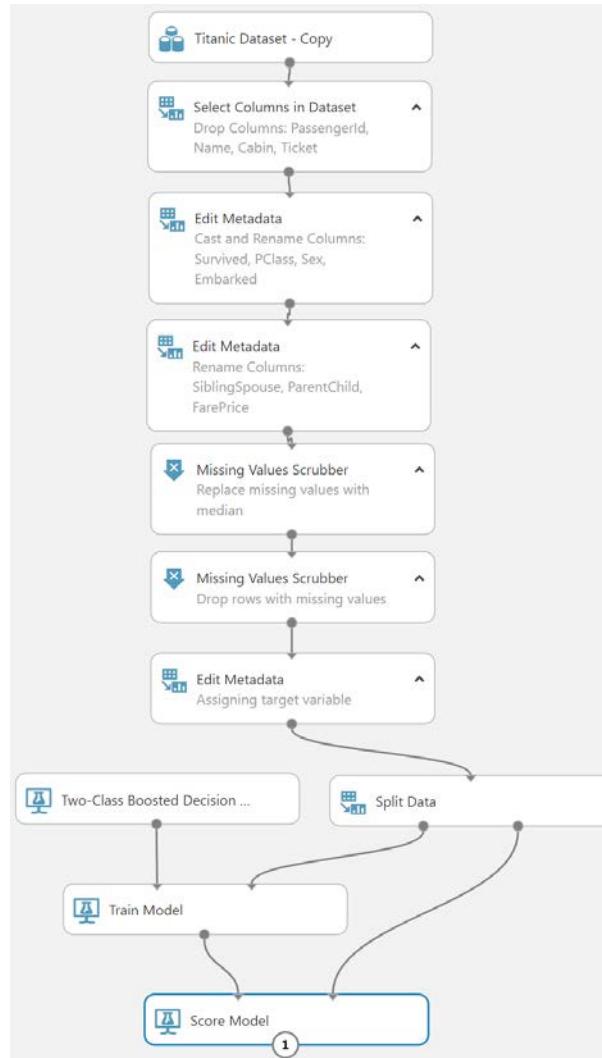
What to do

What to do;

- Create experiment
- Create Classification model
- Using Azure ML.
- Using the Titanic passenger data set
- Build a model for predicting the survival of a given passenger.



ML model when finished



AML model development step

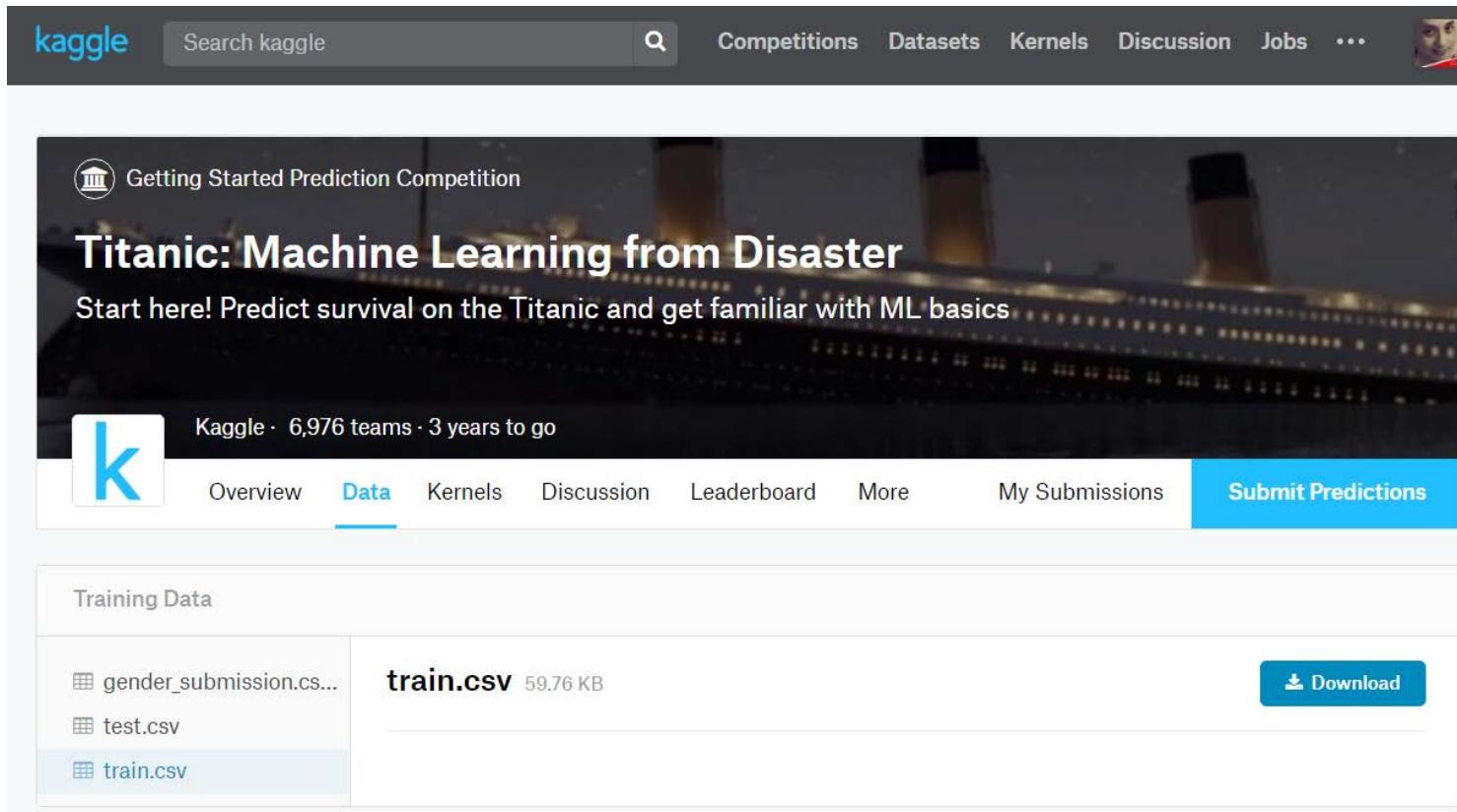


AML model development step

- Create: data preparation
- Train: teach the algorithm with data
- Score: see the performance
- Evaluate: compare performance of each algorithm
- Publish Web Service: production and re-train

Download Data set

kaggle / Titanic: Machine Learning from Disaster
<https://www.kaggle.com/c/titanic/data>



The screenshot shows the Kaggle website interface for the "Titanic: Machine Learning from Disaster" competition. At the top, there's a navigation bar with links for Competitions, Datasets, Kernels, Discussion, Jobs, and a user profile icon. Below the header, the competition title "Titanic: Machine Learning from Disaster" is displayed with a subtitle "Start here! Predict survival on the Titanic and get familiar with ML basics". A large image of the Titanic ship is in the background. The main content area shows the "Data" tab selected, with a "Training Data" section. It lists three files: "gender_submission.csv", "test.csv", and "train.csv". The "train.csv" file is highlighted with a blue background. To its right is a "Download" button with a downward arrow icon.

Data Dictionary

Data Dictionary

Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	
Age	Age in years	
sibsp	# of siblings / spouses aboard the Titanic	
parch	# of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

Variable Notes

Variable Notes

pclass: A proxy for socio-economic status (SES)

- 1st = Upper
- 2nd = Middle
- 3rd = Lower

age: Age is fractional if less than 1. If the age is estimated, is it in the form of xx.5

sibsp: The dataset defines family relations in this way...

- Sibling = brother, sister, stepbrother, stepsister
- Spouse = husband, wife (mistresses and fiancés were ignored)

parch: The dataset defines family relations in this way...

- Parent = mother, father
- Child = daughter, son, stepdaughter, stepson
- Some children travelled only with a nanny, therefore parch=0 for them.

View data set in Microsoft Excel

	A	B	C	D	E	F	G	H	I	J	K	L
1	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
2	1	0	3	Braund, Mr. Owen Hart	male	22	1	0	A/5 21171	7.25		S
3	2	1	1	Cumings, Mrs. John Brae	female	38	1	0	PC 17599	71.2833	C85	C
4	3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2.	7.925		S
5	4	1	1	Futrelle, Mrs. Jacques He	female	35	1	0	113803	53.1	C123	S
6	5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.05		S
7	6	0	3	Moran, Mr. James	male		0	0	330877	8.4583		Q
8	7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625	E46	S
9	8	0	3	Palsson, Master. Gosta	male	2	3	1	349909	21.075		S
10	9	1	3	Johnson, Mrs. Oscar Wenzel	female	27	0	2	347742	11.1333		S
11	10	1	2	Nasser, Mrs. Nicholas C	female	14	1	0	237736	30.0708		C
12	11	1	3	Sandstrom, Miss. Margaret	female	4	1	1	PP 9549	16.7	G6	S
13	12	1	1	Bonnell, Miss. Elizabeth	female	58	0	0	113783	26.55	C103	S
14	13	0	3	Saundercock, Mr. William	male	20	0	0	A/5. 2151	8.05		S

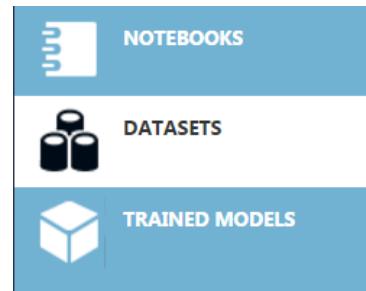
Machine Learning experiment creation working steps

Working steps;

- Import Data set
- Create New Experiment
- Prepare Data
 - Drop the columns PassengerID, Name, Ticket, Cabin
 - Make categorical values: Survived, Pclass, Sex, Embarked
 - Replace missing value with median
 - Drop rows with missing data
 - Create Label
 - Split data 70% training and 30% scoring
- Select Algorithm : Two-Class Boosted Decision
- Train
- Score

Import Data set

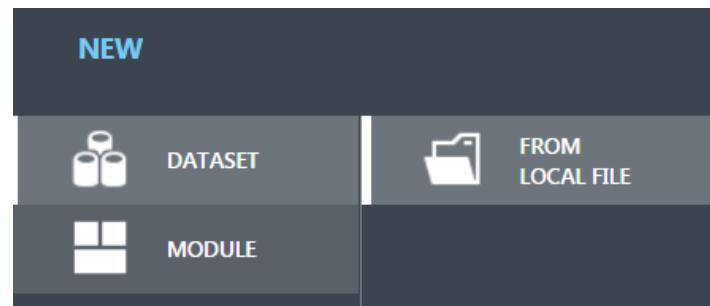
1. Click DATASETS



2. Click NEW



3. Click FROM LOCAL FILE



Upload a new dataset

4. Click Choose File
5. Brows and select train.csv
6. ENTER A NAME FOR THE NEW DATASET
TitanicTrain1
7. SELECT A TYPE FOR THE NEW DATASET
Generic CSV File with a header (.csv)
8. PROVIDE AN OPTIONAL DESCRIPTION
kaggle Titanic: Machine Leering from disaster
9. Click



Upload a new dataset

SELECT THE DATA TO UPLOAD:

train.csv

This is the new version of an existing dataset

ENTER A NAME FOR THE NEW DATASET:

TitanicTrain1

SELECT A TYPE FOR THE NEW DATASET:

Generic CSV File with a header (.csv)

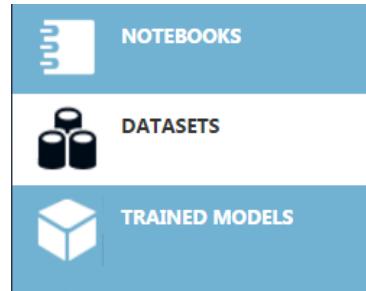
PROVIDE AN OPTIONAL DESCRIPTION:

kaggle Titanic: Machine Learning from Disaster



Verify dataset uploaded

1. Click DATASETS



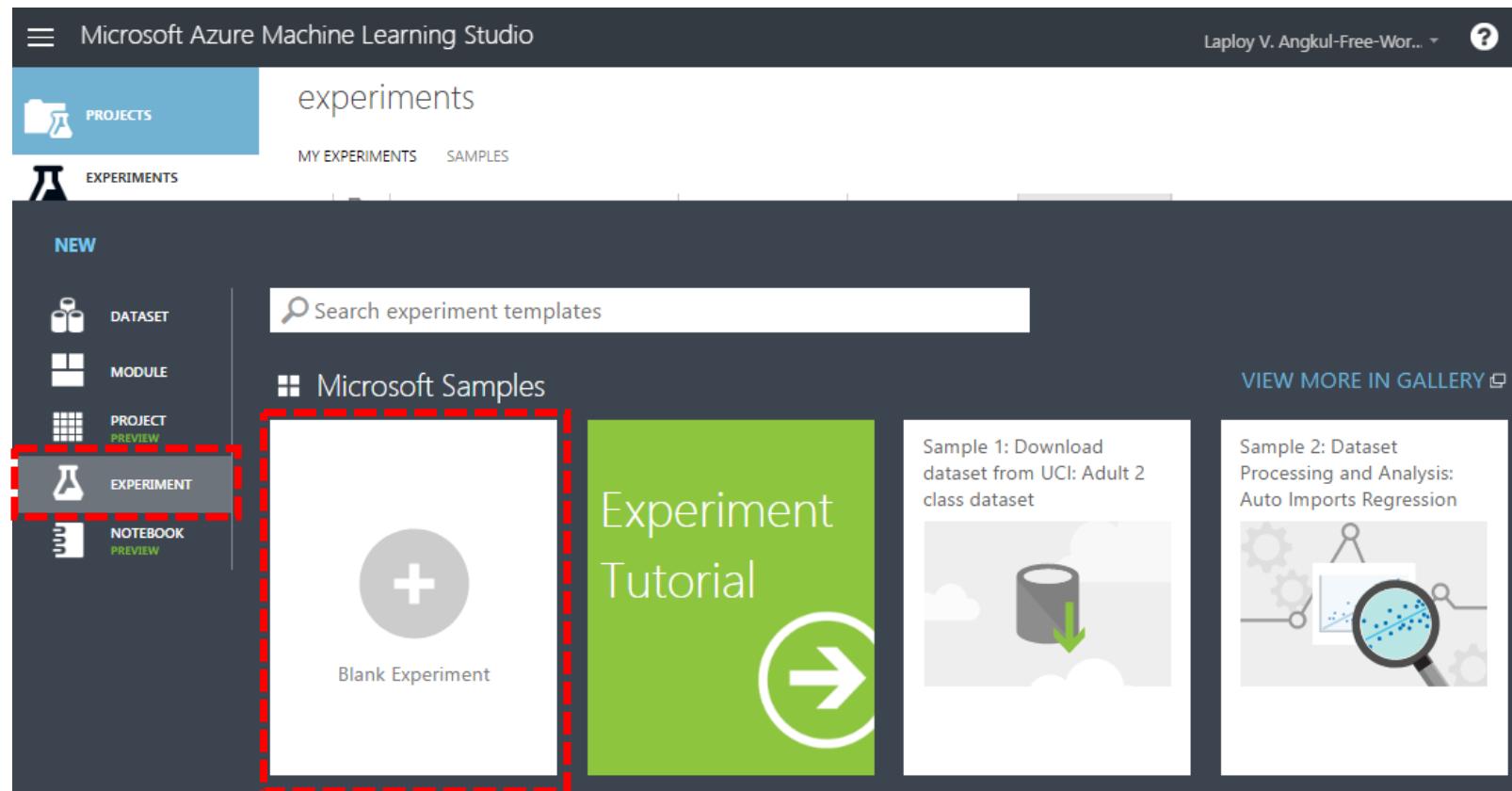
2. Make sure TitanicTrain1 is in MY DATASETS list

A screenshot of the 'MY DATASETS' list in the Microsoft Azure Machine Learning studio. The page title is 'datasets'. There are two tabs at the top: 'MY DATASETS' (which is selected and highlighted in blue) and 'SAMPLES'. Below the tabs is a table with four columns: 'NAME', 'SUBMITTED BY', 'DESCRIPTION', and 'DATA TYPE'. A single dataset row is listed: 'TitanicTrain1' submitted by 'laploy' with a description 'kaggle Titanic: Ma...' and data type 'GenericCSV'.

NAME	SUBMITTED BY	DESCRIPTION	DATA TYPE
TitanicTrain1	laploy	kaggle Titanic: Ma...	GenericCSV

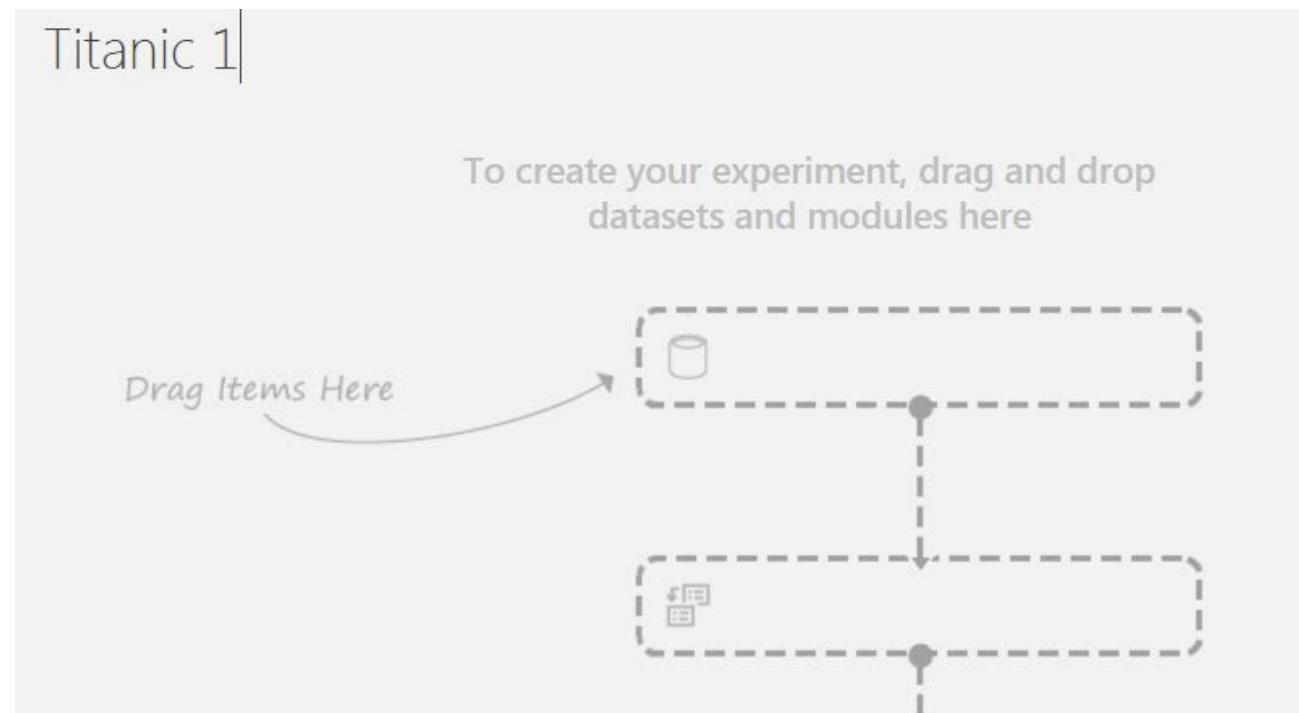
Create New Experiment

Create Blank Experiment



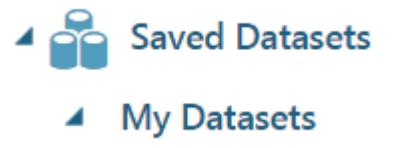
Set experiment name

Type in name = Titanic 1

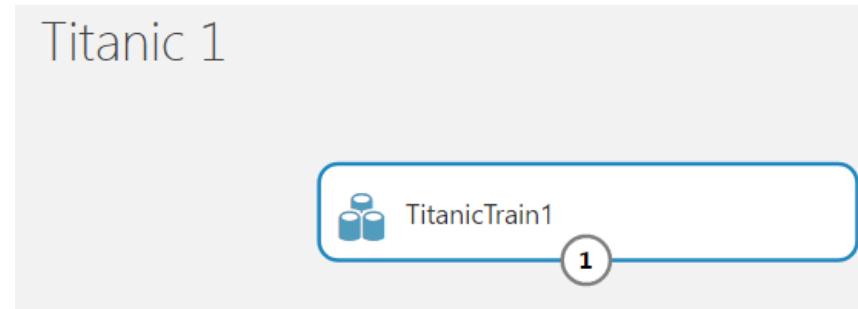


Drag & drop dataset to canvas

1. Click Saved Datasets / My Datasets



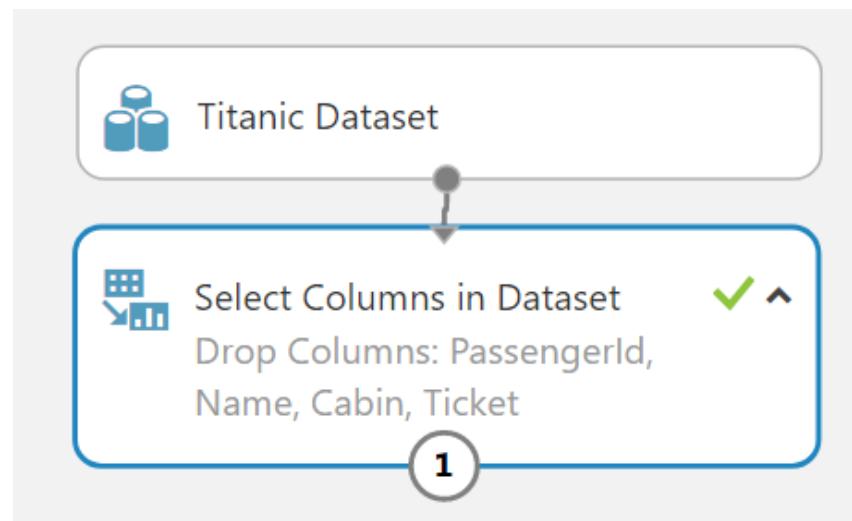
2. Drag & drop TitanicTrain1 to canvas



3. Visualize output

Drop the columns PassengerID, Name, Ticket, Cabin

1. Drag & drop module Select Columns in Dataset
2. Selected column = Drop Columns: PassengerId, Name, Cabin, Ticket
3. Click Launch column selector
4. Visualize



Properties Project

▲ Select Columns in Dataset

Select columns

Selected columns:

All columns

Exclude column names:

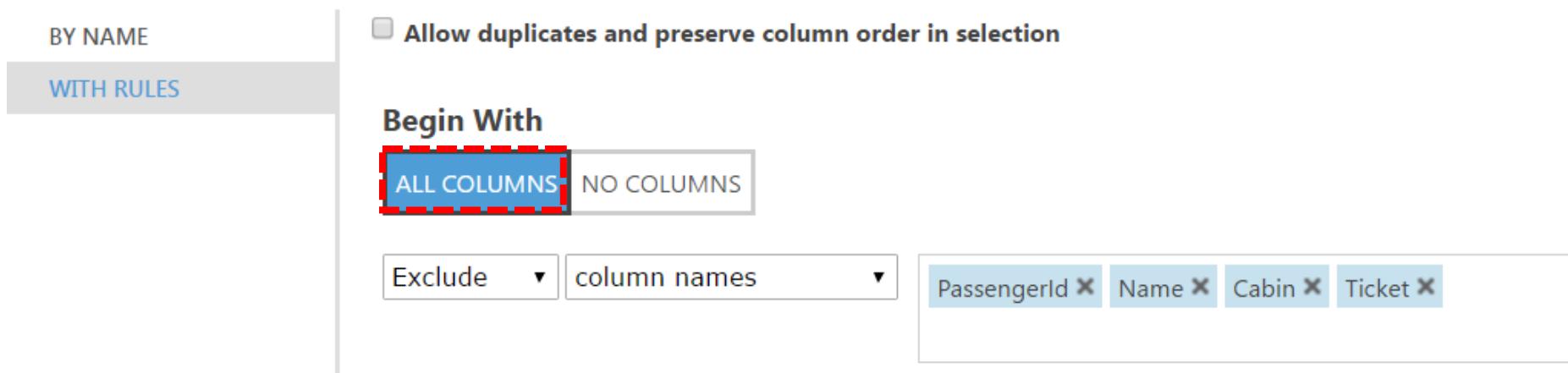
PassengerId,Name,Cabin,Tic

Launch column selector

Drop the columns PassengerID, Name, Ticket, Cabin

5. Begin With = ALL COLUMNS / Exclude / column name
6. Selected column PassengerID, Name, Ticket, Cabin
7. Click 
8. Visualize

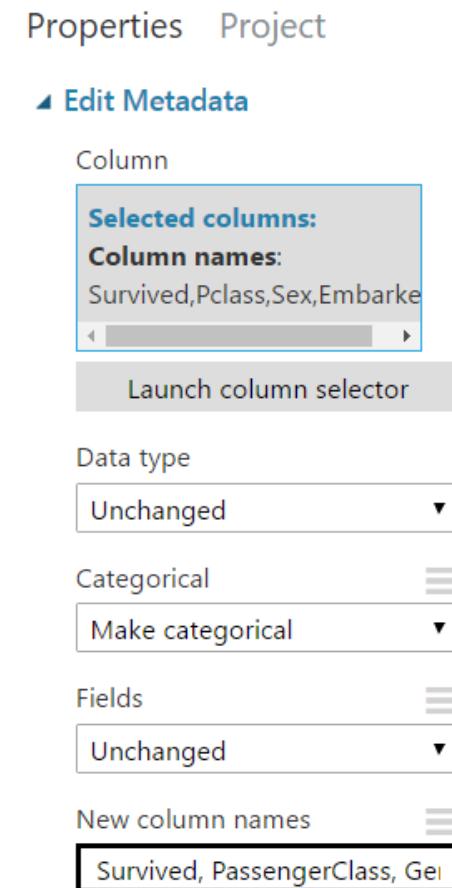
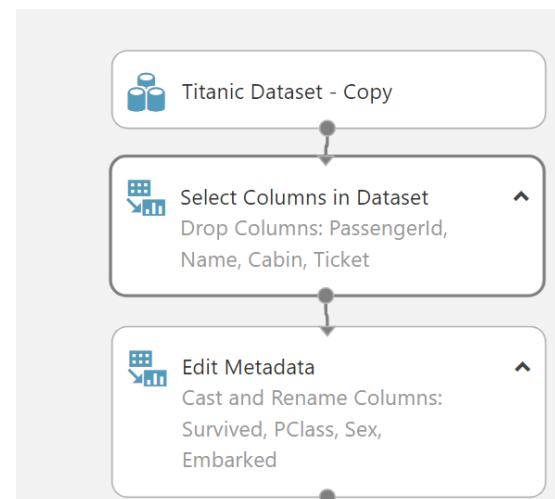
Select columns



The screenshot shows the 'Select columns' dialog box. On the left, there are two tabs: 'BY NAME' (disabled) and 'WITH RULES' (selected). At the top right is a checkbox labeled 'Allow duplicates and preserve column order in selection'. Below this, under 'Begin With', there are two options: 'ALL COLUMNS' (selected, highlighted with a red dashed border) and 'NO COLUMNS'. Further down, there are two dropdown menus: 'Exclude' (set to 'column names') and 'column names' (containing 'PassengerId', 'Name', 'Cabin', and 'Ticket', each with a delete 'X' icon). A large empty rectangular area is at the bottom right.

Make categorical values: Survived, Pclass, Sex, Embarked

1. Drag & drop Edit Metadata
2. Comment = Cast and Rename Columns: Survived, PClass, Sex, Embarked
3. Selected column Survived, Pclass, Sex, Embarked
4. Data type = Unchanged
5. Categorical = Make categorical
6. Fields = Unchanged
7. New column name = Survived, PassengerClass, Gender, PortEmbarkation
8. Visualize



Rename columns

1. Drag & drop Edit Metadata
2. Comment = Rename Columns: SiblingSpouse, ParentChild, FarePrice
3. Selected column SibSp, Parch, Fare
4. Data type = Unchanged
5. Categorical = Unchanged
6. Fields = Unchanged
7. New column name = SiblingSpouse, ParentChild, FarePrice
8. Visualize

The screenshot shows the configuration interface for the 'Edit Metadata' component. A blue box highlights the 'Rename Columns' section, which contains the value 'SiblingSpouse, ParentChild, FarePrice'. A circled '1' is positioned at the bottom right corner of this highlighted area. To the right of the component are several configuration settings:

- Column**:
Selected columns:
Column names:
SibSp,Parch,Fare
- Launch column selector**
- Data type**: Unchanged
- Categorical**: Unchanged
- Fields**: Unchanged
- New column names**: SiblingSpouse, ParentChild, F

Replace missing value with median

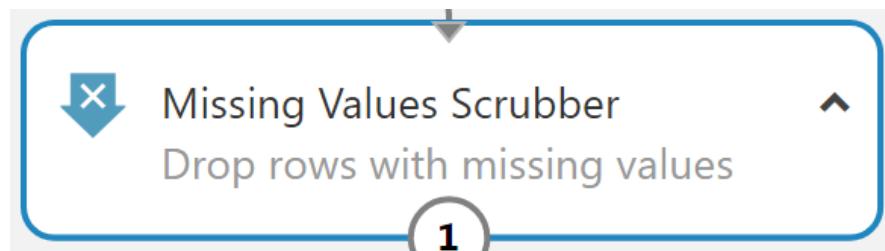
1. Drag & drop Missing Values Scrubber
2. Comment = Replace missing value with median
3. Set properties
4. Visualize

The screenshot shows the Azure Machine Learning studio interface. On the left, a component titled "Missing Values Scrubber" is highlighted with a blue border and a circled number "1" at the bottom. The component has a comment below it: "Replace missing values with median". To the right, the "Properties" pane is open, showing the following settings for the "Missing Values Scrubber" component:

- For missing values:** Replace with median
- Cols with all MV:** KeepColumns
- MV indicator column:** DoNotGenerate

Drop rows with missing data

1. Drag & drop Missing Values Scrubber
2. Comment = Drop rows with missing values
3. Set properties
4. Visualize



Properties Project

Missing Values Scrubber

For missing values

Remove entire row ▾

Cols with all MV

KeepColumns ▾

MV indicator column

DoNotGenerate ▾

Create Label

1. Drag & drop Edit Metadata
2. Comment = Assigning target variable
3. Selected column = Survived
4. Data type = Unchanged
5. Categorical = Unchanged
6. Fields = Label
7. New column name = -
8. Visualize



Properties Project

▲ Edit Metadata

Column

Selected columns:
Column names: Survived

Launch column selector

Data type

Unchanged ▾

Categorical ▾

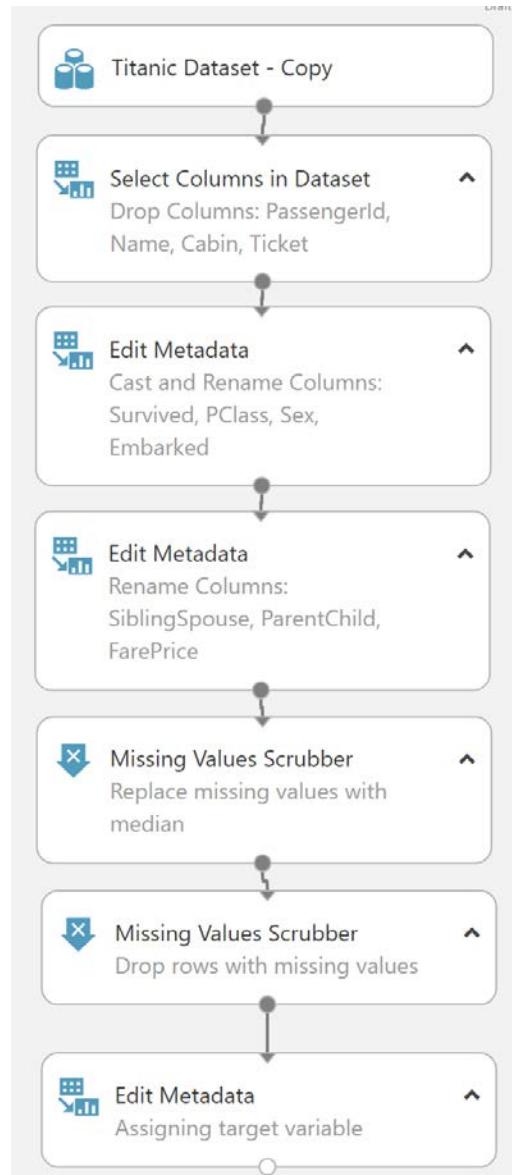
Unchanged ▾

Fields ▾

Label ▾

New column names ▾

Import and Dataset preparation



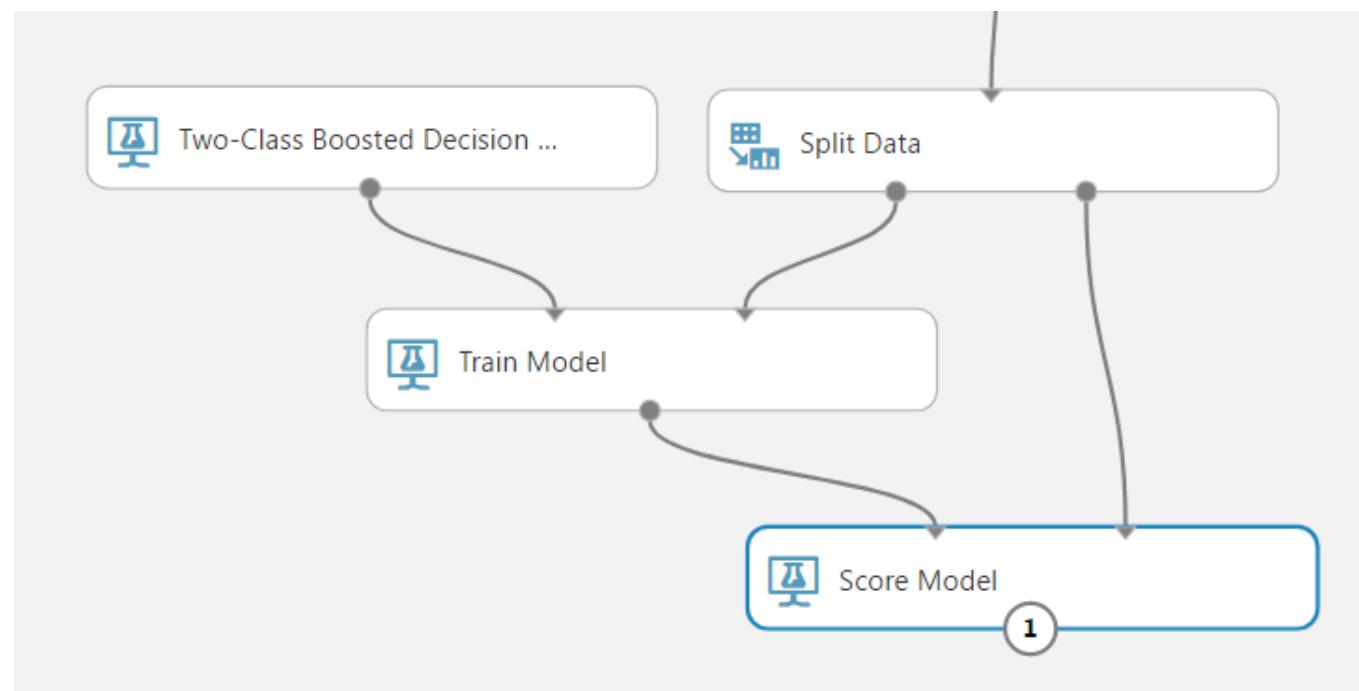
Split data 70% training and 30% scoring

1. Drag & drop Split data
2. Set properties

The screenshot shows the Microsoft Azure Machine Learning studio interface. On the left, there is a workspace titled "titanic1" containing a workflow. The workflow consists of three steps connected by arrows: "Missing Values Scrubber" (Drop rows with missing values), "Edit Metadata" (Assigning target variable), and "Split Data". To the right of the workspace, there are two tabs: "Properties" and "Project". The "Properties" tab is selected, showing the "Split Data" configuration. The "Splitting mode" is set to "Split Rows", with a fraction of "0.7" assigned to the first part. The "Randomized split" option is checked, and the "Random seed" is set to "0". There is also a section for "Stratified split" which is set to "False".

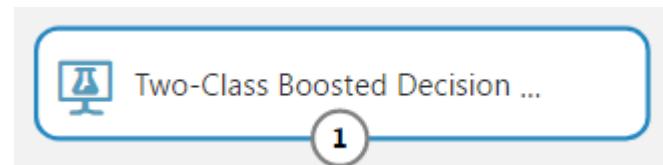
Add Algorithm, Train and Score

- Add Two-Class Boosted Decision tree
 - Add Train Model
 - Add Score Model



Add Two-Class Boosted Decision tree

1. Drag & drop Two-Class Boosted Decision tree
2. Set properties



Properties Project

▲ Two-Class Boosted Decision Tree

Create trainer mode

Single Parameter ▾

Maximum number of leave...

20

Minimum number of samp...

10

Learning rate

0.2

Number of trees construct...

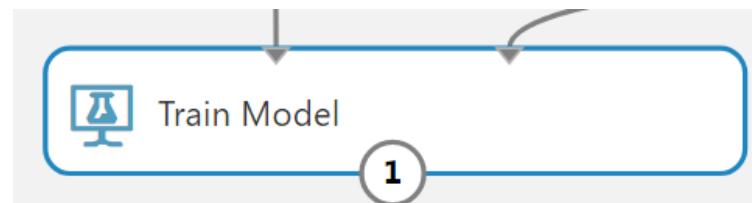
100

Random number seed

Allow unknown catego.. ▾

Add Train Model

1. Drag & drop train model
2. Set column to Survived



Properties Project

▲ Train Model

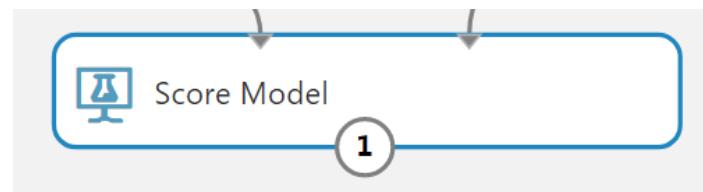
Label column

Selected columns:
Column names: Survived

Launch column selector

Add Score Mode

1. Drag & drop Score Model
2. Set property = Append score column
3. Save
4. Run experiment
5. Visualize



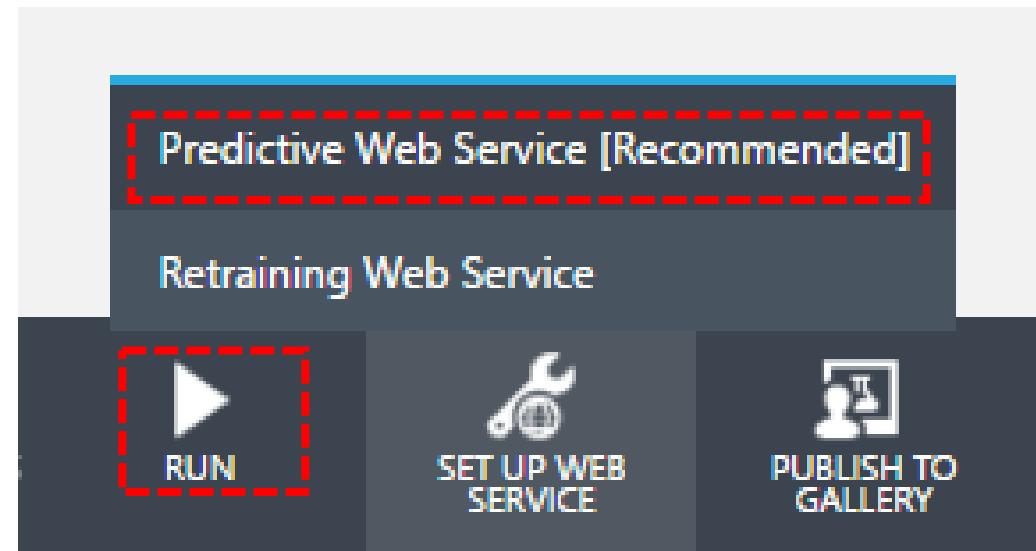
Properties Project

Score Model

Append score column...

Create web service

1. Save as Titanic 2
2. Run Titanic 2
3. Click SET UP WEB SERVICE
4. Click Predictive Web Service
5. Click RUN
6. Click SET UP WEB SERVICE



Create web service Titanic 2 [predictive exp.] page

Microsoft Azure Machine Learning Studio

titanic 2 [predictive exp.]

DASHBOARD CONFIGURATION

General New Web Services Experience [preview](#)

Published experiment

[View snapshot](#) [View latest](#)

Description

No description provided for this web service.

API key

jzA5N8IkLqQSt3ZPLo9n4nxIQHSfOK74ewHlsjJcgZ4O6tdnZi8JmzGIG39ZgIPF4etmj4c

Default Endpoint

API HELP PAGE	TEST	APPS	LAST UPDATED	P
REQUEST/RESPONSE	Test preview	Excel 2013 or later Excel 2010	6/4/2017 1:20:12 PM	
BATCH EXECUTION	Test preview	Excel 2013 or later workbook	6/4/2017 1:20:12 PM	

Creating predictive experiment

DETAILS [i](#) CLOSE [x](#)

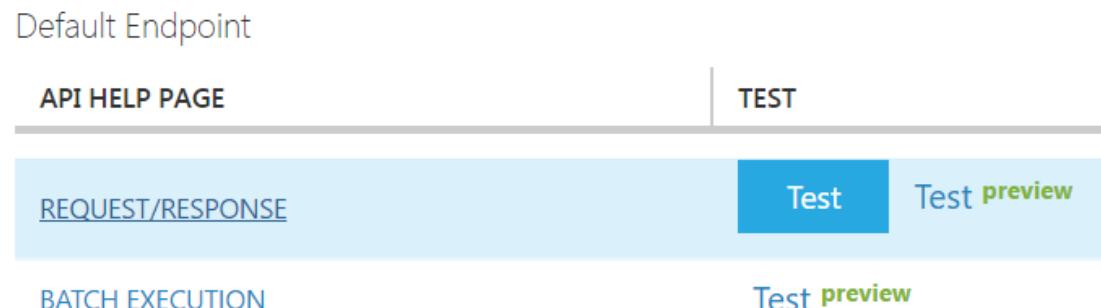
[NEW](#) [DELETE](#) 1 [E](#)

The screenshot shows the Microsoft Azure Machine Learning Studio interface. On the left is a vertical toolbar with icons for dashboard, configuration, general, published experiment, snapshot, latest, description, API key, default endpoint, API help page, test, apps, last updated, and a refresh button. The main area displays a project named "titanic 2 [predictive exp.]". It includes sections for General (with a link to "New Web Services Experience preview"), Published experiment (with links to View snapshot and View latest), Description (empty), and API key (containing a long string of characters). Below this is a table for Default Endpoint, showing REQUEST/RESPONSE and BATCH EXECUTION sections with their respective test and preview links, and download links for Excel 2013 or later and Excel 2010. At the bottom, a progress bar indicates "Creating predictive experiment" and provides options for DETAILS, CLOSE, and X. A navigation bar at the very bottom includes NEW, DELETE, and a count of 1 followed by an edit icon.

Test web service

Web service testing

- REQUEST/RESPONSE Test
- REQUEST/RESPONSE Test preview
- REQUEST/RESPONSE Excel workbook test
- BATCH EXECUTION Test preview
- BATCH EXECUTION Excel workbook test



REQUEST/RESPONSE Test

1. Test with known result
2. Open file kaggle test.csv
3. Take one passenger
4. Click REQUEST/RESPONSE Test
5. Fill in the form

Test Titanic 2 [Predictive Exp.] Service

Enter data to predict

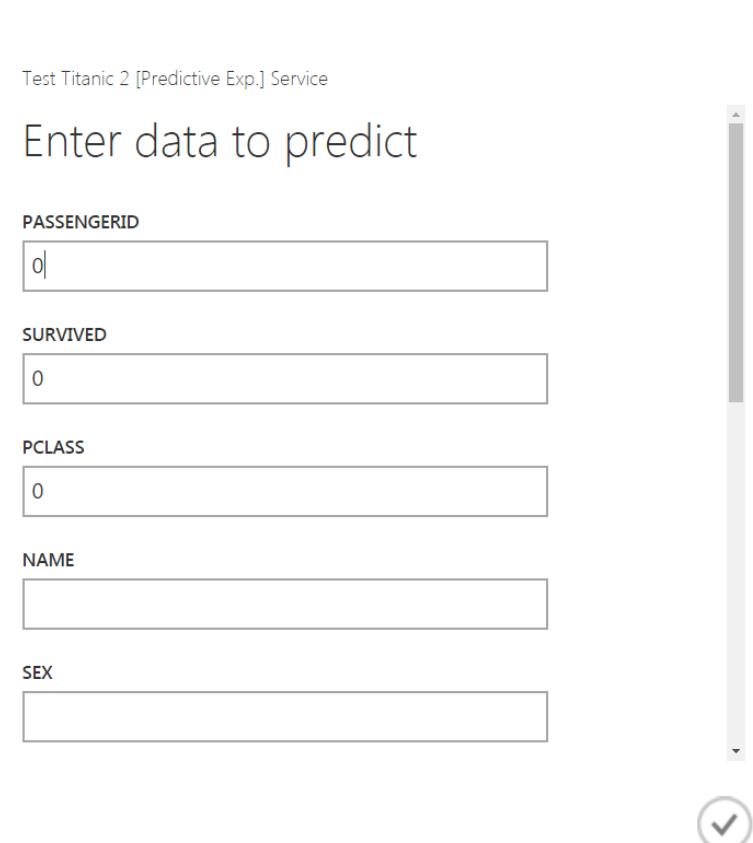
PASSENGERID

SURVIVED

PCLASS

NAME

SEX



REQUEST/RESPONSE Test preview

1. Test with known result
2. Open file kaggle test.csv
3. Take one passenger
4. Click REQUEST/RESPONSE Test preview
5. Click Enable (Sample Data)
6. Fill in the form
7. Click Test Request-Response

Titanic 2 [Predictive Exp.]

default

View in Studio

Request-Response Batch

Sample Data

Enable

Your prediction results will display here.

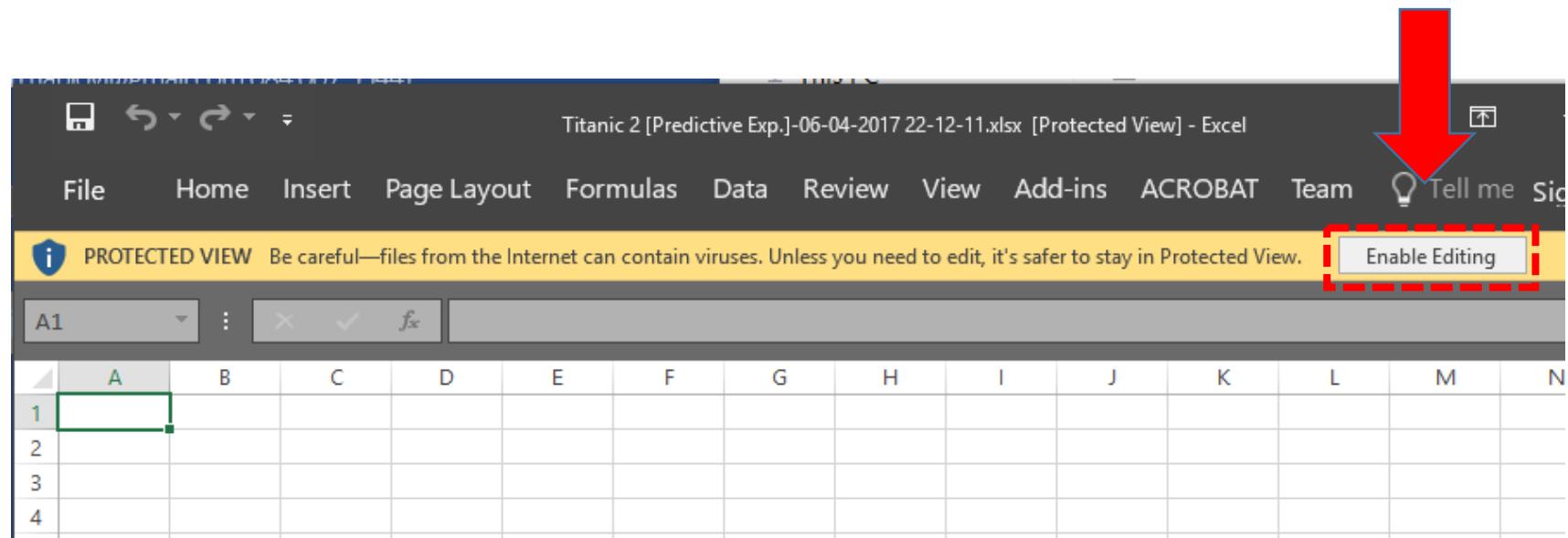
PassengerId: 1

Survived: 1

Pclass: 1

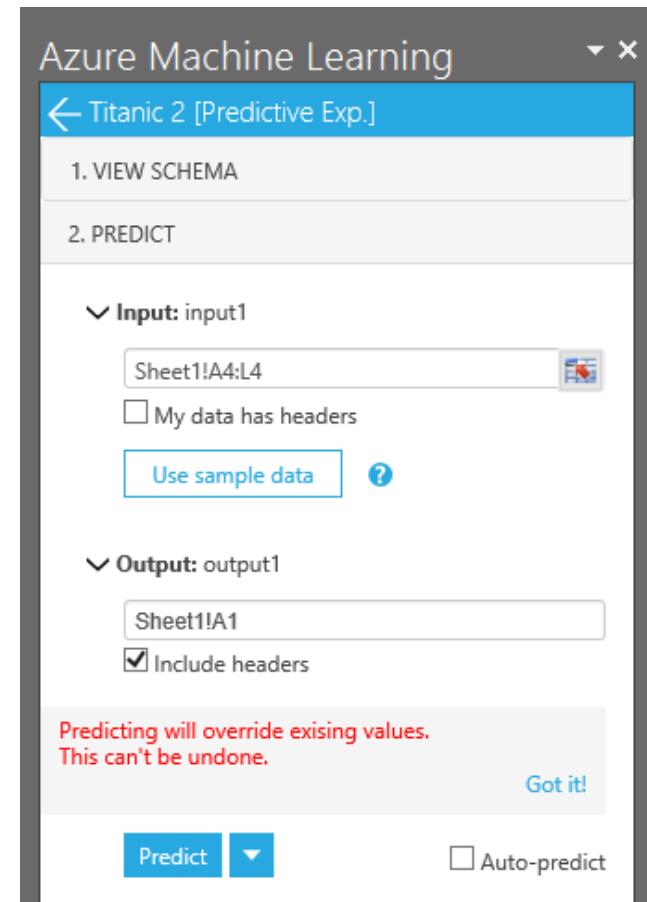
REQUEST/RESPONSE Excel workbook test

1. Test with known result
2. Open file kaggle test.csv
3. Take one passenger
4. Click REQUEST/RESPONSE Excel 2013 or later
5. Open file Titanic 2 [Predictive Exp.] on Desktop
6. Click Enable Editing



REQUEST/RESPONSE Excel workbook test

1. Input = Sheet1!A4:L4
2. My data has headers = uncheck
3. Output = Sheet1!A1
4. Include headers = check
5. Copy a line from file kaggle test.csv to A4
6. Click Predict



REQUEST/RESPONSE Excel workbook test

Test result

The screenshot shows a Microsoft Excel spreadsheet titled "Titanic 2 [Predictive Exp.] 06-04-2017 22-12-11.xlsx - Excel". The ribbon menu includes File, Home, Insert, Page Layout, Formulas, Data, Review, View, Add-ins, ACROBAT, Team, and Tell me what you want. The active cell is M11. The table has 12 columns labeled A through L. Column A is "Survived", B is "PassengerClass", C is "Gender", D is "Age", E is "SiblingSpo", F is "ParentChil", G is "FarePrice", H is "PortEmbar", I is "ScoredLat", J is "ScoredLong", K is "ScoredProbabilities", and L is an empty column. Row 1 contains the column headers. Row 2 contains values: 0, 2, male, 52, 0, 0, 13, S, 0, 0.000509969. Row 3 is blank. Row 4 contains values: 715, 0, 2, Greenbe, male, 52, 0, 0, 250647, 13, S.

	Survived	PassengerClass	Gender	Age	SiblingSpo	ParentChil	FarePrice	PortEmbar	ScoredLat	ScoredLong	Scored Probabilities
1	0	2	male	52	0	0	13	S	0	0	0.000509969
2											
3											
4	715	0	2	Greenbe	male	52	0	0	250647	13	S
5											
6											

More information

More information on Classification Model

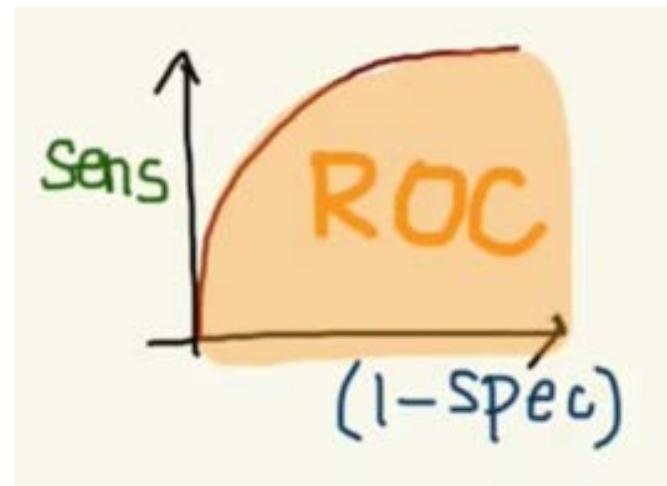
Two-Class Boosted Decision Tree

<https://msdn.microsoft.com/en-us/library/azure/dn906025.aspx>

Machine learning algorithm cheat sheet for Microsoft Azure Machine Learning Studio

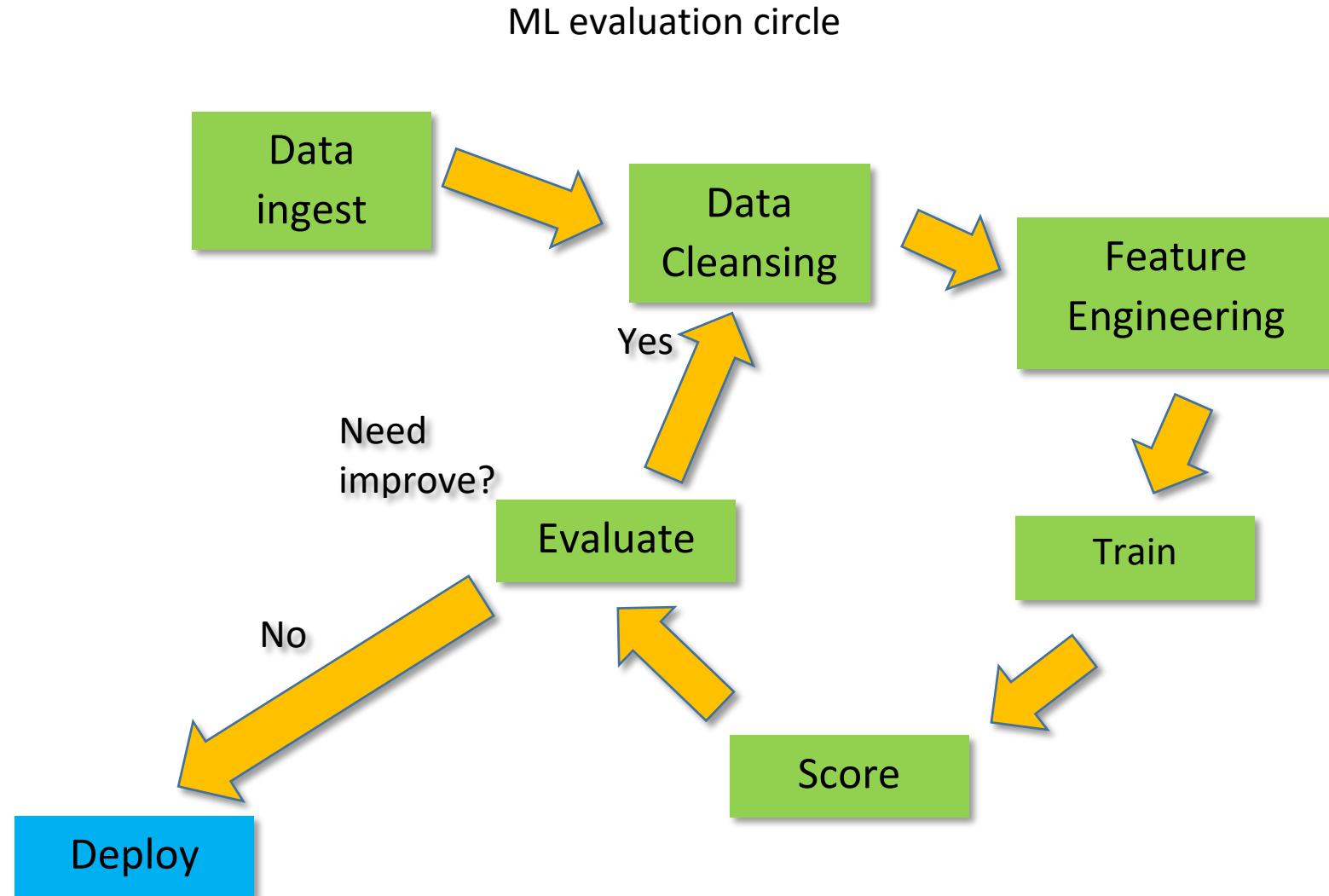
<https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-algorithm-cheat-sheet>

ML EVALUATION



In this session

- ML train and evaluation circle
- How to read Histogram
- How to read Box Plot
- Adding Evaluate Model
- How to read ROC curve
- Area Under the Curve (AUC)
- How to read Evaluation metrics



How to read Scoring results

Titanic Evaluate > Score Model > Scored dataset

rows columns
267 10

	Survived	PassengerClass	Gender	Age	SiblingSpouse	ParentChild	FarePrice	PortEmbarkation	Scored Labels	Scored Probabilities
view as										
1	3		male	20	1	1	15.7417	C	0	0.128143
1	2		female	25	1	1	30	S	1	0.999319
0	3		male	28	0	0	7.8958	C	0	0.40695
1	3		female	28	1	1	22.3583	C	1	0.993964
0	3		male	28	0	0	9.5	S	0	0.000195
0	1		male	29	0	0	30	S	1	0.97861
1	1		male	49	1	0	56.9292	C	1	0.932772

- This table = Scored dataset
- Row = 267 / Columns = 10
- Total column = 10 / Left 8 = features / Right 2 = prediction results
- Scored Label 0 = dead 1 = survived
- Scored Probabilities (SP) SP <=0.5 == dead / SP > 0.5 == survived

How to read Scoring Statistics

▲ Statistics

Mean	28.8265
Median	28
Min	0.42
Max	80
Standard Deviation	12.3791
Unique Values	61
Missing Values	0
Feature Type	Numeric Feature

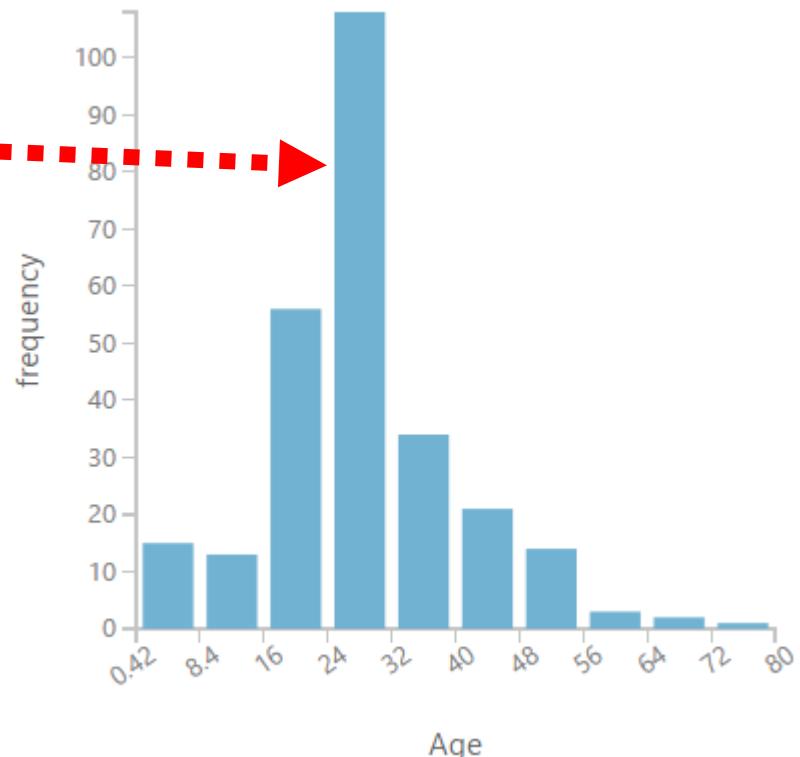
Show Statistics of the Scored dataset

- Mean = Sum of all the values divided by the number of values
- Median = The midpoint of the data after being ranked
- Standard Deviation = The square root of the variance
- Unique Values
- Missing Value

How to read Score Histogram

Histogram

- Representation: distribution of numerical data
- Bin: series of intervals (bin) 
- Count: values fall into each interval



How to read Box Plot

Box Plot

Box Plot (whisker) is a standardized way of displaying the distribution of data

- Median: marks the mid-point of the data
- Box: middle 50% of scores for the group.
- Upper quartile: 75% of the scores fall below the upper quartile.
- Lower quartile: 25% of scores fall below the lower quartile.
- Whiskers: scores outside the middle 50%

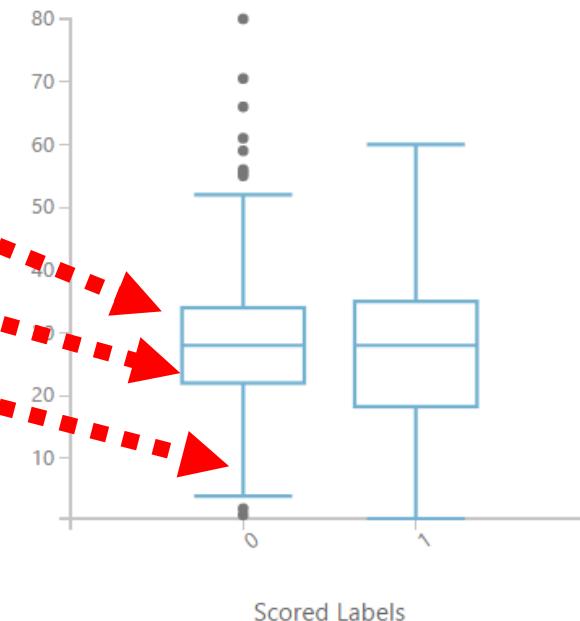
0 = dead

Visualizations

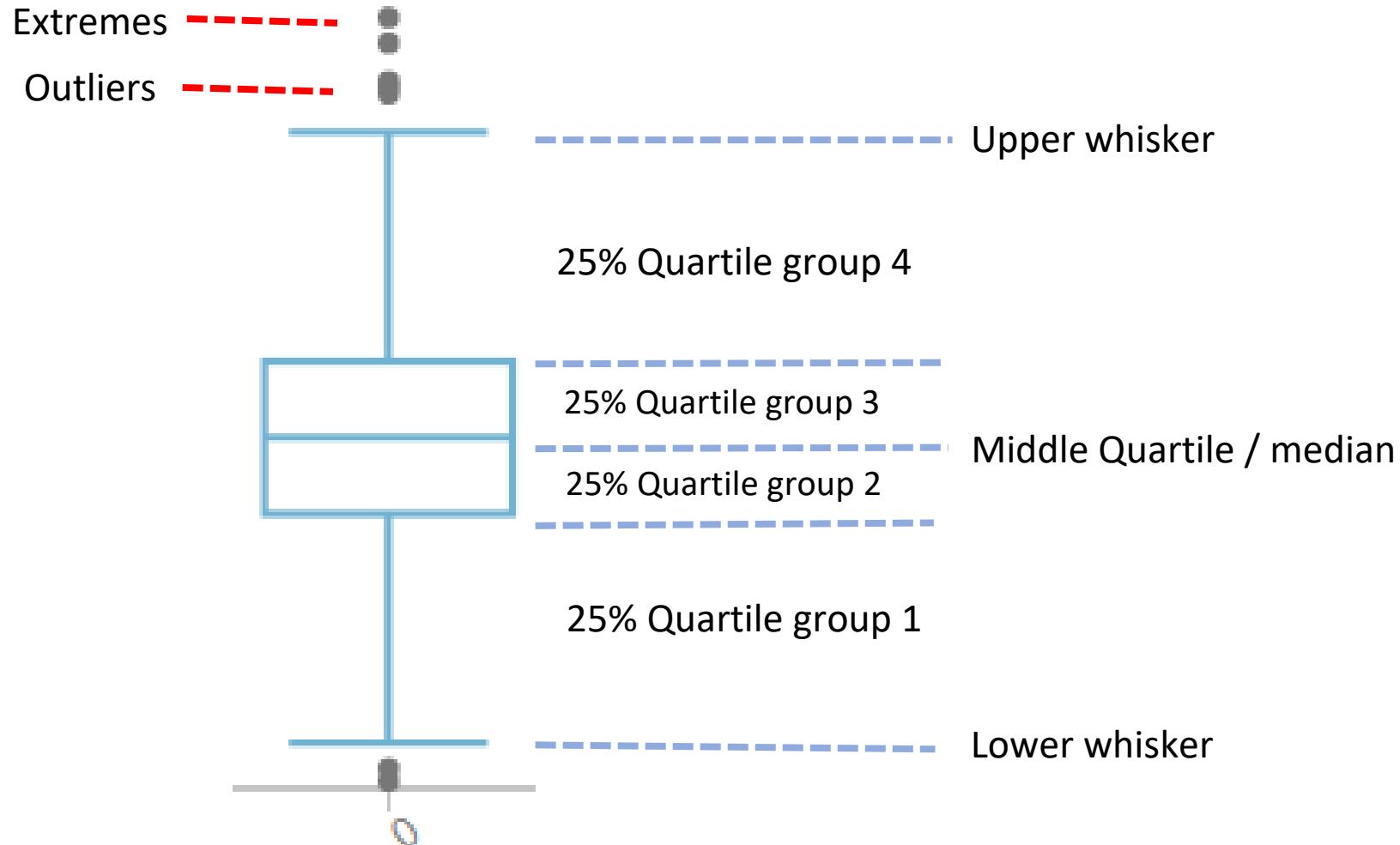
Age

MultiboxPlot

compare to



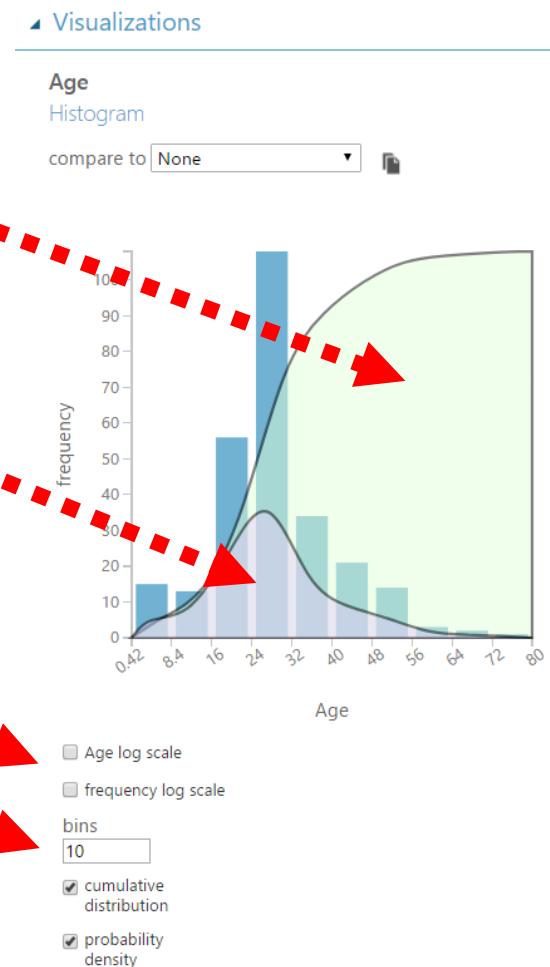
Box Plot Definitions



Histogram option

Histogram options

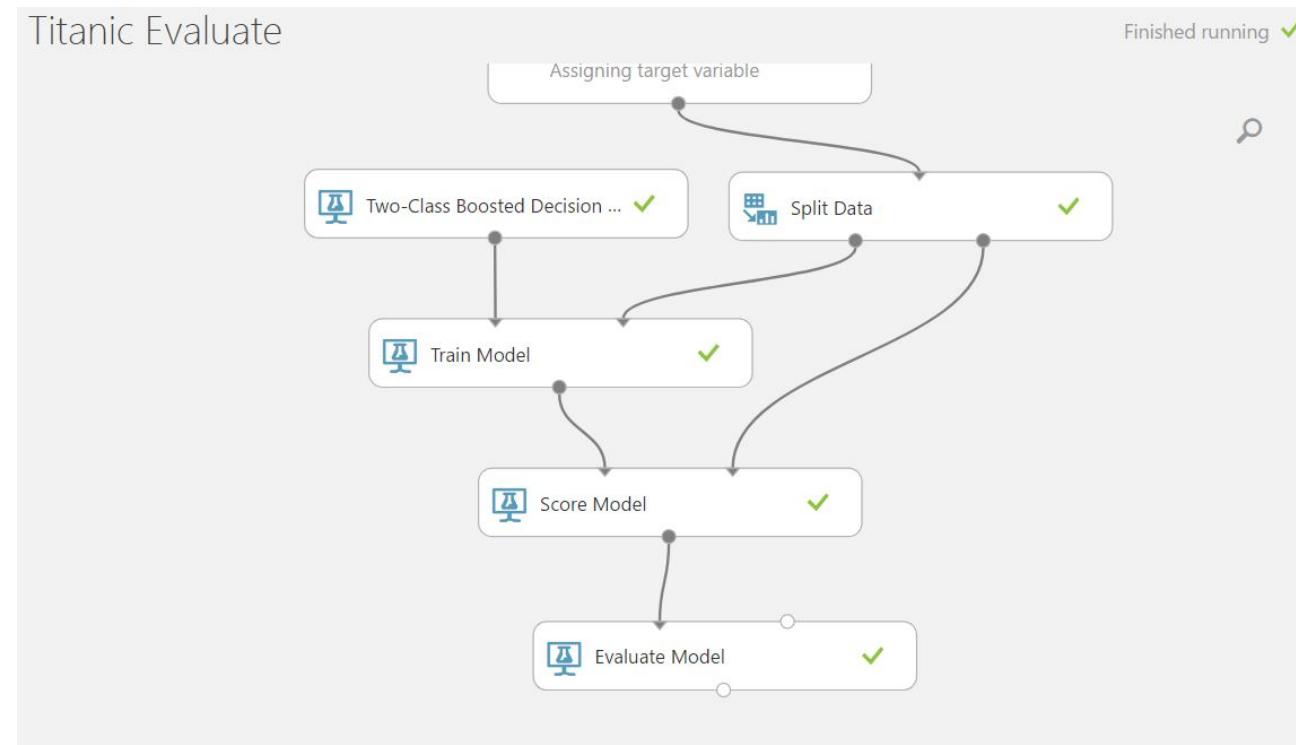
- Cumulative distribution function (cdf): shows "How common are samples that are less than or equal to this value?"
- Probability density function (pdf): shows "How common are samples at exactly this value?"
- Scale: scaling the distribution
- bins: number of bin



Adding Evaluate Model

Adding Evaluate Model

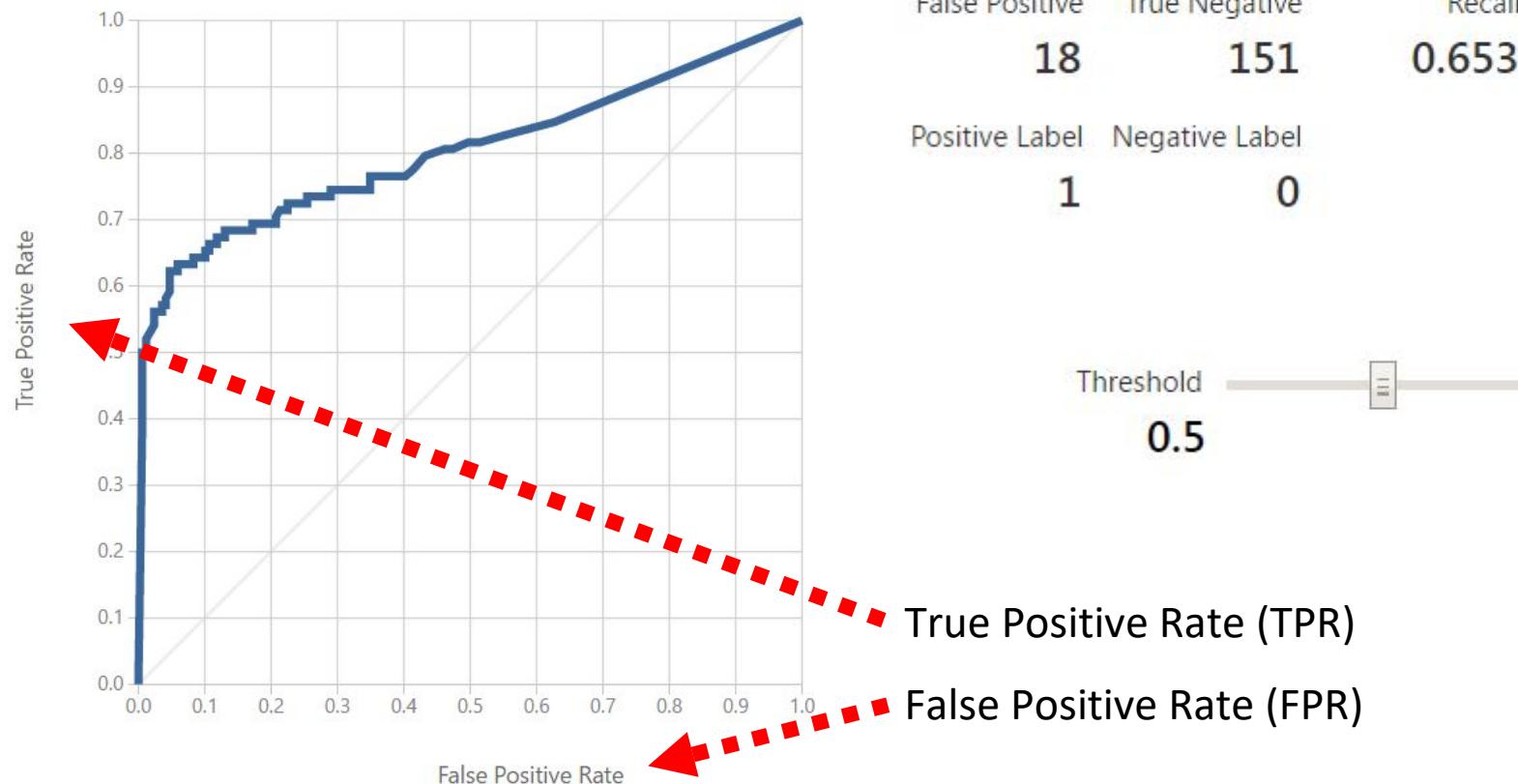
1. Open Titanic 1 Experiment
2. Save as Titanic Evaluate
3. Add Evaluate Model
4. Run the Experiment



Receiver Operating Characteristic (ROC) Curve

Titanic Evaluate > Evaluate Model > Evaluation results

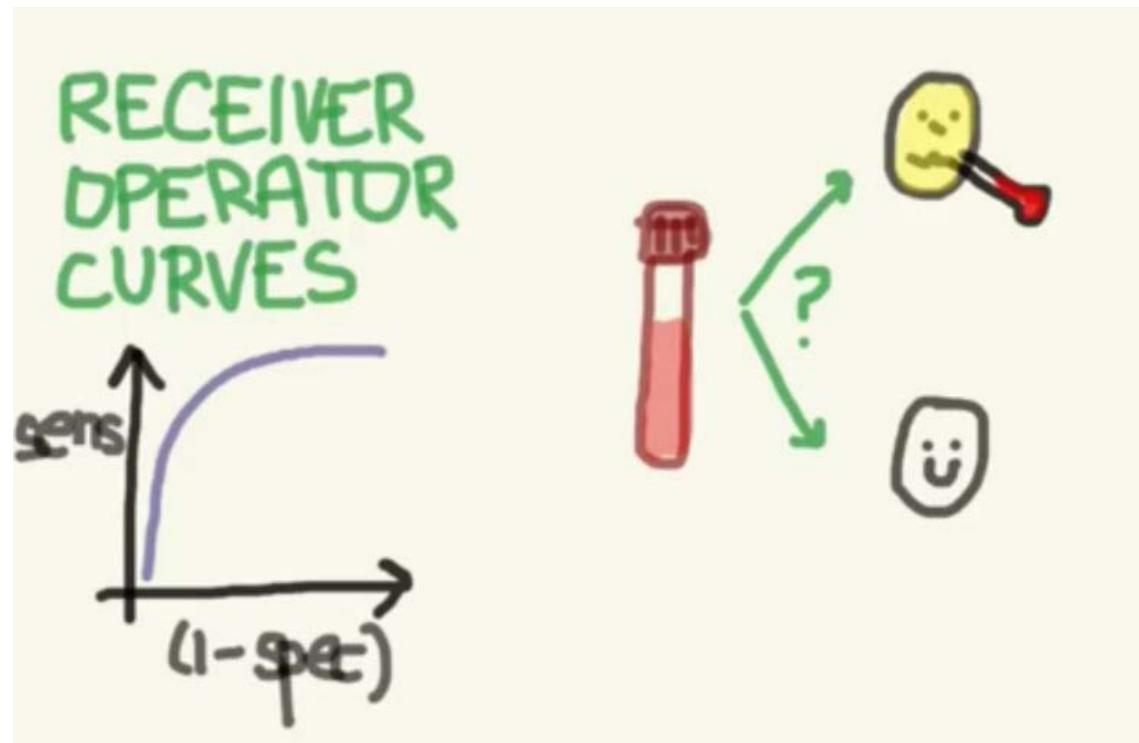
ROC PRECISION/RECALL LIFT



True Positive	False Negative	Accuracy	Precision
64	34	0.805	0.780
False Positive	True Negative	Recall	F1 Score
18	151	0.653	0.711
Positive Label	Negative Label		
1	0		
Threshold	0.5	AUC	0.817

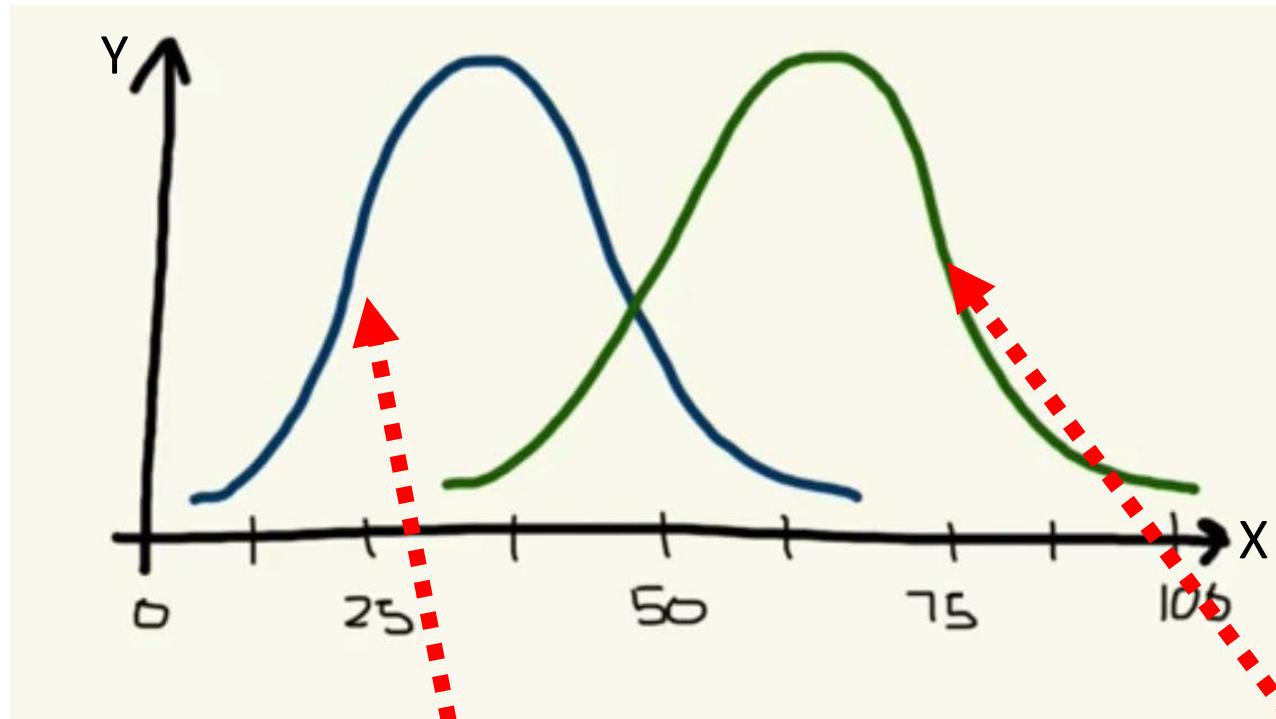
How to read ROC curve

ROC curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied.



ROC curve prediction result who have disease who don't

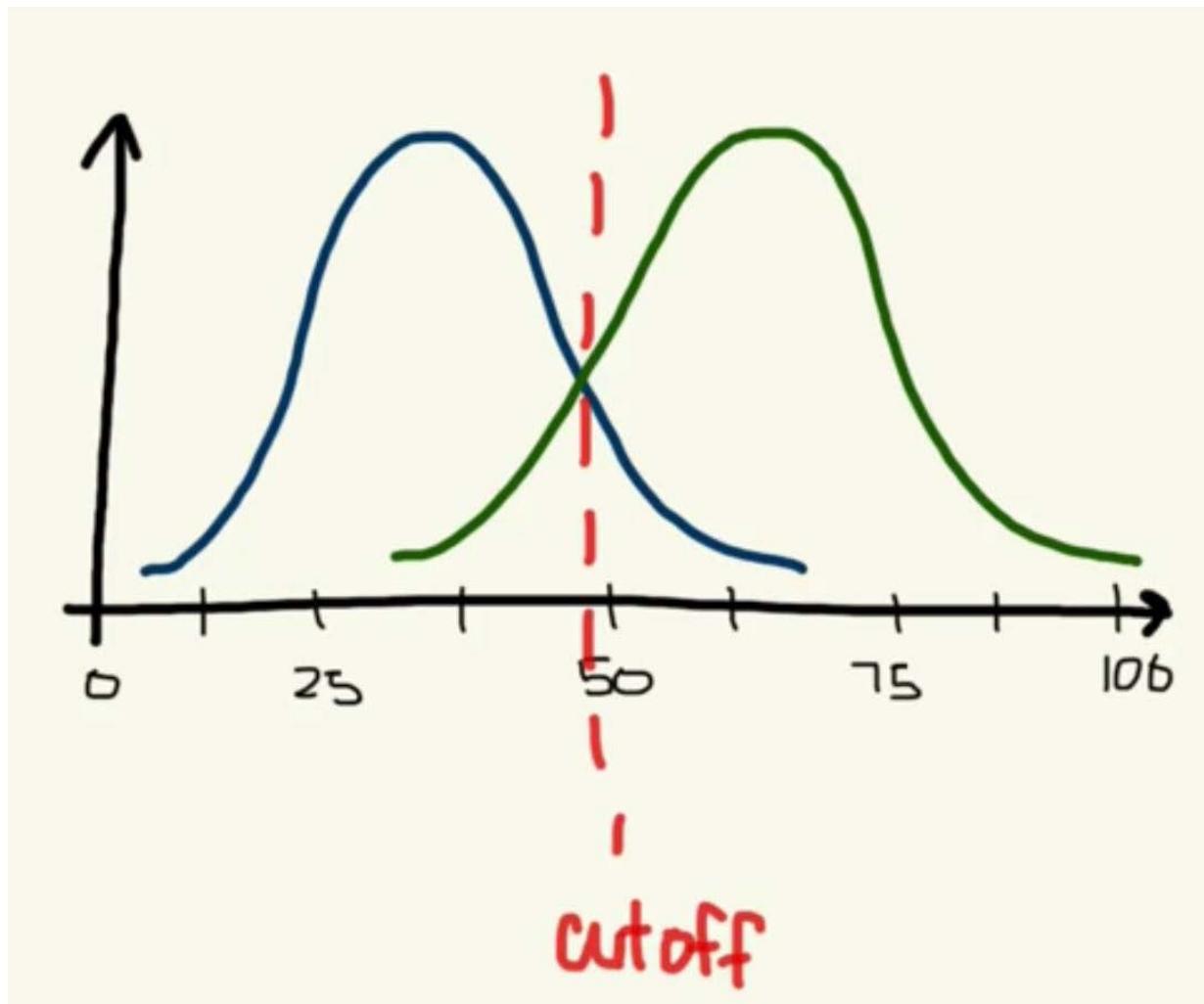
Distribution score



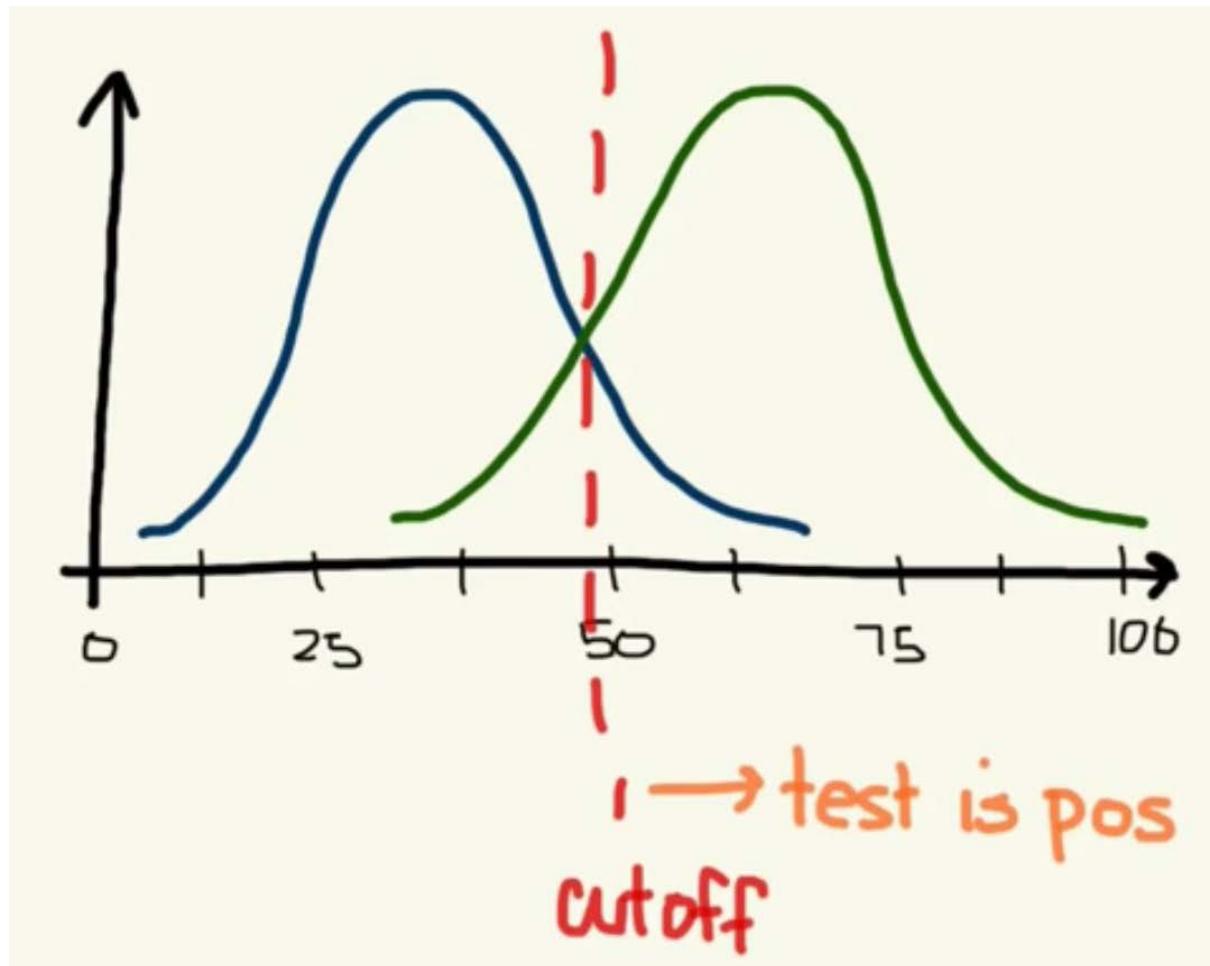
Left distribution = patient who do NOT have disease (survived) / Right = have disease (dead)

x axis = score / y axis = number of patient

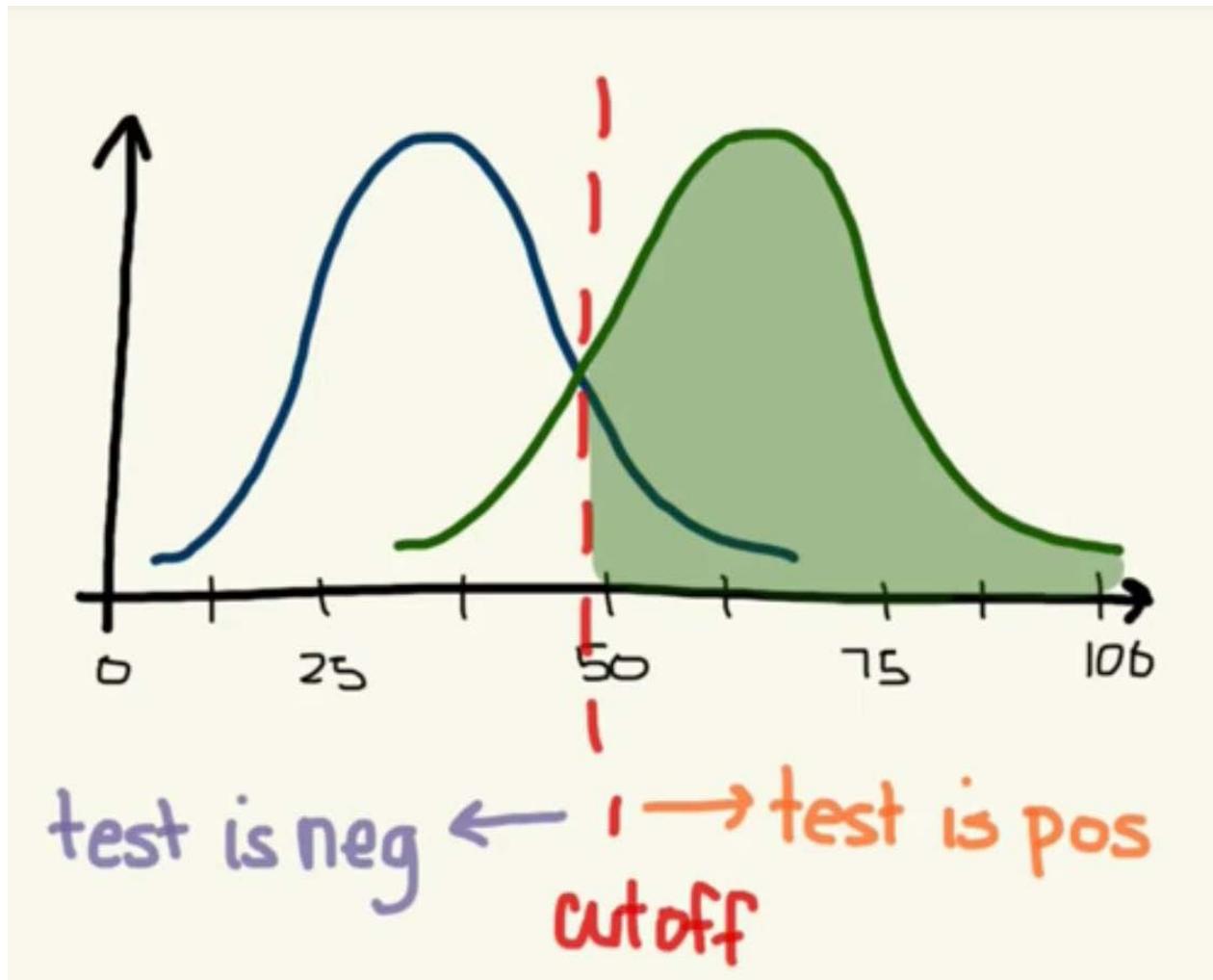
Cutoff line



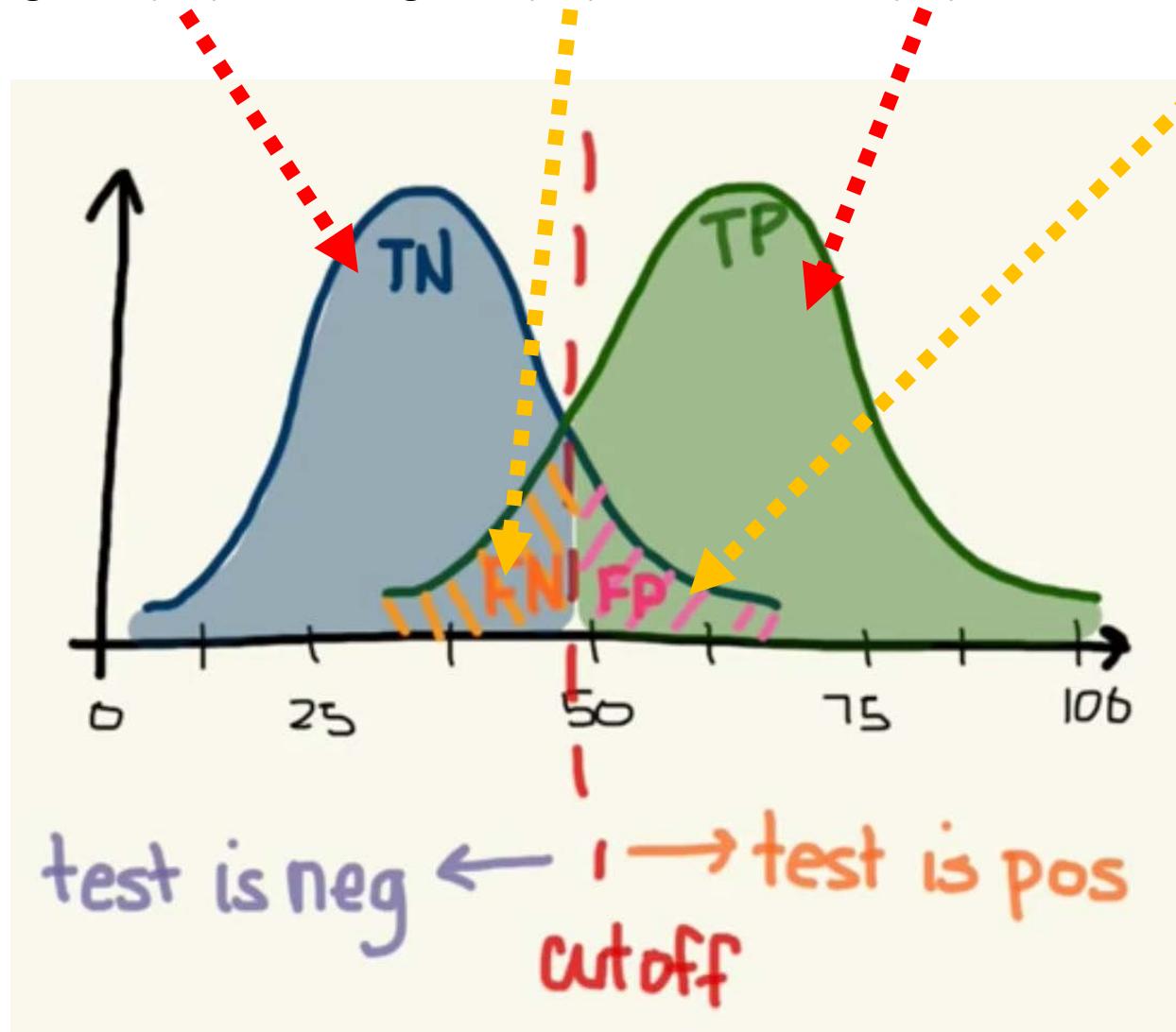
Area where the test is positive



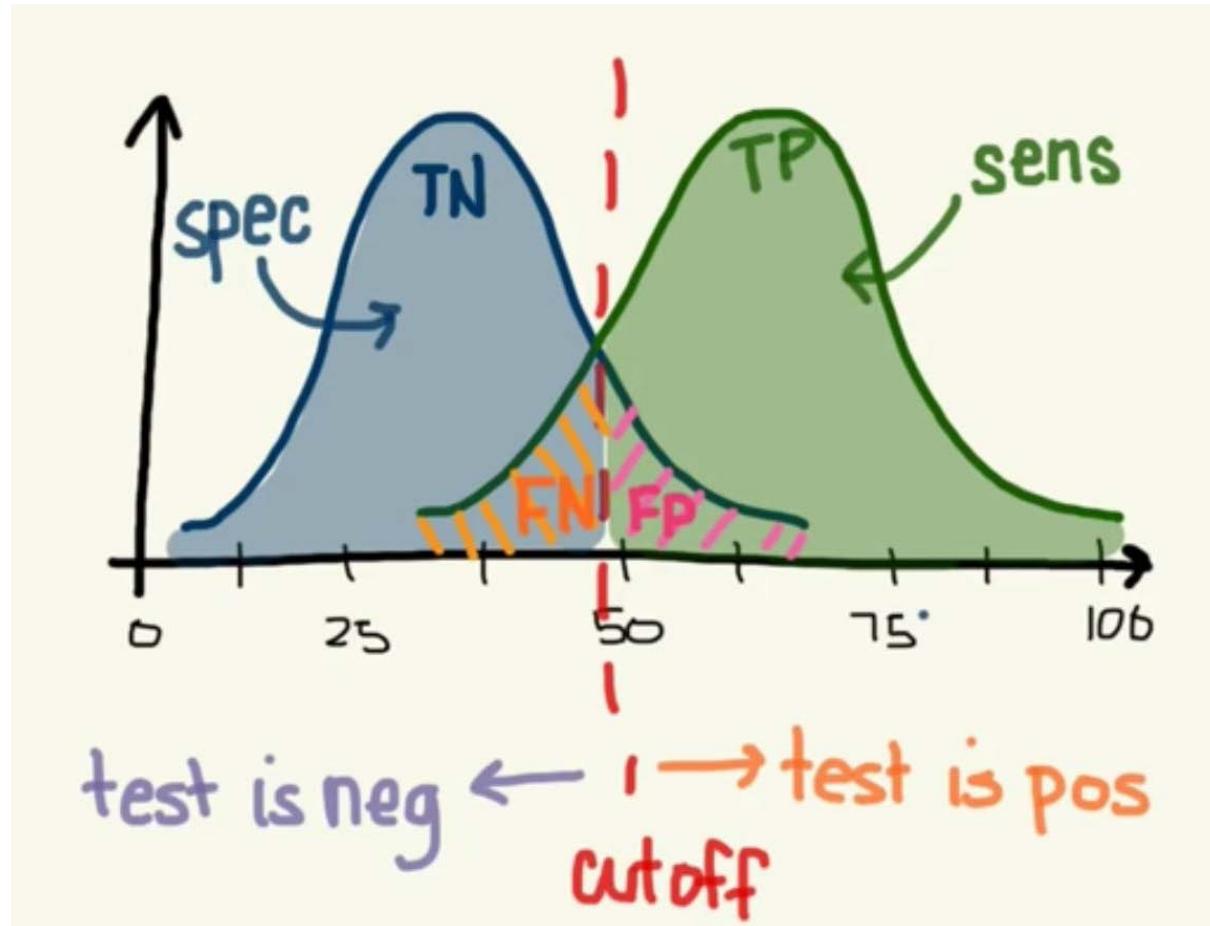
Area where the test is negative



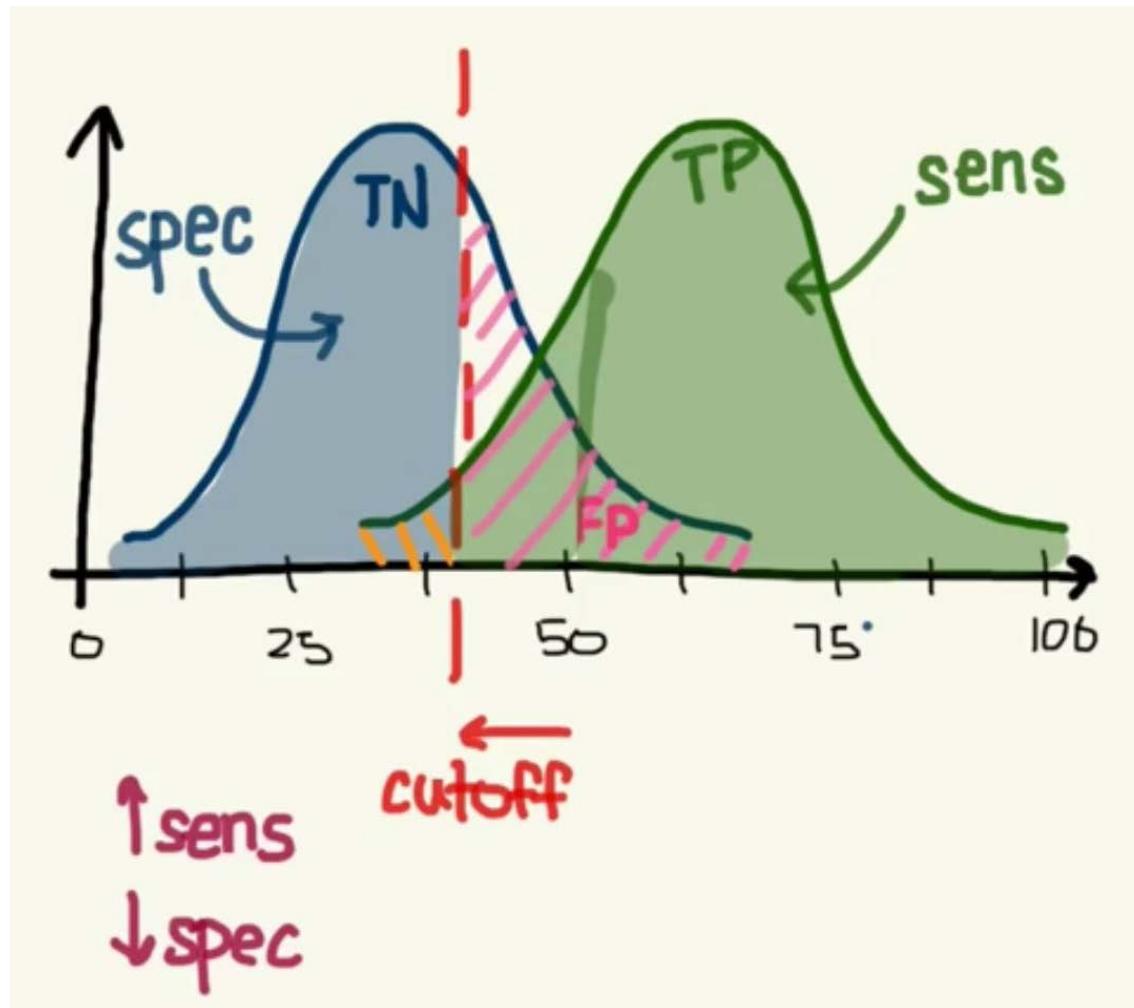
True Negative (TN), False Negative (FN) / True Positive (TP), False Positive (FP)



ROC Specificity / Sensitivity
Specificity = True Negative Rate
Sensitivity (Recall) = True Positive Rate



Move cutoff to the left Sens++ / Spec--



Move cutoff to right Sens-- / Spec++

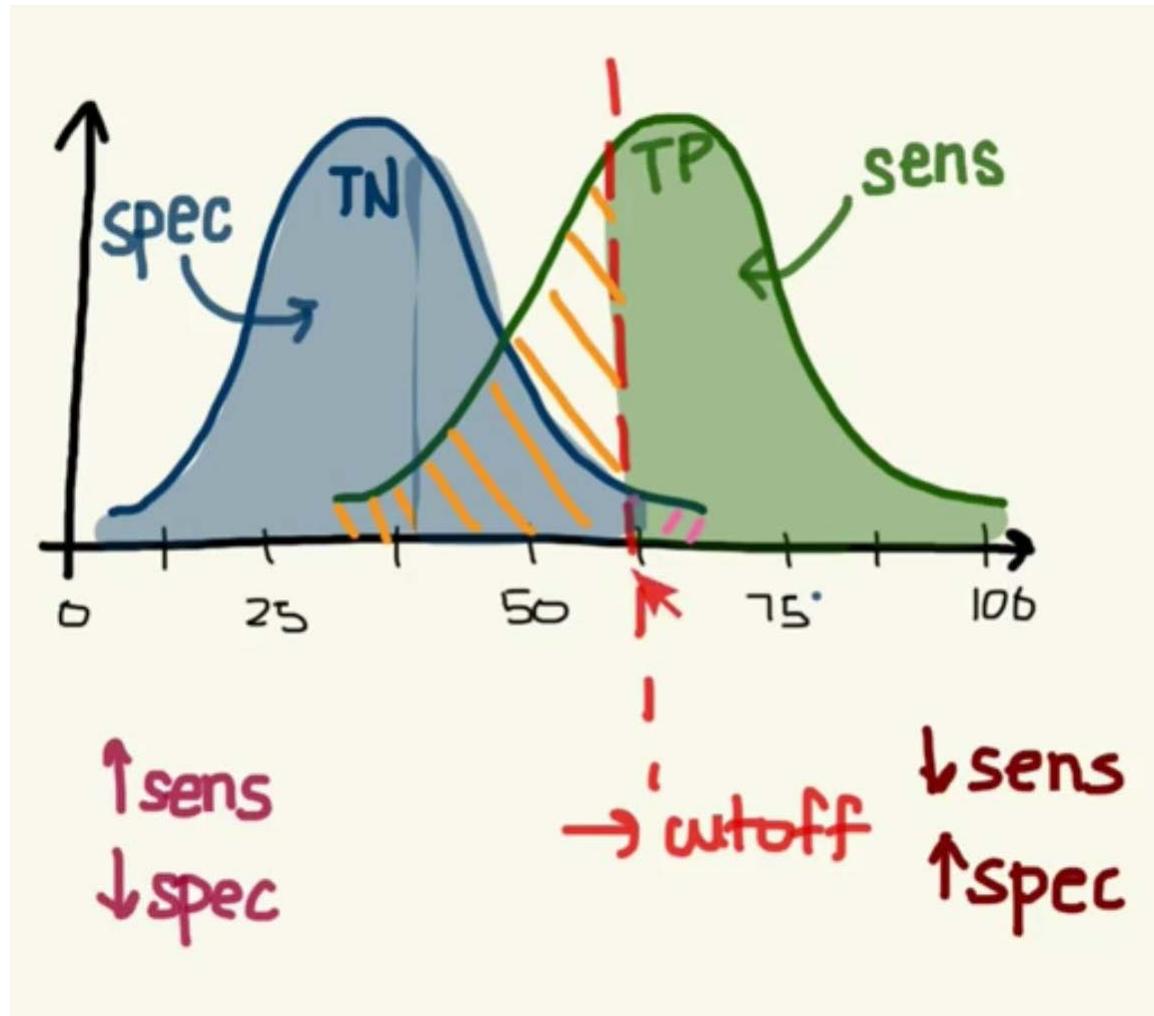
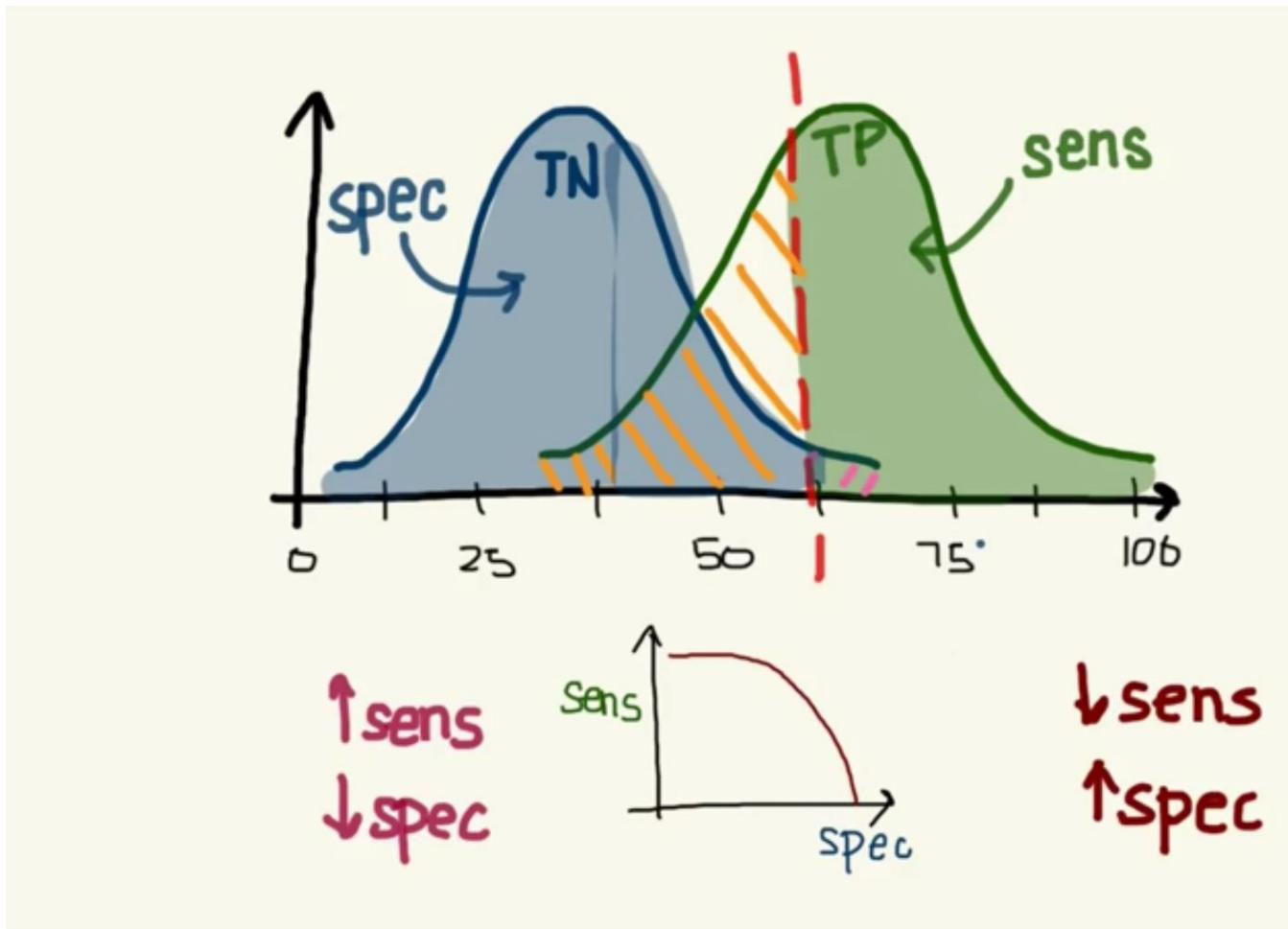
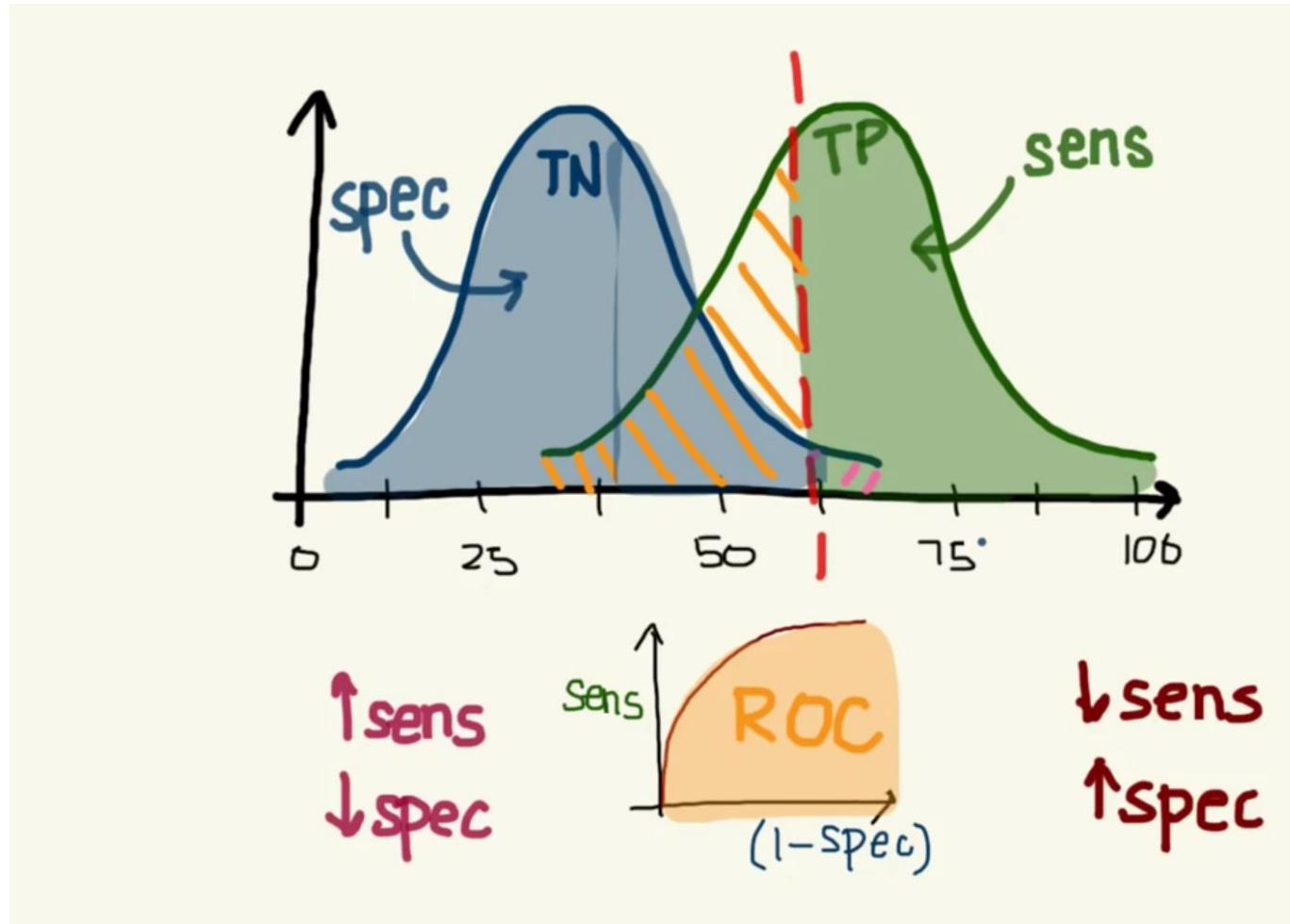


Chart proportion of Sens / Spec

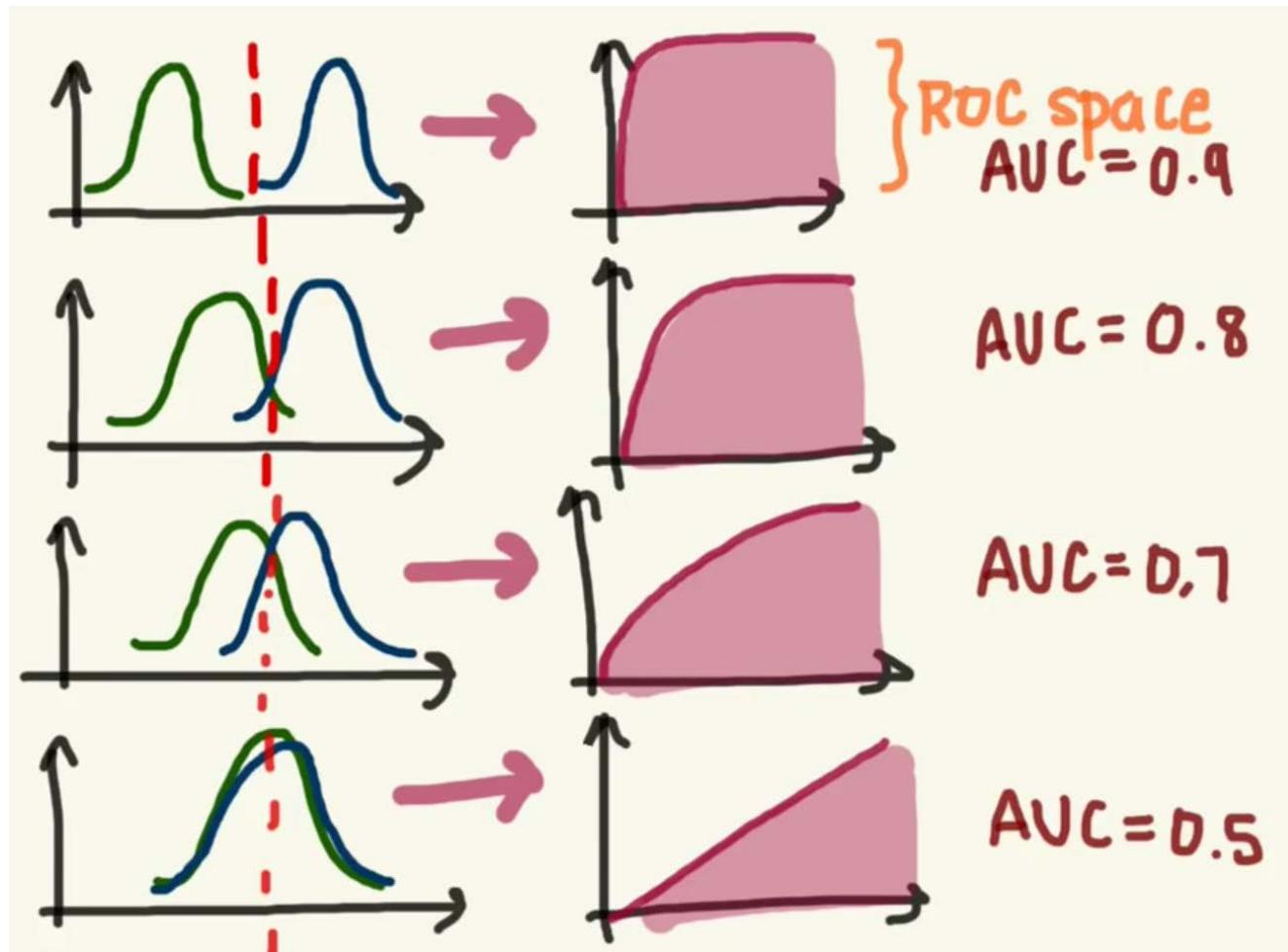


ROC curve = proportion of Sens / (1 – Spec)

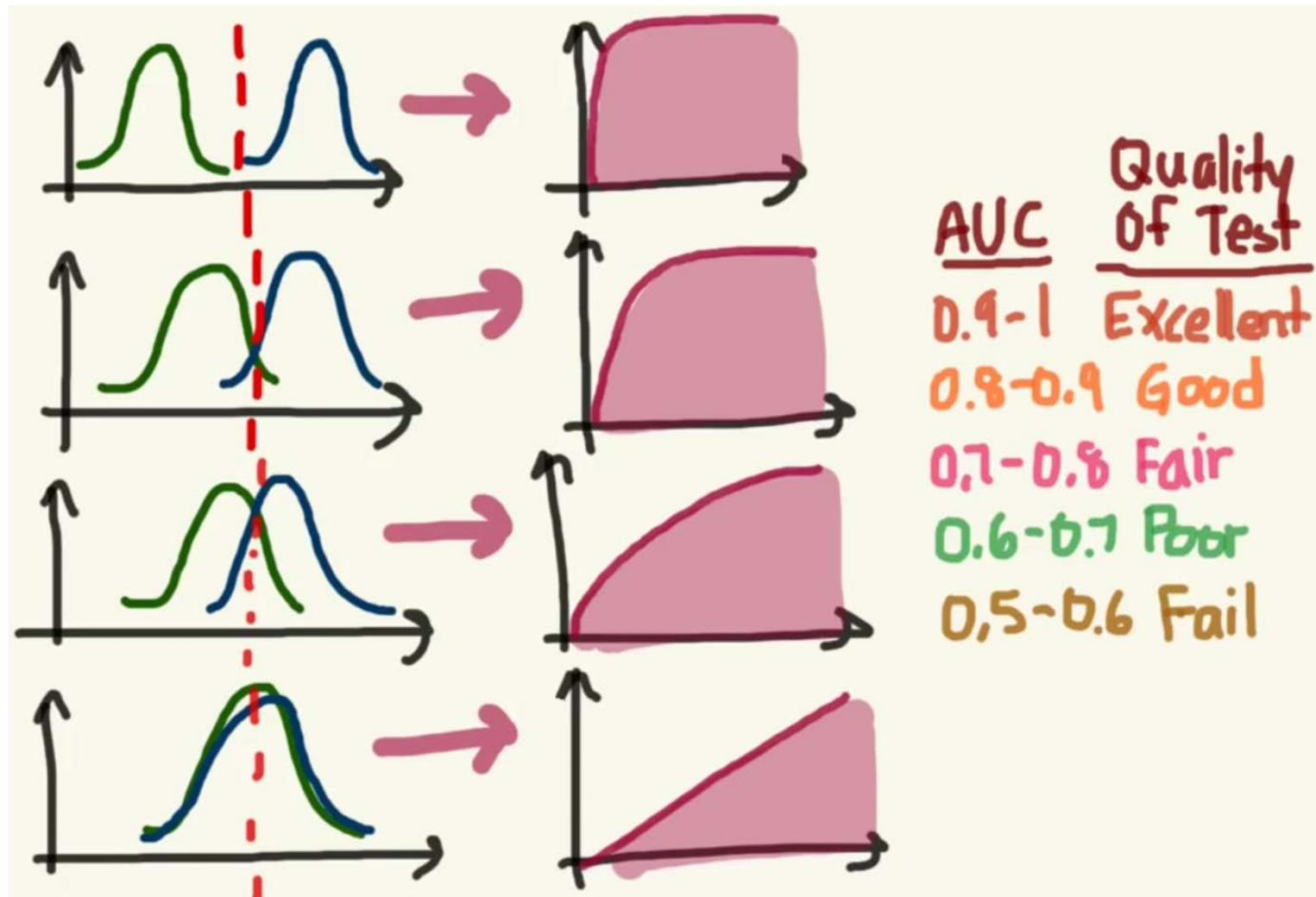


Area Under the Curve (AUC)

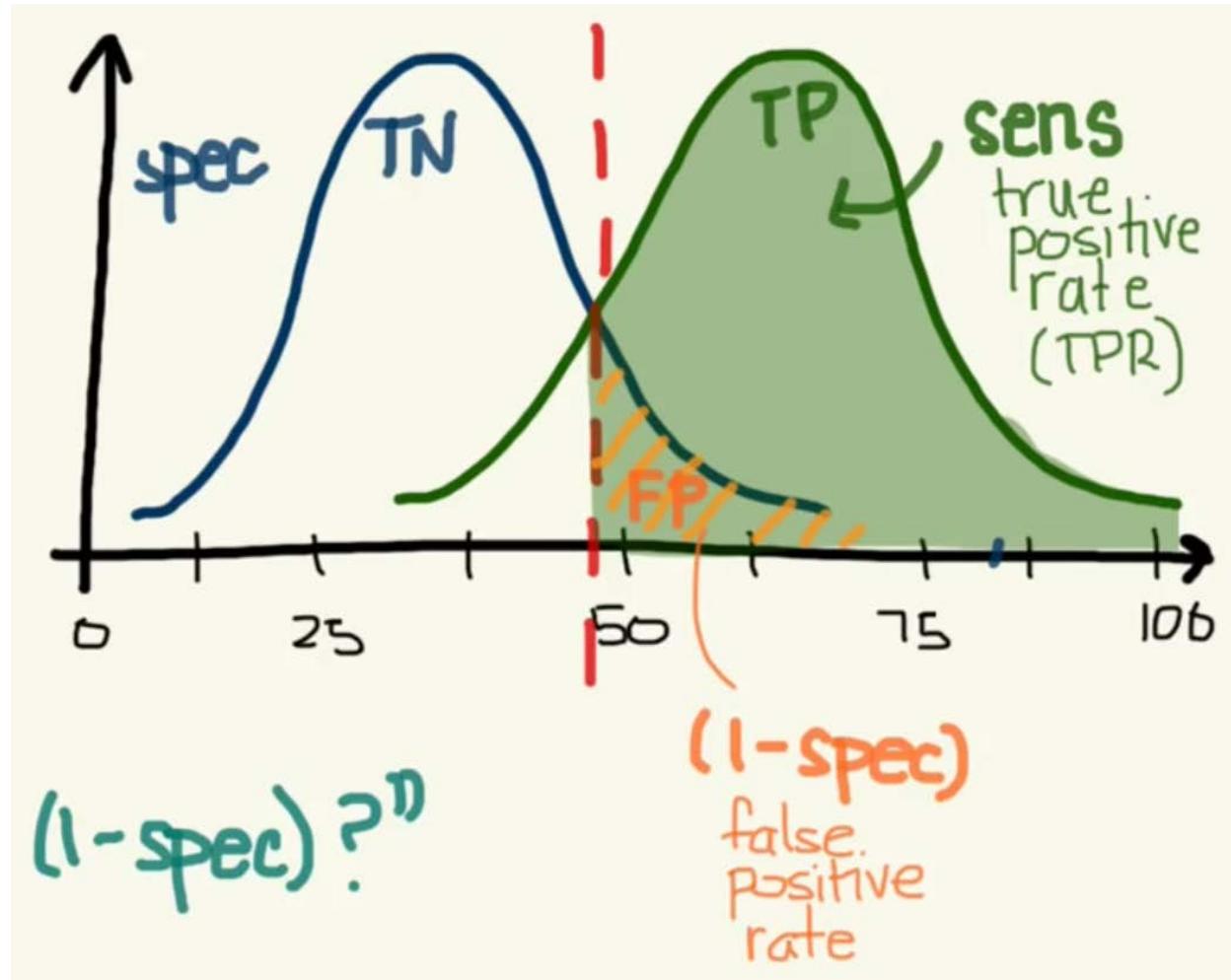
AUC is used to determine which of the used models predicts the classes best.



AUC score



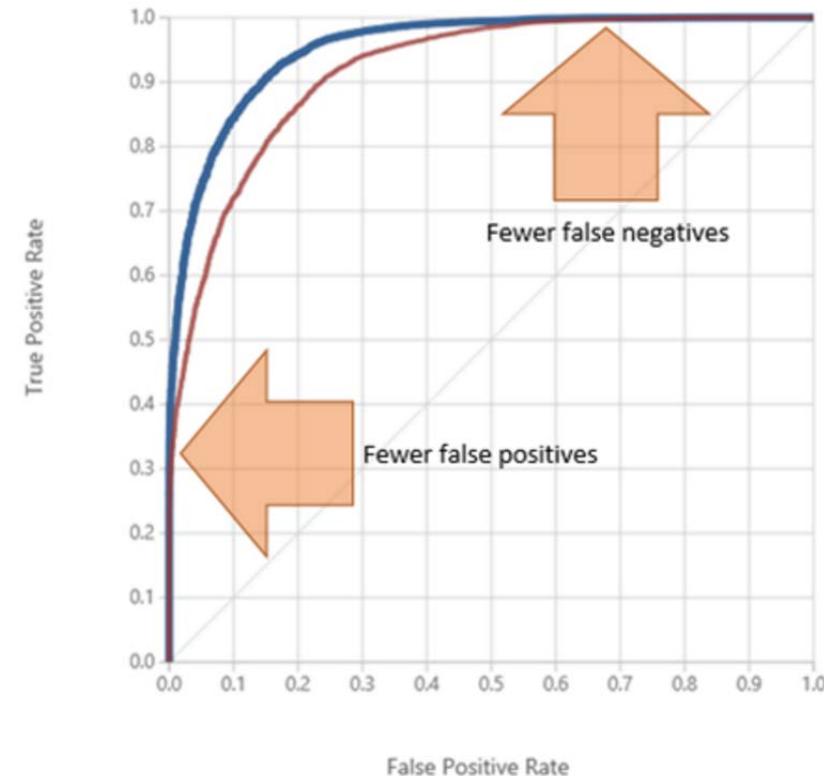
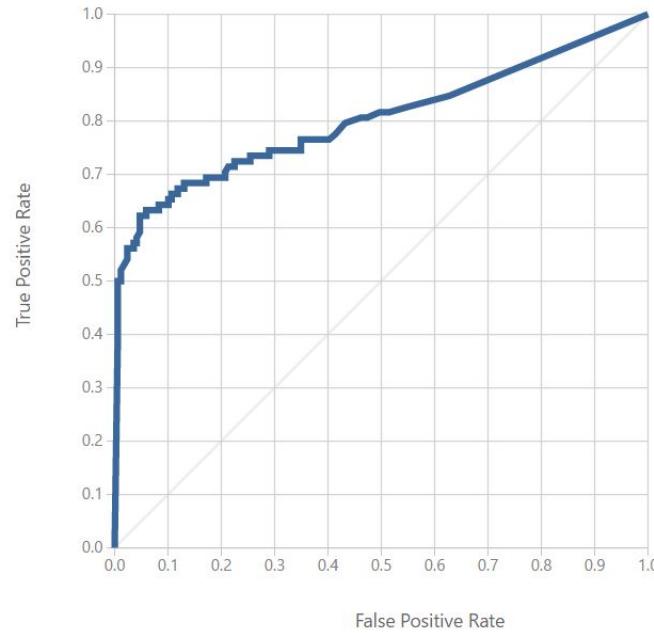
What is (1-spec)



ROC Curve

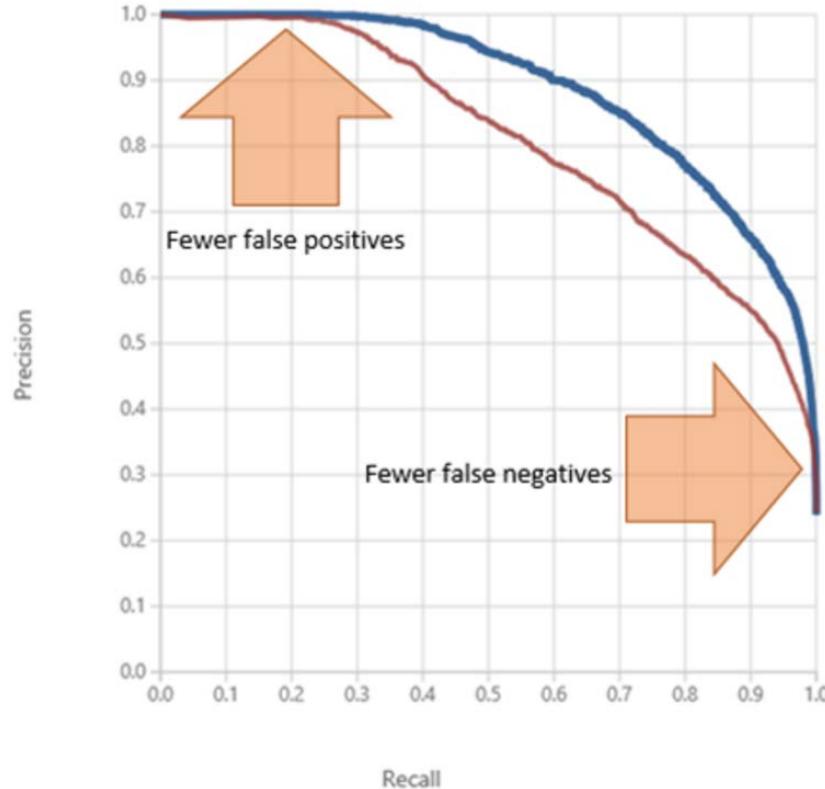
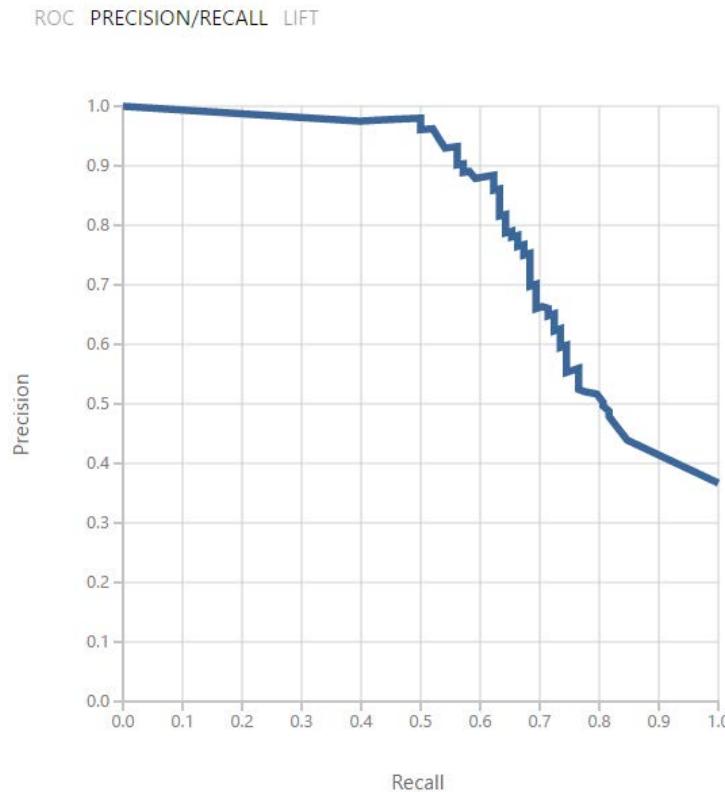
Titanic Evaluate > Evaluate Model > Evaluation results

ROC PRECISION/RECALL LIFT



ROC curve displays the fraction of true positives out of the total actual positives. The higher and further to the left, the more accurate the model is. As you do experiments you want to see the curve move higher and to the left.

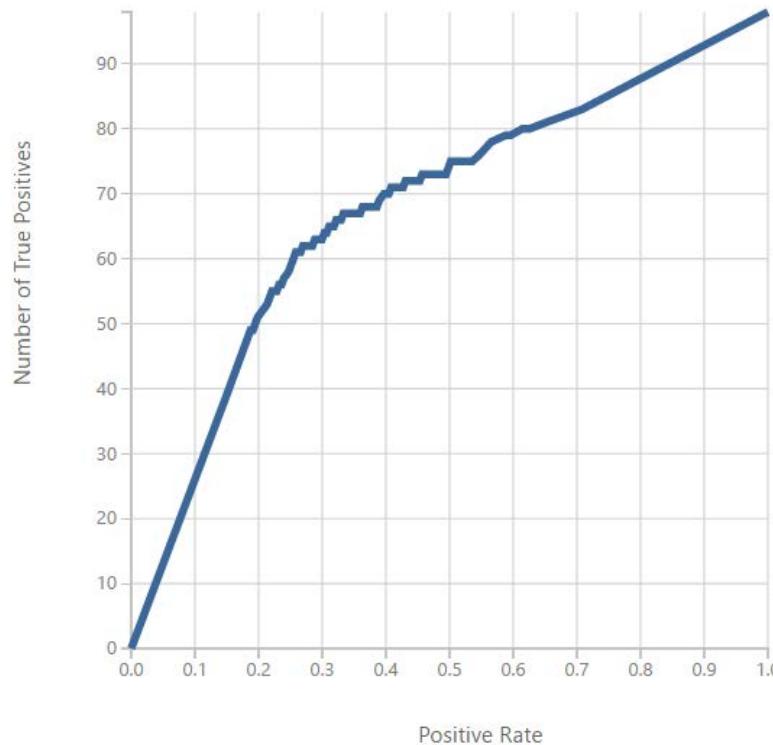
PRECISION/RECALL



Precision represents the fraction of retrieved instances that are relevant, whereas recall represents the fraction of relevant instances that are retrieved. The “sweet spot” for the ideal model is in the upper right corner

LIFT curve

ROC PRECISION/RECALL LIFT



Lift curve is a variation on the ROC curve. It measures the fraction of true positives, in relation to the target response probability.

Reading Evaluation metrics

Score Bin	Positive Examples	Negative Examples	Fraction Above Threshold
(0.900,1.000]	59	8	0.251
(0.800,0.900]	3	4	0.277
(0.700,0.800]	0	1	0.281
(0.600,0.700]	0	1	0.285

Accuracy	F1 Score	Precision	Recall	Negative Precision	Negative Recall	Cumulative AUC
0.824	0.715	0.881	0.602	0.805	0.953	0.023
0.820	0.721	0.838	0.633	0.813	0.929	0.038
0.816	0.717	0.827	0.633	0.813	0.923	0.041
0.813	0.713	0.816	0.633	0.812	0.917	0.045

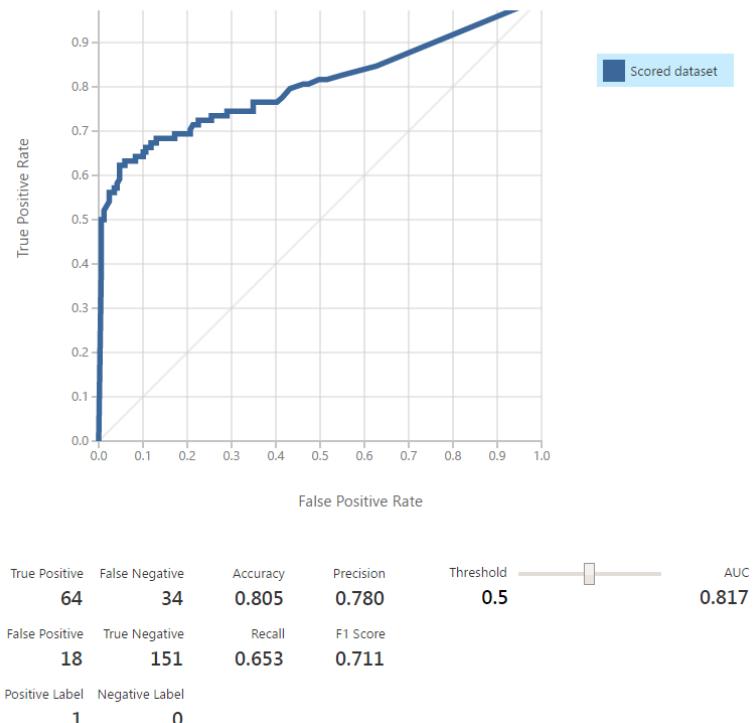
Evaluation metrics variable

- True Positive (TP): Correctly identified e.g. Sick people correctly diagnosed as sick
- False Positive (FP): Incorrectly identified e.g. healthy people incorrectly identified as sick
- True Negative (TN): Correctly rejected e.g. healthy people correctly identified as healthy
- False Negative (FN): Incorrectly rejected e.g. Sick people incorrectly identified as healthy
- Accuracy : The proportion of the total number of predictions that is correct. $(TP + TN) / (TP + TN + FP + FN)$
- Precision: is the proportion of positive cases that were correctly identified. $TP / (TP + FP)$
- Recall: Sensitivity or Recall is the proportion of actual positive cases which are correctly identified. $TP / (TP + FN)$
- F1 Score: is the harmonic mean of precision and Recall. $2TP / (2TP + FP + FN)$
- Threshold: Threshold is the value above which it belongs to first class and all other values to the second class. E.g. if the threshold is 0.5 then any patient scored more than or equal to 0.5 is identified as sick else healthy.

Titanic evaluation results

- Positive Label: 1 = survived
- Negative Label: 0 = dead
- True Positive: correctly predict survived
- True Negative: correctly predict dead
- False Positive: incorrectly predict survived
- False Negative: incorrectly predict dead

Titanic Evaluate > Evaluate Model > Evaluation results



More information

How to evaluate model performance in Azure Machine Learning

<https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-evaluate-model-performance>

This experiment ML model

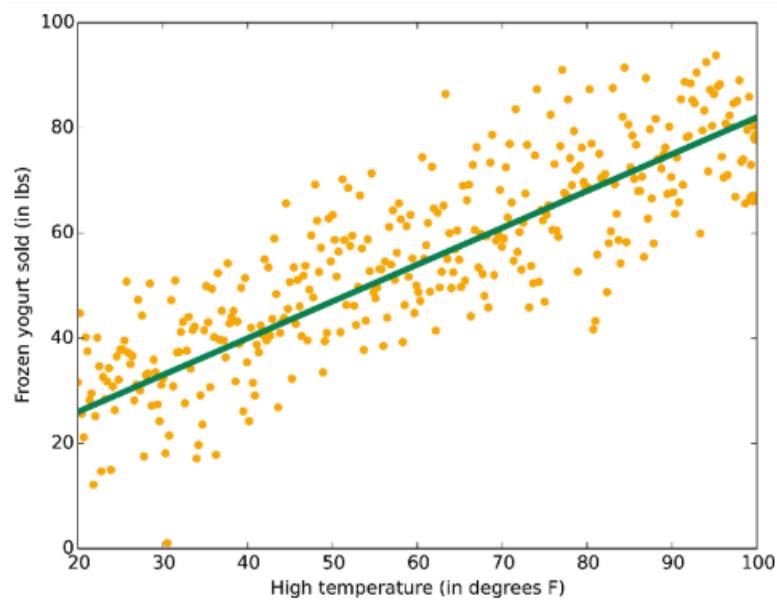
Adding Evaluation model

<https://gallery.cortanaintelligence.com/Experiment/Titanic-1-2>

Adding "Two-Class Decision Forest"

<https://gallery.cortanaintelligence.com/Experiment/Titanic-compare-two-algorithm>

ML ALGORITHM



In this session

- Supervised vs Unsupervised
- Algorithm group
- Linear regression
- Logistic regression
- Decision trees
- Neural networks
- Support vector machines (SVMs)
- Bayesian methods
- Considerations when choosing an algorithm
- Cheat Sheet
- Algorithm's performance comparison

Supervised vs Unsupervised

Supervised

- Train with know answer
- Can give answer with any new input, after sufficient training
- Create a function from inputs to give answer
- If the answers are expressed in classes, it is called classification problem
- If the answer space is continuous, it is called regression problem.

Unsupervised

- Training with unknown answer
- Can find the structure or relationships between different inputs
- Most important = clustering
- Anomaly detection

Algorithm group

Supervised: Make predictions based on a set of examples

- Classification: predict a category
- Regression: predicted a value
- Anomaly detection: identify data unusual

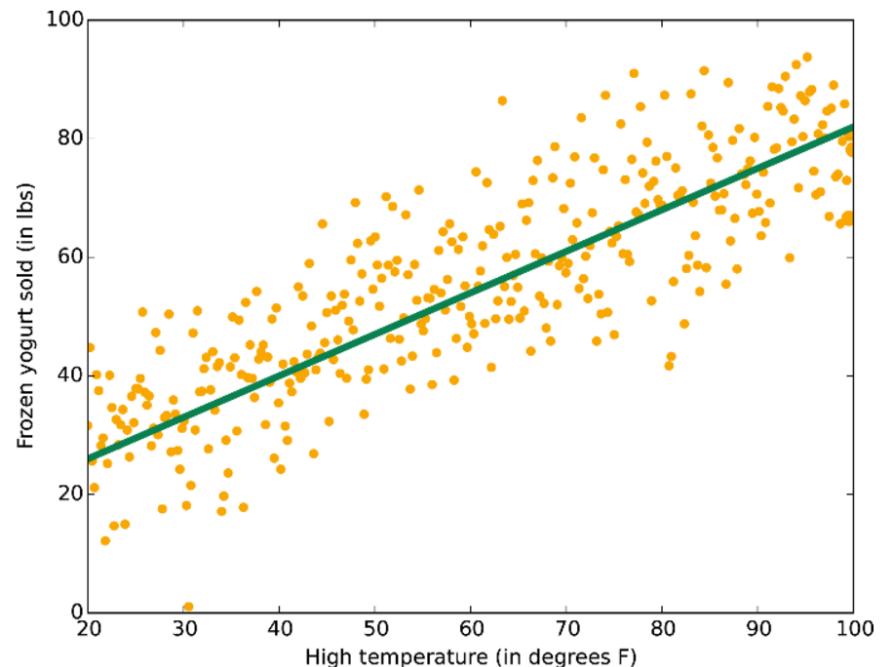
Unsupervised: data points have no labels associated with them

- Clustering: discovering structure

Linear regression

- Use when data fits a line
- It's a workhorse
- Simple and fast
- May be overly simplistic for some problems.

Higher temperature predicts better frozen yogurt sold

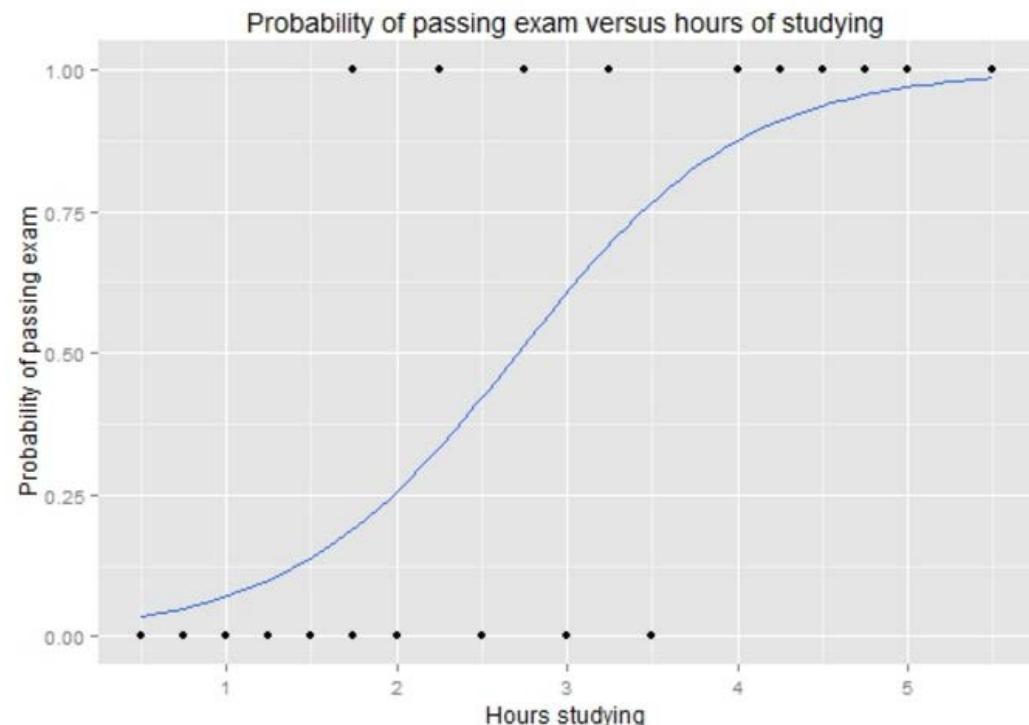


Logistic regression

- Tool for two-class and multiclass classification
- Fast and simple
- Uses an 'S'-shaped curve
- Fit for dividing data into groups
- Linear approximation

Graph of a logistic regression curve showing probability of passing an exam versus hours studying

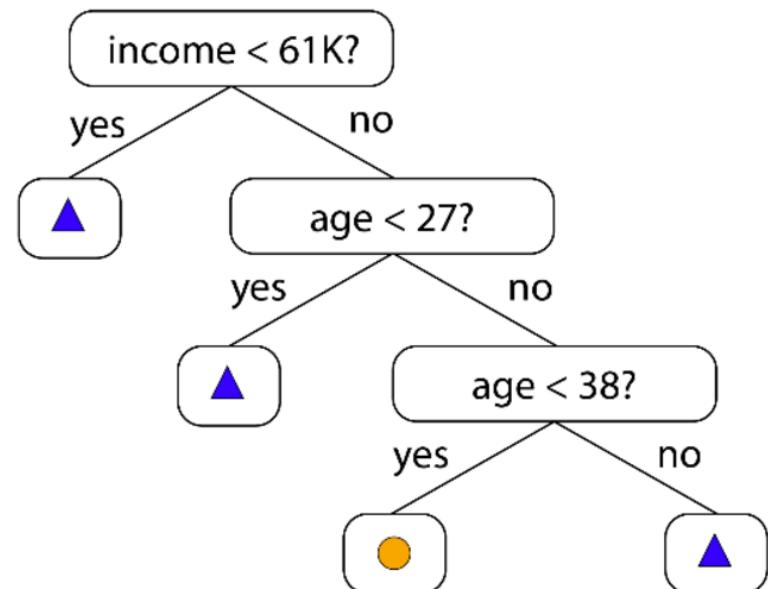
Hours of study	Probability of passing exam
1	0.07
2	0.26
3	0.61
4	0.87
5	0.97



Decision trees

- Subdivide the feature space into regions with mostly the same label
- Decision forests (regression, two-class, and multiclass)
- Decision jungles (two-class and multiclass)
- Boosted decision trees (regression and two-class)
- Foundational machine learning concept

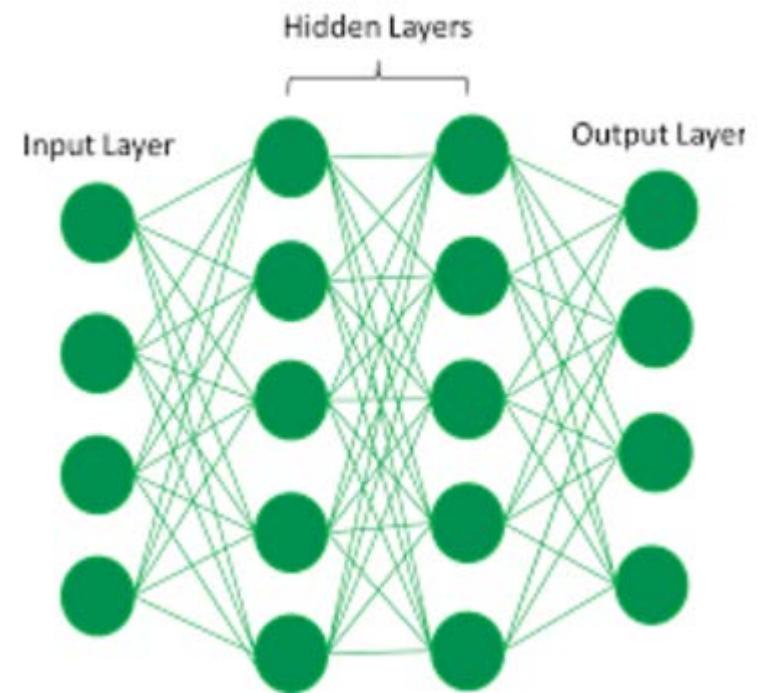
A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences



Neural networks

- Brain-inspired
- Multiclass, two-class, and regression
- Many-layered networks = "deep learning"
- Take a long time to train
- Have more parameters

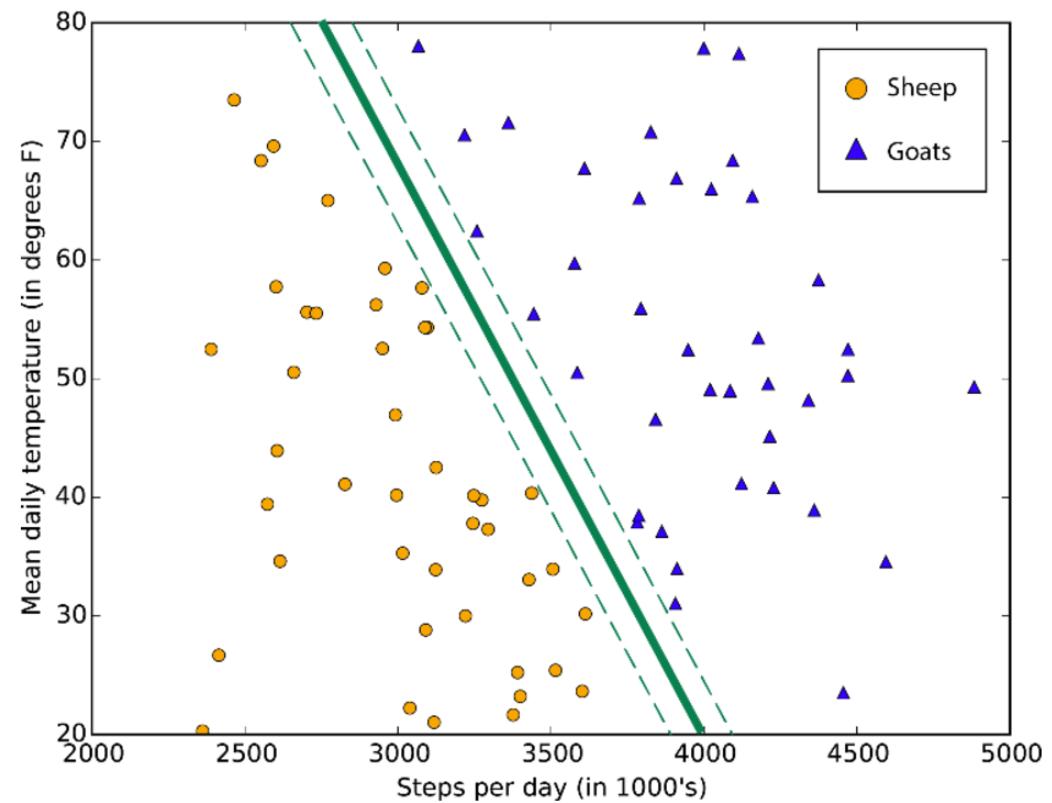
Deep learning use a cascade of many layers of nonlinear processing units for feature extraction and transformation



Support vector machines (SVMs)

- Find the boundary that separates classes
- When the two classes can't be clearly separated
- Uses a linear kernel
- Run fairly quickly
- Feature-intense data (DNA)
- Requiring only a modest amount of memory

A typical support vector machine class boundary maximizes the margin separating two classes



Bayesian methods

- Make the assumption of data points
- One data point is related with others
- Number of minutes until the next subway train arrives
- Two measurements taken a day apart are independent
- Two measurements taken a minute apart are not independent
- The value is highly predictive

This expression describes how an existing belief (“prior”) held before any evidence is considered, is updated by the evidence to produce a new level of belief (“posterior”).

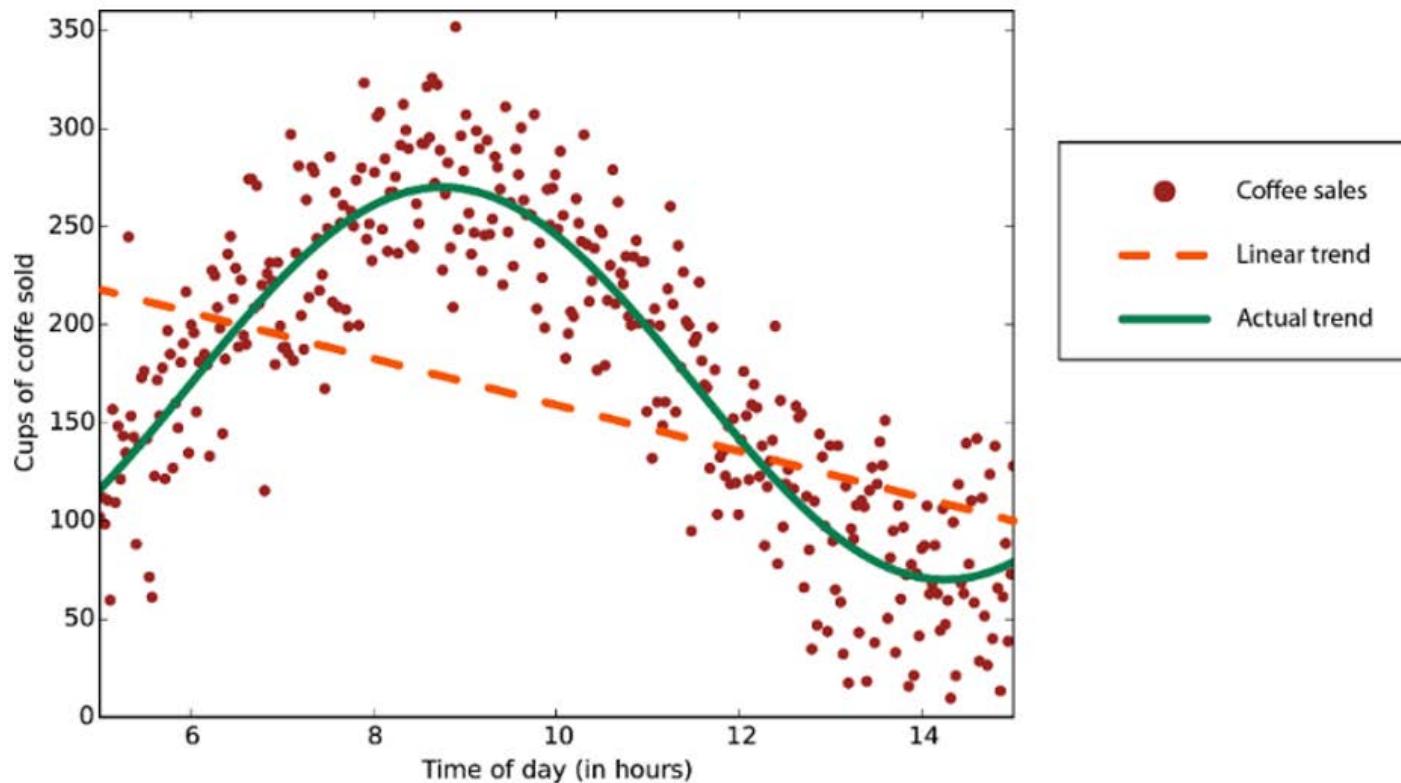
The Posterior	The Evidence	The Prior
$P(H E)$	The probability of getting this evidence if this hypothesis were true	The probability of H being true, before gathering evidence
$\frac{P(H E) P(H)}{P(E)}$	The probability that the hypothesis (H) is true given the evidence (E)	The marginal probability of the evidence (Prob of E over all possibilities)

Considerations when choosing an algorithm

Considerations when choosing an algorithm

- Accuracy: most accurate isn't always necessary
- Training time: more accuracy = longer time
- Linearity: most are liner but not always

Considerations when choosing an algorithm

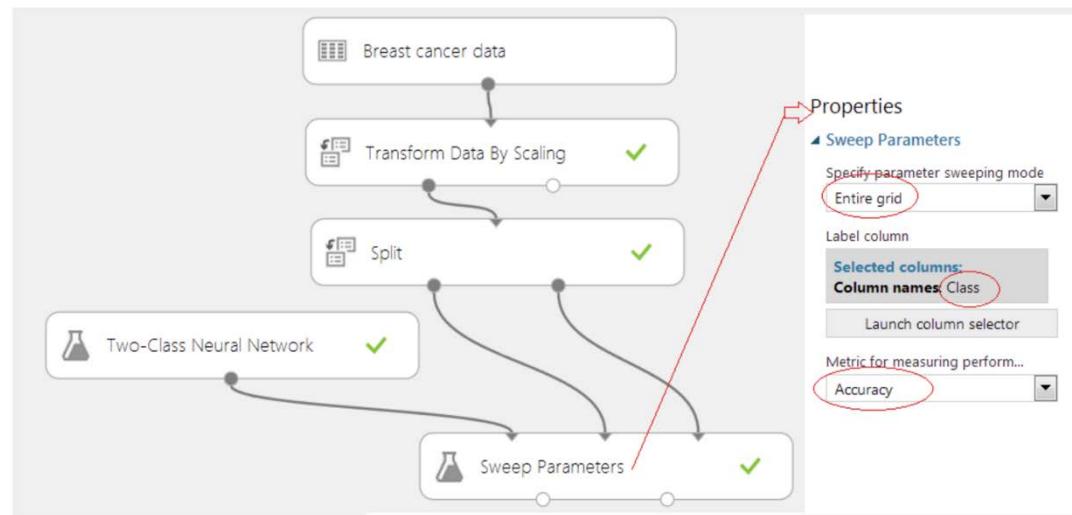


Data with a nonlinear trend - using a linear regression method would generate much larger errors than necessary

Considerations when choosing an algorithm

Algorithm's parameters

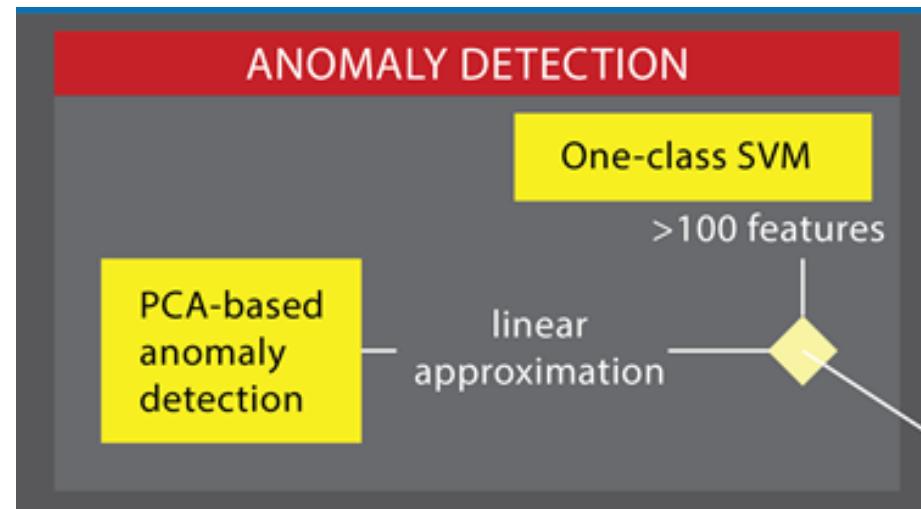
- Are the knobs a data scientist turns when setting up an algorithm
- Affect the algorithm's behavior
- Must understand the in-side out of algorithm
- Use parameter sweeping to automatically tries all parameter



Considerations when choosing an algorithm

Number of features

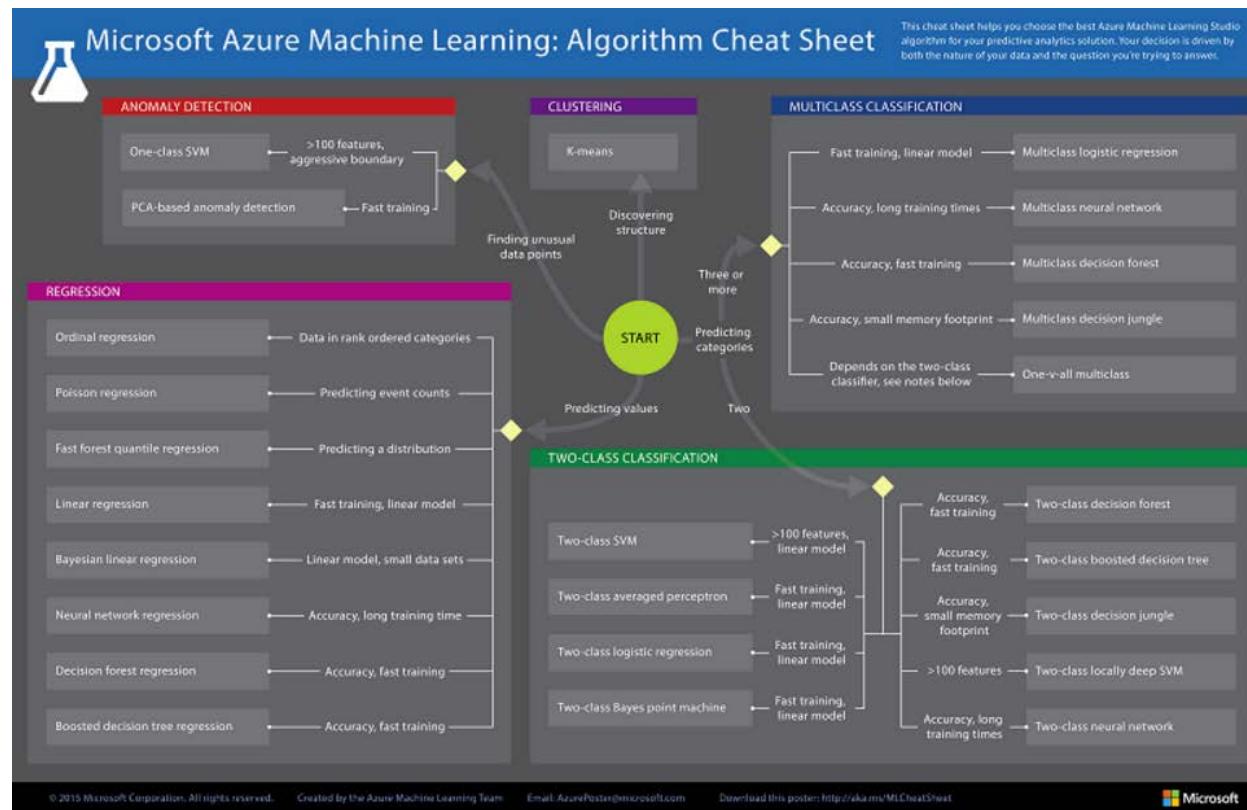
- Can be very large for genetics or textual data
- The large number can bog down some algorithms
- Making training time long
- Go deep
- Support Vector Machines (SVM)

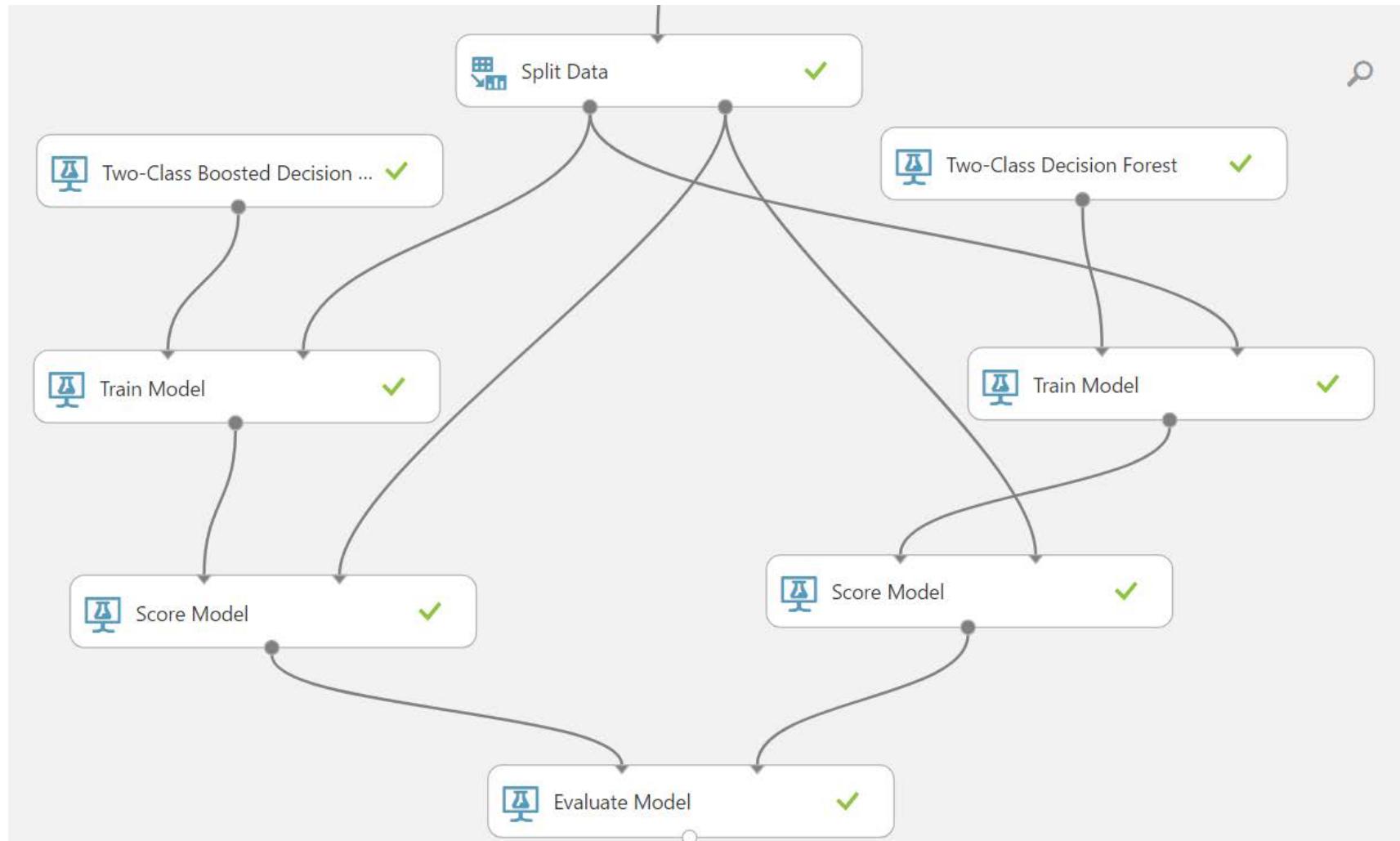


Cheat Sheet

Machine Learning Algorithm Cheat Sheet (11x17 in.)

<http://download.microsoft.com/download/A/6/1/A613E11E-8F9C-424A-B99D-65344785C288/microsoft-machine-learning-algorithm-cheat-sheet-v6.pdf>





Algorithm's performance comparison

Algorithm's performance comparison

1. Open Experiment Titanic
2. Save as Titanic two algorithm
3. Drag & drop modules
 - a. Two-Class Decision Forest module
 - b. Train Module
 - c. Score Module
4. Set module properties
5. Save Experiment
6. Run Experiment
7. View Visualize / ROC Curve and Evaluation metrics

Algorithm's performance comparison

Modules properties setting

▲ Two-Class Decision Forest

Resampling method

Bagging

Create trainer mode

Single Parameter

Number of decision trees

8

Maximum depth of the decision trees

32

Number of random splits per node

128

Minimum number of samples per le...

1

Allow unknown values for categ...

▲ Train Model

Label column

Selected columns:
Column names: Survived

[Launch column selector](#)

START TIME 6/11/2017 1:51:50 PM
END TIME 6/11/2017 1:51:55 PM
ELAPSED TIME 0:00:04.620
STATUS CODE Finished
STATUS DETAILS None

[View output log](#)

▲ Score Model

<input checked="" type="checkbox"/> Append score columns to output	6/11/2017 1:51:56 PM
START TIME	6/11/2017 1:51:56 PM
END TIME	6/11/2017 1:51:59 PM
ELAPSED TIME	0:00:03.084
STATUS CODE	Finished
STATUS DETAILS	None

More information

A Tour of Machine Learning Algorithms

<http://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/>

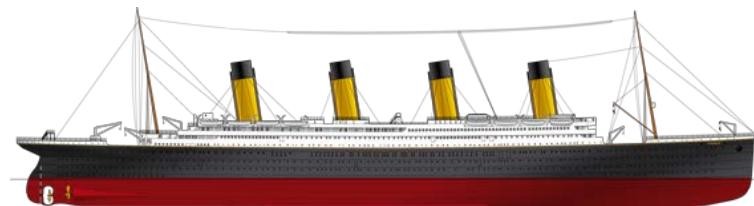
The screenshot shows the homepage of machinelearningmastery.com. At the top is a navigation bar with a menu icon and the word "Navigation". Below it is a logo featuring a stylized head with gears and the text "MACHINE LEARNING MASTERY". The main menu includes "Start Here", "Blog", "Books", "About", and "Contact". A search bar with a magnifying glass icon is positioned below the menu. A call-to-action button at the bottom left encourages users to "Get Your Start in Machine Learning". The central content area features the title "A Tour of Machine Learning Algorithms" by Jason Brownlee on November 25, 2013, in Machine Learning Algorithms. Below the title are social sharing buttons for Facebook, Twitter, Google+, LinkedIn, and StumbleUpon, along with a count of 2188 shares. A brief summary of the post follows.

In this post, we take a tour of the most popular machine learning algorithms.

It is useful to tour the main algorithms in the field to get a feeling of what methods are available.

Get Your Start in Machine Learning It can feel overwhelming when algorithm names are thrown around without explanation. This tour will help you understand the basic concepts behind the most common machine learning algorithms.

C# MACHINE LEARNING APPLICATION



In this session

- Titanic Survival Predictor Web app
- Titanic Survival Predictor C# Win App
- Create User Interface
- Add class Titanic
- Titanic unit test
- Adding code to Class Form1

Titanic Survival Predictor Web app

<http://demos.datasciencedojo.com/demo/titanic/>

The screenshot shows the Data Science Dojo's Titanic Survival Predictor. The main heading is "Titanic Survival Predictor". Below it is a descriptive text: "Would you survive the Titanic disaster? Given such information as your gender, age, and accommodation class, we'll be able to figure out your odds of survival from the doomed liner." To the left is a form with the following fields:

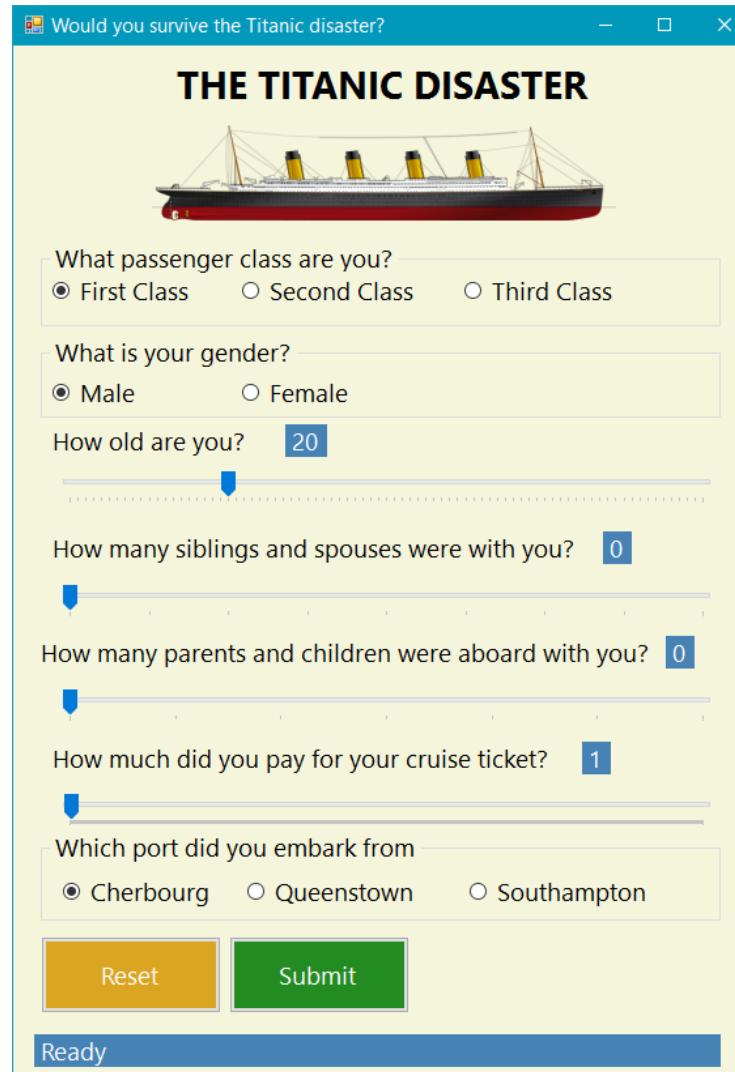
- What passenger class are you?
First Class | Second Class | Third Class
- What is your gender?
Male | Female
- How old are you?
Age slider from 0 to 80, currently at 9.
- How many siblings and spouses were with you?
Siblings/Spouses slider from 0 to 8, currently at 0.
- How many parents and children were aboard with you?
Parents/Children slider from 0 to 6, currently at 0.
- How much did you pay for your cruise ticket? (in 1910 USD)
Ticket Price slider from \$0 to \$512, currently at \$359.
- Which port did you embark from?
Cherbourg | Queenstown | Southampton

To the right is a "PREDICTION TABLE" section:

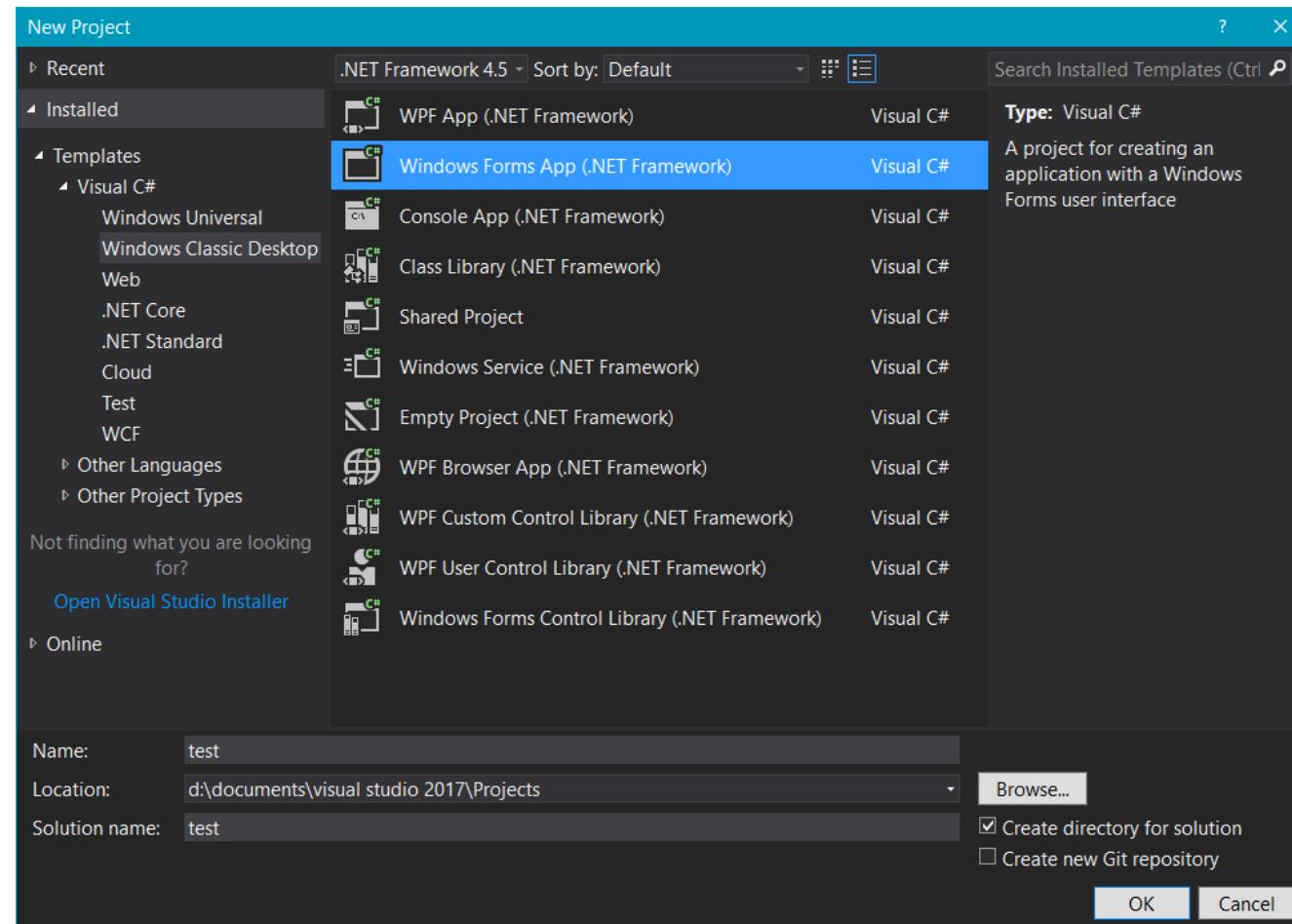
Chances of Survival	Prediction	Message
79.71%	Survived	Congratulations you made it out!
79.71%	Survived	Congratulations you made it out!

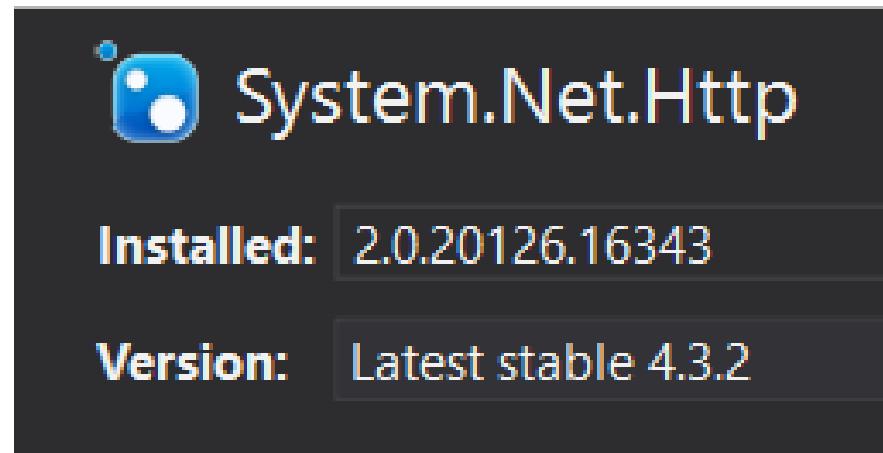
Show 10 entries | Search: [] | Previous | Next | Clear Tables

Titanic Survival Predictor C# Win App



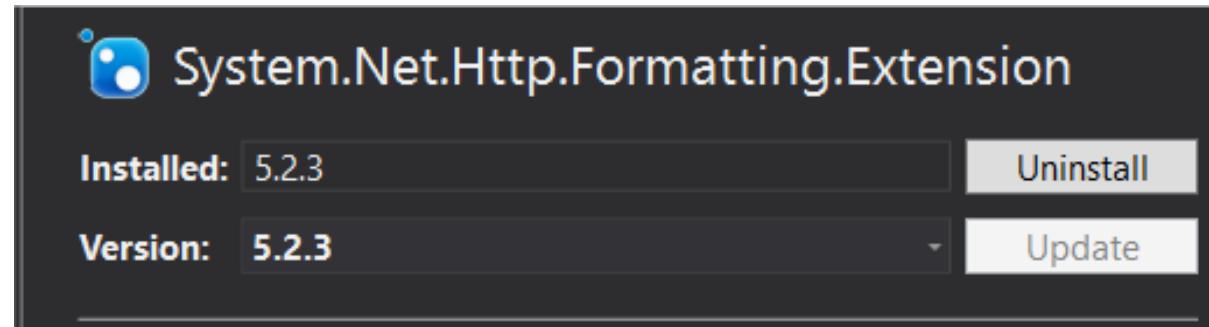
Name = Test
Location = d:\temp





1. Right click at project name
2. Click Manage NuGet Packages
3. Click Browse
4. Enter System.Net.Http in to the search box
5. Click Install

Add package System.Net.Http.Formatting.Extension



1. Right click at project name
2. Click Manage NuGet Packages
3. Click Browse
4. Enter System.Net.Http.Formatting.Extension in to the search box
5. Click Install

Create User Interface

Control naming convention

- radioButtonFirstClass
- radioButtonSecondClass
- radioButtonThirdClass
- radioButtonFemale
- radioButtonMale
- trackBarAge
- labelAge
- trackBarSib
- labelSib
- trackBarPar
- labelPar
- trackBarPay
- labelPay
- radioButtonCher
- radioButtonSout
- radioButtonQueens
- buttonReset
- buttonSubmit
- labelResult

Add class Input

1. Add new Class
2. Name = Input

```
10  //</remarks>
11  namespace test
12  {
13      public class Input...
14      public class StringTable
15      {
16          public string[] ColumnNames { get; set; }
17          public string[,] Values { get; set; }
18      }
19  }
```

```
13  public class Input
14  {
15      public int PassengerId { get; set; }
16      public int Survived { get; set; }
17      public int Pclass { get; set; }
18      public string Name { get; set; }
19      public string Sex { get; set; }
20      public int Age { get; set; }
21      public int SibSp { get; set; }
22      public int Parch { get; set; }
23      public string Ticket { get; set; }
24      public int Fare { get; set; }
25      public string Cabin { get; set; }
26      public string Embarked { get; set; }
```

Add class Input

Add Constructor and Clear methods

```
27     public Input()
28     {
29         Clear();
30     }
31     public void Clear()
32     {
33         PassengerId = 0;
34         Survived = 0;
35         Pclass = 0;
36         Name = "value";
37         Sex = "0";
38         Age = 1;
39         SibSp = 0;
40         Parch = 0;
41         Ticket = "value";
42         Fare = 0;
43         Cabin = "value";
44         Embarked = "value";
45     }
46 }
```

Add class Titanic

```
12  using System;
13  using System.Collections.Generic;
14  using System.Threading.Tasks;
15  using System.Net.Http;
16  using System.Net.Http.Headers;
17
18  namespace test
19  {
20      public class Titanic
21      {
22          public string resultMessage { get; set; }
23          const string apiKey = "key";
24          // Replace above with your API key for the web service
25          const string baseAddress = "url";
26          private string[] inputColumnName =...;
27
28
29
30          private void CreateRequestBody(ref string[] cn, ref string[,] v, Input input)...  
31
32
33
34
35          private string GetResult(string result)...  
36
37
38
39          public async Task<T> GetPrediction<T>(Input input)...  
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111 }
```

Add class Titanic

Copy and paste API key from Web Service page Titanic 2 [predictive exp.]

The screenshot shows the Microsoft Azure Machine Learning Studio interface. On the left is a vertical sidebar with icons for different experiment components. The main area is titled "titanic 2 [predictive exp.]". It has tabs for "DASHBOARD" and "CONFIGURATION". Under "General", there's a link to "New Web Services Experience" (preview). Below that are links for "Published experiment", "View snapshot", and "View latest". There's also a "Description" section with the note "No description provided for this web service." A large yellow arrow points to the "API key" field, which contains a long, randomly generated string of characters. At the bottom, there are sections for "REQUEST/RESPONSE" and "BATCH EXECUTION", each with a "Test" button. A "TEST" tab is selected. The "LAST UPDATED" section shows two entries: "Excel 2013 or later | Excel 2010 or earlier" updated on 6/4/2017 1:20:12 PM, and "Excel 2013 or later workbook" updated on 6/4/2017 1:20:12 PM.

Add class Titanic

Copy and paste Base Address from Request Response API Documentation for Titanic 2 [Predictive Exp.]

```
        var scoreRequest = new
        {
            Inputs = new Dictionary<string, StringTable> () {
                {
                    "input1",
                    new StringTable()
                    {
                        ColumnNames = new string[] {"PassengerId", "Survived", "Pclass", "Name", "Sex", "Age", "SibSp",
                        Values = new string[,] { { "0", "0", "0", "value", "value", "0", "0", "0", "value", "0", "value" }
                    }
                },
                },
                GlobalParameters = new Dictionary<string, string>() {
                };
            const string apiKey = "abc123";           place this with the API key for the web service
            client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue( "Bearer", apiKey);
            client.BaseAddress = new Uri("https://ussouthcentral.services.azureml.net/workspaces/ede12cb3aaaf24c7e826493f4e30
            // WARNING: The 'await' statement below can result in a deadlock if you are calling this code from the UI thread
            // One way to address this would be to call ConfigureAwait(false) so that the execution does not attempt to res
            // For instance, replace code such as:
            //     result = await DoSomeTask()
            // with the following:
            //     result = await DoSomeTask().ConfigureAwait(false)
```

Add class Titanic

inputColumnName array class member

```
26     |    private string[] inputColumnName = {  
27     |        "PassengerId", "Survived", "Pclass", "Name", "Sex", "Age",  
28     |        "SibSp", "Parch", "Ticket", "Fare", "Cabin", "Embarked" };
```

Add class Titanic

Method CreateRequestBody

```
30     private void CreateRequestBody(ref string[] cn, ref string[,] v, Input input)
31     {
32         cn = inputColumnName;
33         v = new string[,] 
34         {
35             {
36                 input.PassengerId.ToString(),
37                 input.Survived.ToString(),
38                 input.Pclass.ToString(),
39                 input.Name.ToString(),
40                 input.Sex.ToString(),
41                 input.Age.ToString(),
42                 input.SibSp.ToString(),
43                 input.Parch.ToString(),
44                 input.Ticket.ToString(),
45                 input.Fare.ToString(),
46                 input.Cabin.ToString(),
47                 input.Embarked.ToString()
48             },
49             {
50                 input.PassengerId.ToString(),
51                 input.Survived.ToString(),
52                 input.Pclass.ToString(),
53                 input.Name.ToString(),
54                 input.Sex.ToString(),
55                 input.Age.ToString(),
56                 input.SibSp.ToString(),
57                 input.Parch.ToString(),
58                 input.Ticket.ToString(),
59                 input.Fare.ToString(),
60                 input.Cabin.ToString(),
61                 input.Embarked.ToString()
62             }
63         };
64     }
```

Add class Titanic

Method GetResult

```
"{\\"Results\\":{\\\"output1\\\":{\\\"type\\\":\\\"table\\\",\\\"value\\\":{\\\"ColumnNames\\\":[],\\\"Survived\\\",\\\"PassengerClass\\\",\\\"Gender\\\",\\\"Age\\\",\\\"Sibl\\ngSpouse\\\",\\\"ParentChild\\\",\\\"FarePrice\\\",\\\"PortEmbarkation\\\",\\\"Scored Labels\\\",\\\"Scored\\nProbabilities\\\"},\\\"ColumnTypes\\\":[],\\\"Int32\\\",\\\"Int32\\\",\\\"String\\\",\\\"Double\\\",\\\"Int32\\\",\\\"Int32\\\",\\\"Double\\\",\\\"String\\\",\\\"Int32\\\",\\\"Double\\\"},\\\"Values\\\":[[\\\"0\\\",\\\"1\\\",\\\"0\\\",\\\"1\\\",\\\"0\\\",\\\"0\\\",\\\"0\\\",\\\"C\\\",\\\"1\\\",\\\"0.997975647449493\\\"],[\\\"0\\\",\\\"1\\\",\\\"0\\\",\\\"1\\\",\\\"0\\\",\\\"0\\\",\\\"0\\\",\\\"C\\\",\\\"1\\\",\\\"0.997975647449493\\\"]]}}}
```

```
65     private string GetResult(string result)
66     {
67         string s = string.Empty;
68         var cleaned = result.Replace("\\\"", string.Empty);
69         cleaned = cleaned.Replace("[", string.Empty);
70         cleaned = cleaned.Replace("]", string.Empty);
71         cleaned = cleaned.Replace("}", string.Empty);
72         string[] ra = cleaned.Split(",".ToCharArray());
73         if (ra[39] == "0")
74             s += "You Dead.";
75         else
76             s += "You Survived.";
77         s += " Possibility = " + ra[40].Substring(0, 5) + "%.";
78         return s;
79     }
```

Method GetPrediction

```
80     public async Task GetPrediction(Input input)
81     {
82         string[] cn = new string[12];
83         string[,] v = new string[12, 12];
84         CreateRequestBody(ref cn, ref v, input);
85         var myRequest =
86         {
87
88             Inputs = new Dictionary<string, StringTable>() { { "input1", new StringTable() { ColumnNames = cn, Values = v } } },
89             GlobalParameters = new Dictionary<string, string>()
90         };
91         var client = new HttpClient();
92         client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", apiKey);
93         client.BaseAddress = new Uri(baseAddress);
94         HttpResponseMessage response = await client.PostAsJsonAsync("", myRequest);
95         if (response.IsSuccessStatusCode)
96         {
97             string result = await response.Content.ReadAsStringAsync();
98             Console.WriteLine("Result: {0}", result);
99             resultMessage = GetResult(result);
100        }
101    else
102    {
103        Console.WriteLine(string.Format("The request failed with status code: {0}", response.StatusCode));
104        // Print the headers - they include the request ID and the timestamp, which are useful for debugging the failure
105        Console.WriteLine(response.Headers.ToString());
106        string responseContent = await response.Content.ReadAsStringAsync();
107        Console.WriteLine(responseContent);
108    }
109 }
```

Titanic unit test

1. Open Form1 class code
2. Add test method at the bottom of the class

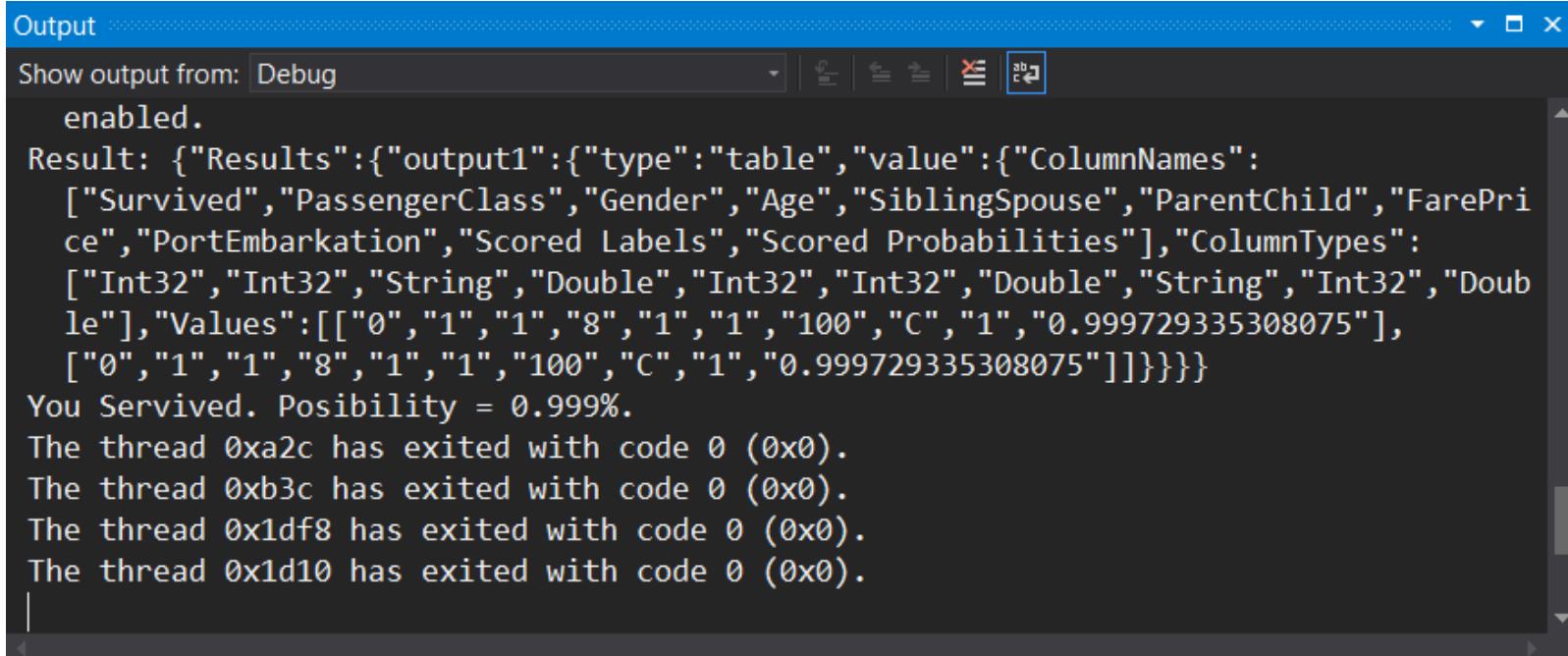
```
131     private async void test()
132     {
133         Input myInput = new Input();
134         myInput.Age = 8;
135         myInput.Cabin = "value";
136         myInput.Embarked = "C";
137         myInput.Fare = 100;
138         myInput.Name = "value";
139         myInput.Parch = 1;
140         myInput.PassengerId = 0;
141         myInput.Pclass = 1;
142         myInput.Sex = "1";
143         myInput.SibSp = 1;
144         myInput.Survived = 0;
145         myInput.Ticket = "value";
146
147         Titanic myTitanic = new Titanic();
148         await myTitanic.GetPrediction(myInput);
149         Console.WriteLine(myTitanic.resultMessage);
150     }
```

Titanic unit test

3. Add this code to Form Load method

```
31  private void Form1_Load(object sender, EventArgs e)
32  {
33      test();
34  }
```

4. Run program and watch Output



The screenshot shows the Visual Studio Output window with the following content:

```
Output
Show output from: Debug
enabled.

Result: {"Results": {"output1": {"type": "table", "value": {"ColumnNames": ["Survived", "PassengerClass", "Gender", "Age", "SiblingSpouse", "ParentChild", "FarePrice", "PortEmbarkation", "Scored Labels", "Scored Probabilities"], "ColumnTypes": ["Int32", "Int32", "String", "Double", "Int32", "Int32", "Double", "String", "Int32", "Double"], "Values": [[[0, 1, 1, 8, 1, 1, "100", "C", 1, 0.999729335308075], [0, 1, 1, 8, 1, 1, "100", "C", 1, 0.999729335308075]]]}}}

You Survived. Possibility = 0.999%.
The thread 0xa2c has exited with code 0 (0x0).
The thread 0xb3c has exited with code 0 (0x0).
The thread 0x1df8 has exited with code 0 (0x0).
The thread 0x1d10 has exited with code 0 (0x0).
```

Adding code to Class Form1

Class overview

```
1  /// <summary> Titanic-Survival-Predictor
11
12  using ...
22
23  namespace test
24  {
25      public partial class Form1 : Form
26      {
27          3 references
28          private Input myInput = new Input();
29          private Titanic myTitanic = new Titanic();
30
31          1 reference
32          public Form1()...
33
34          1 reference
35          private void Form1_Load(object sender, EventArgs e)...
36
37          2 references
38          private void Reset()...
39
40
41          Buttons
42
43          TrackBar
44
45          RadioButtons
46
47
48      }
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141 }
```

Adding code to Class Form1

Constructor & Form load

```
30     □ 1 reference
31     public Form1()
32     {
33         InitializeComponent();
34         trackBarAge.Minimum = 0;
35         trackBarAge.Maximum = 80;
36         trackBarSib.Minimum = 0;
37         trackBarSib.Maximum = 8;
38         trackBarPar.Minimum = 0;
39         trackBarPar.Maximum = 6;
40         trackBarPay.Minimum = 0;
41         trackBarPay.Maximum = 512;
42     □ 1 reference
42     private void Form1_Load(object sender, EventArgs e)
43     {
44         Reset();
45     } 2 references
```

Adding code to Class Form1

Method Reset

```
46  private void Reset()
47  {
48      myInput.Clear();
49
50      labelResult.Text = "Ready";
51      radioButtonFirstClass.Checked = true;
52      radioButtonMale.Checked = true;
53
54      labelAge.Text = "20";
55      trackBarAge.Value = 20;
56
57      labelSib.Text = "0";
58      trackBarSib.Value = 0;
59
60      labelPar.Text = "0";
61      trackBarPar.Value = 0;
62
63      labelPay.Text = "1";
64      trackBarPay.Value = 1;
65
66      radioButtonCher.Checked = true;
67 }
```

Adding code to Class Form1

Method button click

```
69 #region Buttons
70 private void buttonReset_Click(object sender, EventArgs e)
71 {
72     Reset();
73 }
74 private async void buttonSubmit_Click(object sender, EventArgs e)
75 {
76     labelResult.Text = "Processing....";
77     await myTitainc.GetPrediction(myInput);
78     labelResult.BeginInvoke(new Action(() => { labelResult.Text = myTitainc.resultMessage; }));
79 }
80 #endregion
```

Adding code to Class Form1

Method TrackBar

```
82     #region TrackBar
83     1 reference
84     private void trackBarAge_Scroll(object sender, EventArgs e)
85     {
86         labelAge.Text = trackBarAge.Value.ToString();
87         myInput.Age = trackBarAge.Value;
88     }
89     1 reference
90     private void trackBarSib_Scroll(object sender, EventArgs e)
91     {
92         labelSib.Text = trackBarSib.Value.ToString();
93         myInput.SibSp = trackBarSib.Value;
94     }
95     1 reference
96     private void trackBarPay_Scroll(object sender, EventArgs e)
97     {
98         labelPay.Text = trackBarPay.Value.ToString();
99         myInput.Fare = trackBarPay.Value;
100    1 reference
101    private void trackBarPar_Scroll(object sender, EventArgs e)
102    {
103        labelPar.Text = trackBarPar.Value.ToString();
104        myInput.Parch = trackBarPar.Value;
105    }
106    endregion
```

Adding code to Class Form1

Method radio button / passenger class

```
105     #region RadioButtons
106     private void radioButtonFirstClass_CheckedChanged(object sender, EventArgs e)
107     {
108         myInput.Pclass = 1;      //1 = 1st, 2 = 2nd, 3 = 3rd
109     }
110     private void radioButtonSecondClass_CheckedChanged(object sender, EventArgs e)
111     {
112         myInput.Pclass = 2;      //1 = 1st, 2 = 2nd, 3 = 3rd
113     }
114     private void radioButtonThirdClass_CheckedChanged(object sender, EventArgs e)
115     {
116         myInput.Pclass = 3;      //1 = 1st, 2 = 2nd, 3 = 3rd
117     }
118     private void radioButtonMale_CheckedChanged(object sender, EventArgs e)...
119     private void radioButtonFemale_CheckedChanged(object sender, EventArgs e)...
120     private void radioButtonCher_CheckedChanged(object sender, EventArgs e)...
121     private void radioButtonQueens_CheckedChanged(object sender, EventArgs e)...
122     private void radioButtonSout_CheckedChanged(object sender, EventArgs e)...
123
124 #endregion
```

Adding code to Class Form1

Method radio button / Sex & Embark

```
118     private void radioButtonMale_CheckedChanged(object sender, EventArgs e)
119     {
120         myInput.Sex = "0";      // male = 0, female = 1
121     }
1 reference
122     private void radioButtonFemale_CheckedChanged(object sender, EventArgs e)
123     {
124         myInput.Sex = "1";      // male = 0, female = 1
125     }
1 reference
126     private void radioButtonCher_CheckedChanged(object sender, EventArgs e)
127     {
128         myInput.Embarked = "C";      // C = Cherbourg, Q = Queenstown, S = Southampton
129     }
1 reference
130     private void radioButtonQueens_CheckedChanged(object sender, EventArgs e)
131     {
132         myInput.Embarked = "Q";      // C = Cherbourg, Q = Queenstown, S = Southampton
133     }
1 reference
134     private void radioButtonSout_CheckedChanged(object sender, EventArgs e)
135     {
136         myInput.Embarked = "S";      // C = Cherbourg, Q = Queenstown, S = Southampton
137     }
138 #endregion
```

More Information

Source code (MSVS 2017 Solution)

<https://github.com/laploy/Titanic-Survival-Predictor>

More information on Microsoft Azure ML Web Service

Azure Machine Learning Web Services: Deployment and consumption

<https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-deploy-consume-web-service-guide>

CREATING AZURE STORAGE ACCOUNT



In this session

- What is Azure Storage?
- Azure Storage Options
- Creating Azure Resource Group
- Create a storage account
- Microsoft Azure Storage Explorer
- Download and Install ASE

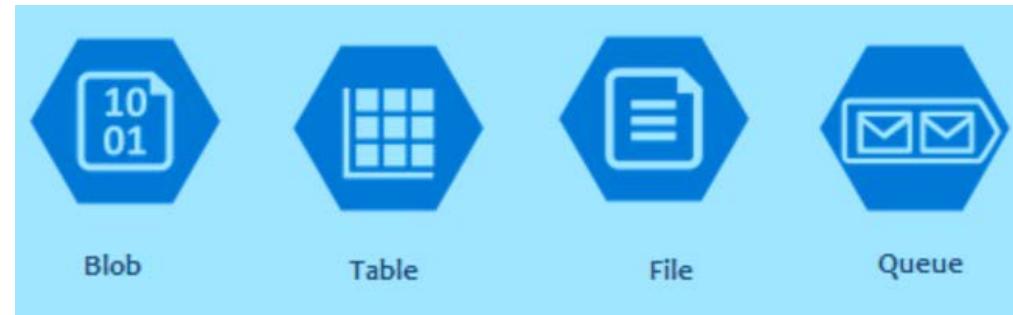
What is Azure Storage?

What is Azure Storage?

- Cloud storage
- Scalable
- Support small data
- Support Big data
- Support Hundreds of terabytes
- Durable and highly available
- Support small application
- Support large-scale applications
- Support Azure Virtual Machines
- Support IoT

In this course we'll use AS to store ML batch process output

Azure Storage Options

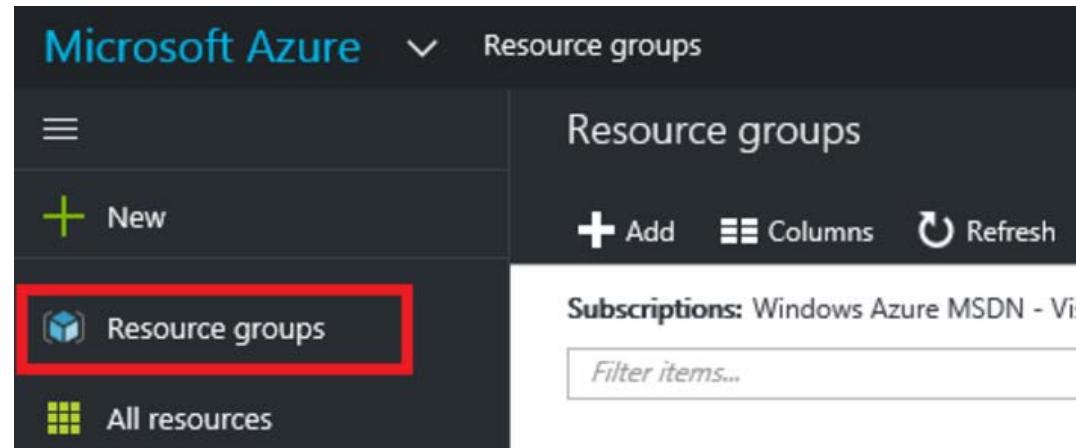


Azure Storage Options

- Blob: unstructured object data
- Table: structured datasets NoSQL
- Queue: messaging for workflow processing
- File: shared storage for legacy applications

In this course we'll use Blob only

Creating Azure Resource Group



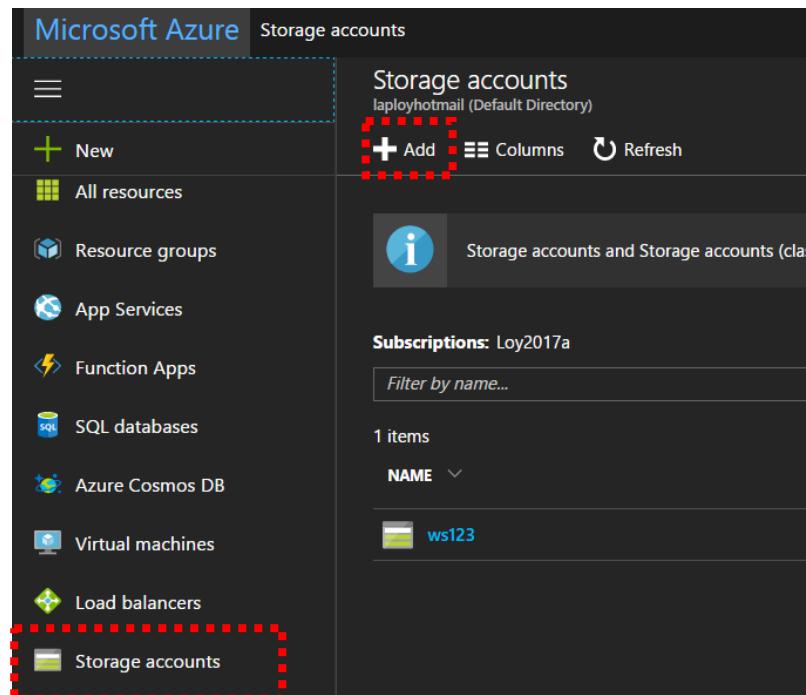
Creating Azure Resource Group

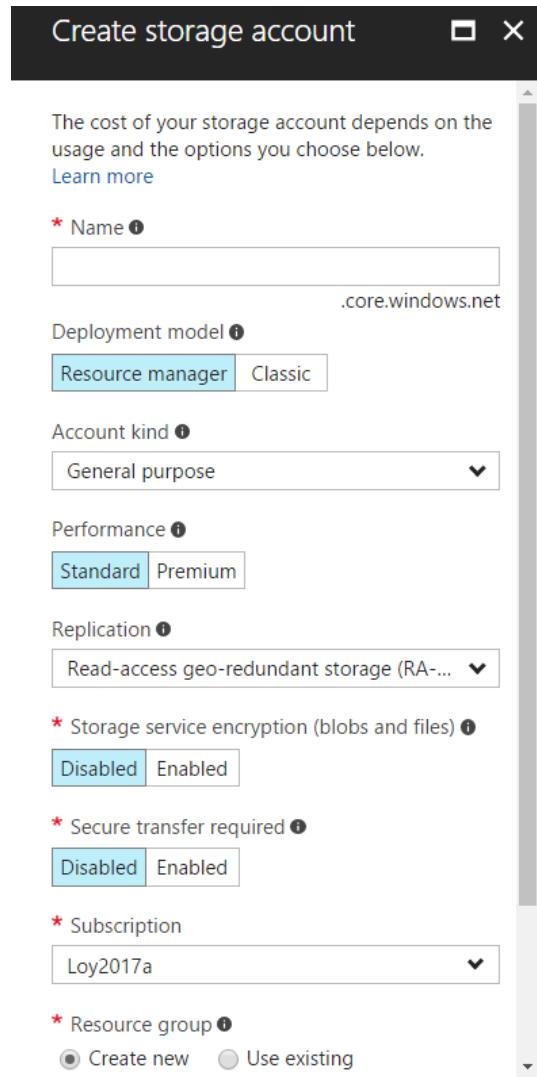
- Go to Azure portal
- Click Resource groups
- Click + Add
- Enter name = rg1
- Subscription = your subscription name
- Location = South East Asia
- Click Create

Create a storage account

Create a storage account

1. Go to Azure portal.
2. Click Storage account
3. Click + Add

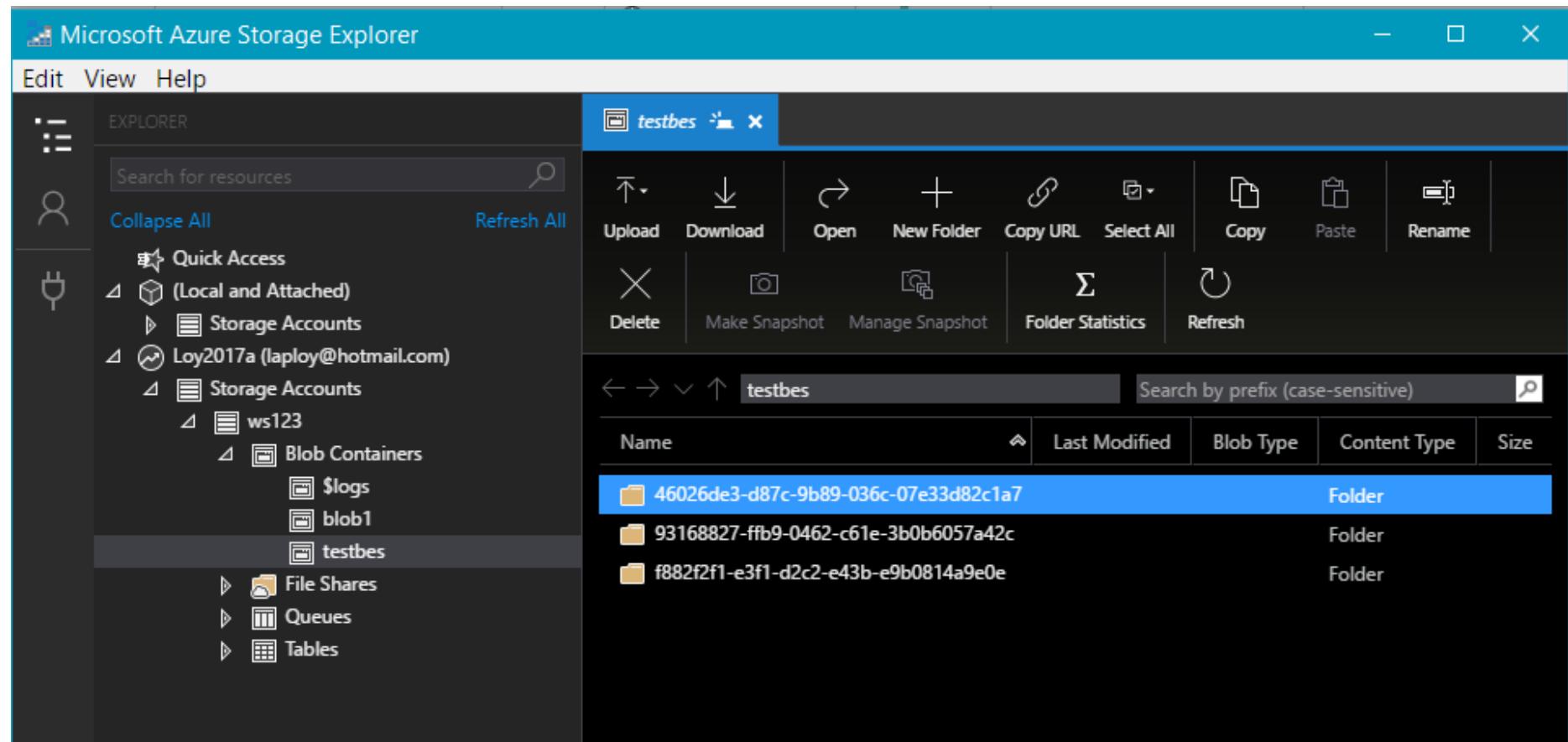




4. Enter a name = sa1
5. Deployment model = Resource Manager
6. Account kind = General purpose
7. Performance = Standard
8. Replication = Default
9. Encryption = Disabled
10. Secure = Disabled
11. Subscription = your sub name
12. Resource group = Use existing
13. Geographic = South East Asia
14. Click Create

Microsoft Azure Storage Explorer

Azure Storage Explorer (ASE) is a standalone app that enables you to easily work with Azure Storage data on Windows, macOS, and Linux.



Download and Install ASE



Download and Install ASE

1. Go to ASE home page storageexplorer.com
2. Click Free download for Windows
3. Download and Install

More information

More information on Microsoft Azure Storage

Introduction to Microsoft Azure Storage (eBook)

<https://opbuildstorageprod.blob.core.windows.net/output-pdf-files/en-us/Azure.azure-documents/live/storage.pdf>

C# AND BATCH EXECUTION API



In this session

- Check list before we continue
- Test titanic Batch Execution API
- C# Batch API development steps
- Find Base Address
- Find Storage Account Name
- Find Storage Account Key
- Find Storage container Name
- Find Storage container Name
- Find Storage container Name
- Find Web Service API Key
- Create C# Batch Execution API

Check list before we continue

Check list before we continue

1. Create Microsoft Azure Account
2. Create Microsoft Azure Resource Group
3. Create Microsoft Azure Storage Account
4. Create Microsoft Azure ML Account
5. Create Titanic ML experiment
6. Deploy Web Service Titanic
7. Test Web Service REQUEST/RESPONSE using Azure ML WS dashboard
8. Test Web Service REQUEST/RESPONSE using Excel
9. Test Web Service BATCH EXECUTION using Excel

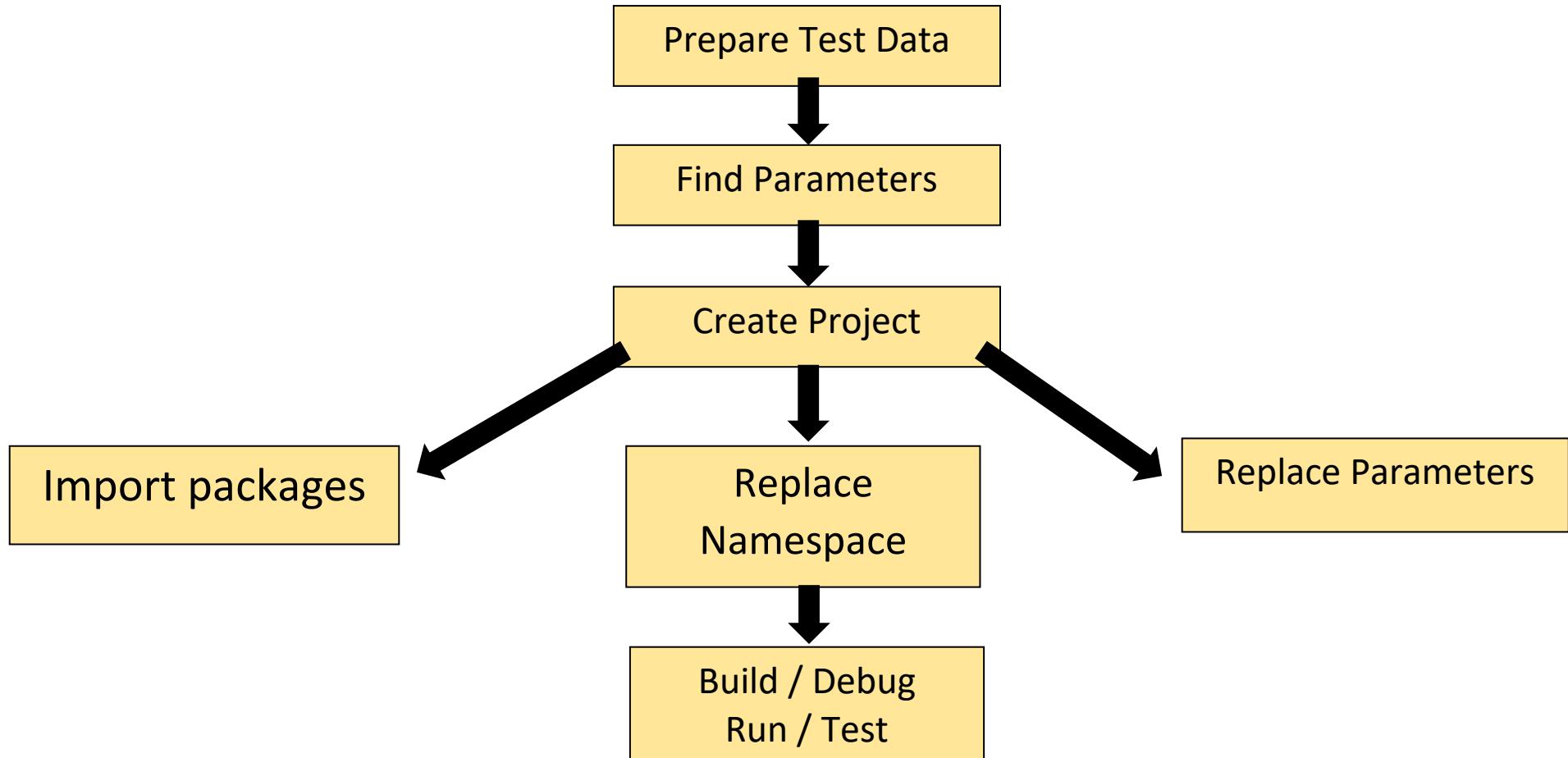
Test titanic Batch Execution API

Test titanic Batch Execution API

1. Go to github.com/laploy/ML
2. Download file data.csv
3. Place file data.csv in to folder c:\temp
4. Download file test3.zip
5. Unzip
6. Run program test3

Download data.csv and test3.zip from github.com/laploy/ML

C# Batch App development steps



Find Base Address

Base Address

1. Go to Azure ML home page
2. Click Web services (left side-bar)
3. Click Titanic
4. Click API HELP PAGE / BATCH EXECUTION
5. Find topic submit (but not start) a Batch Execution job (remove query)

Submit (but not start) a Batch Execution job

Request

	Method	Request URI
POST		<code>https://ussouthcentral.services.azureml.net/workspaces bs?api-version=2.0</code>

Find Storage Account Name

Storage Account Name

1. Go to Azure portal (portal.azure.com)
2. Click Storage
3. Storage Account = Storage Account Name

The screenshot shows the Microsoft Azure Storage accounts page. At the top, it displays "Microsoft Azure" and "Storage accounts". On the left, there's a sidebar with icons for Storage accounts, Blob storage, Queue storage, Table storage, and SQL databases. The main area shows a table with one item:

NAME	TYPE	KIND
Storage account	Storage account	Storage

A red dashed box highlights the entire row. A message in the center of the page says: "Storage accounts and Storage accounts (classic) can now be managed together in the combined list below."

Find Storage Account Key

Storage Account Key

1. Go to Azure portal (portal.azure.com)
2. Click Storage
3. Click Storage Account / Account
4. Click Settings / Access keys / Copy

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the 'Microsoft Azure' logo, the account name 'ws123 - Access keys', and a search bar labeled 'Search resources'. On the left, there's a sidebar with various icons and links: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, SETTINGS (highlighted with a red dashed box), Access keys (selected and highlighted with a red dashed box), and Configuration. The main content area is titled 'ws123 - Access keys' and 'Storage account'. It contains instructions about using access keys for authentication and regenerating them. Below this, it shows the 'Storage account name' (partially obscured) and 'Default keys'. There are two entries in the 'Default keys' table:

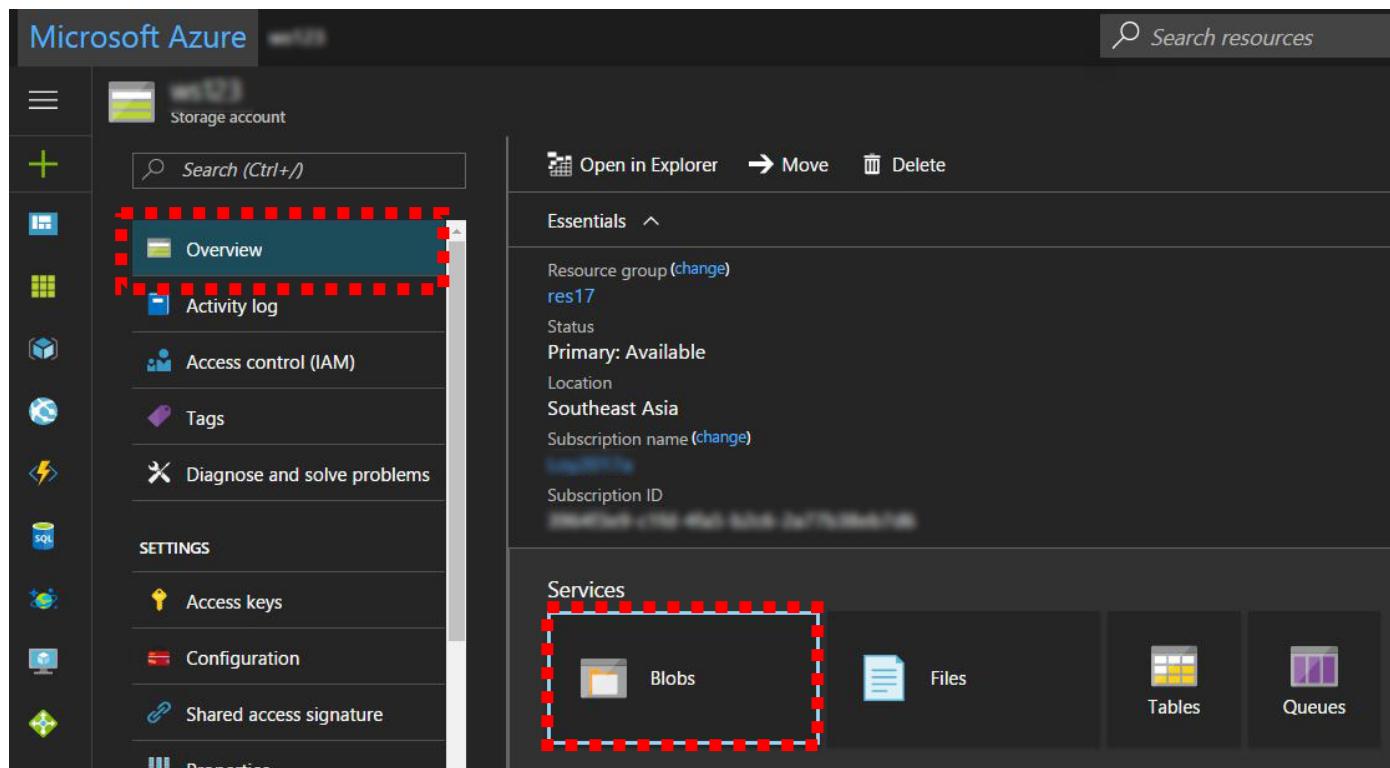
NAME	KEY
key1	[REDACTED]
key2	[REDACTED]

Each key entry has a small icon for copying the value.

Find Storage container Name

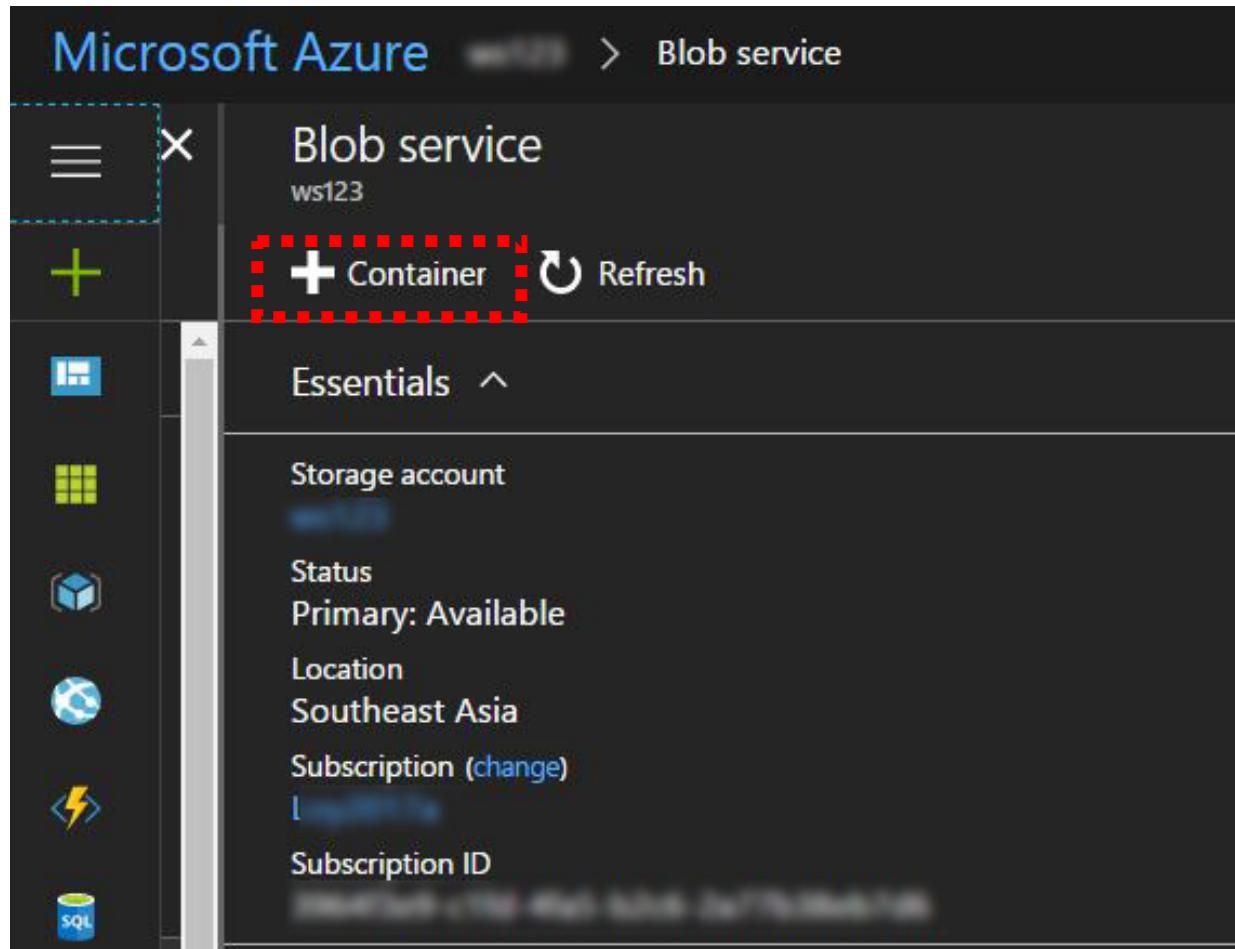
Storage container name

1. Go to Azure portal (portal.azure.com)
2. Click Storage / Over view
3. Click Services / Blobs



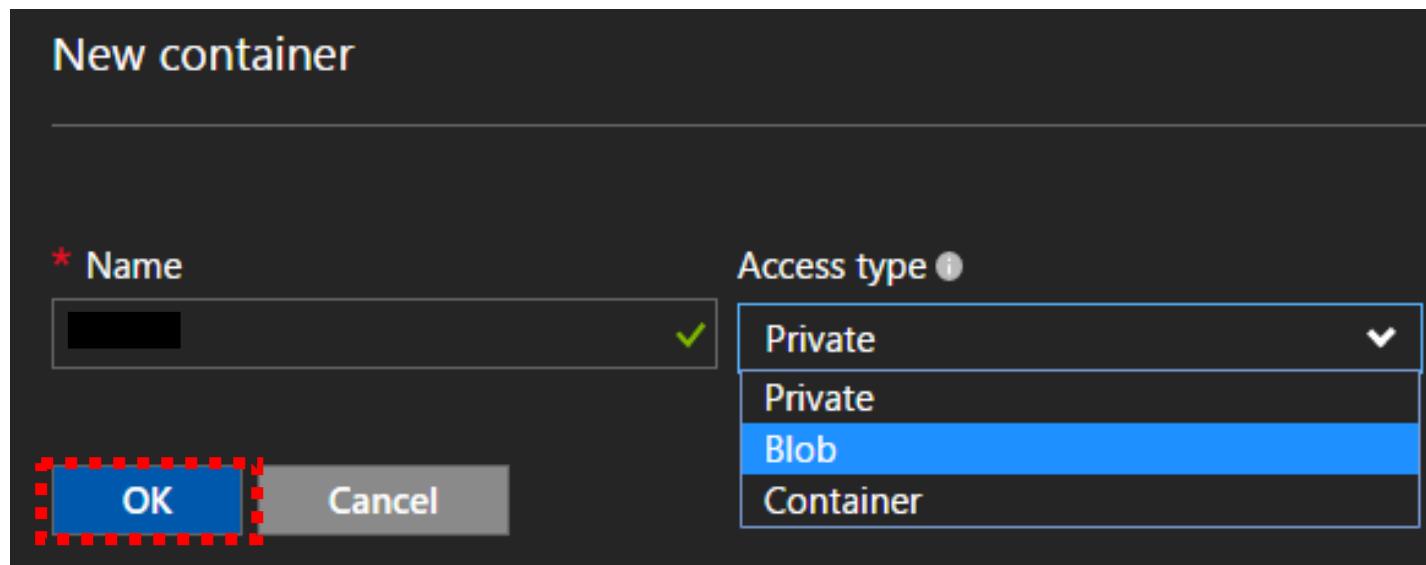
Find Storage container Name

4. Click + Container



Find Storage container Name

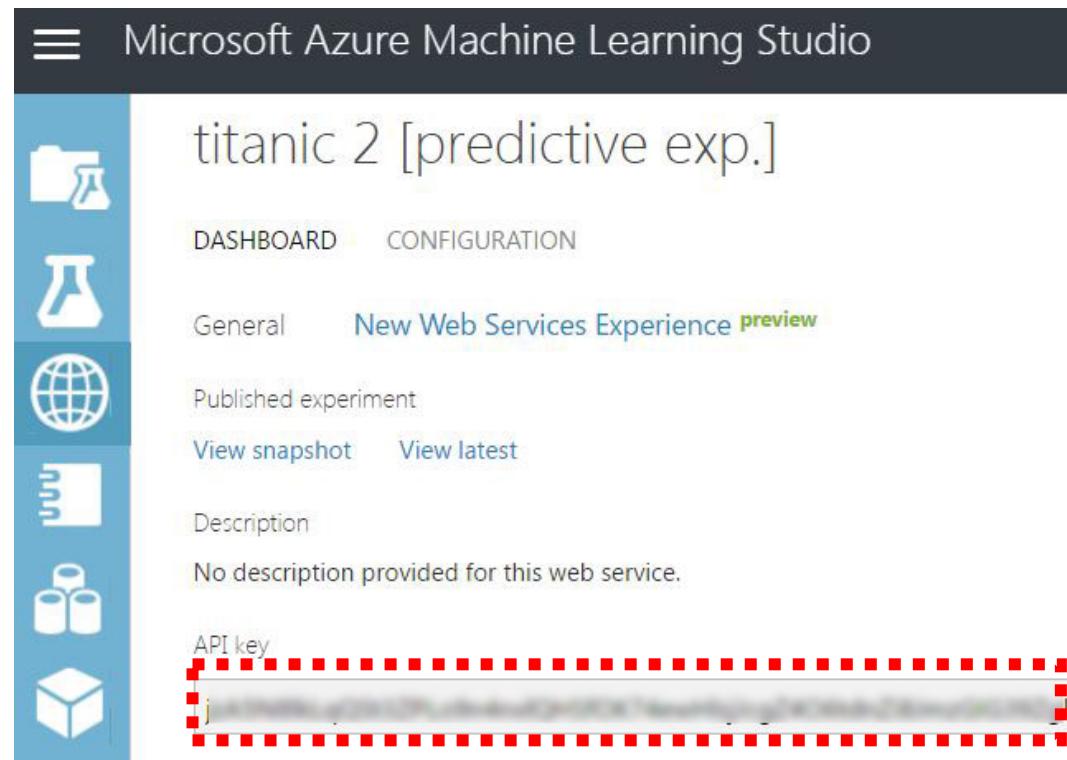
5. Enter name = blob1
6. Access type = Blob
7. Click OK



Find Web Service API Key

Find Web Service API Key

1. Go to Azure ML home page
2. Click Web services (left side-bar)
3. Click Titanic
4. Copy API key



Create C# Batch Execution API

- Go to your Azure ML home page
- Click Web services (at the left side-bar)
- Click Titanic Web Service
- Click API HELP PAGE / BATCH EXECUTION
- Select / Copy Sample C# code
- Open Visual Studio 2017
- Create New Project
 - Visual C#
 - Windows Classic Desktop
 - Console App (.NET Framework)
 - Name = TitanicBE

Create C# Batch Execution API

- Past code in to Main
- Change name space to TitanicBE
- Add nugget
 - Microsoft.AspNet.WebApi.Client
 - Microsoft.WindowsAzure.Storage.dll
- Replace value of
 - const string StorageAccountName
 - const string StorageAccountKey
 - const string StorageContainerName
 - const string apiKey

Create C# Batch Execution API

- Change input1data.csv to c:\temp\data.csv
- Change const string OutputFileLocation to c:\temp\myResult.csv
- Change input1datablob.csv to Intitanic.csv
- Build program
- Debug
- Run program
- Check the API job result at c:\temp\myResult.csv

You can download source code here

<https://github.com/laploy/bs>

More information

How to consume an Azure Machine Learning Web service

<https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-consume-web-services>

MISSING VALUE HANDLING

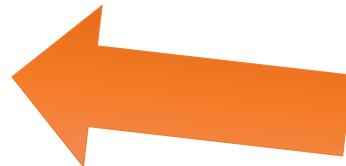


In this session

1. Replace missing values with the mean
2. Replace missing values with the median
3. Replace missing values with an interpolated estimate
4. Replace missing values with a constant
5. Replace missing values using imputation
6. Replace missing values with a missing rank
7. Replace missing values with a dummy
8. Replace missing values with 0
9. Create an indicator variable for "missing."
10. Replace missing values with a string
11. Add an indicator variable showing which strings are considered "missing."
12. Delete columns that are missing too many values to be useful
13. Delete rows that are missing critical values

We need data that is:

- Relevant
- Connected
- Accurate
- Enough to work with



Example of missing values dataset

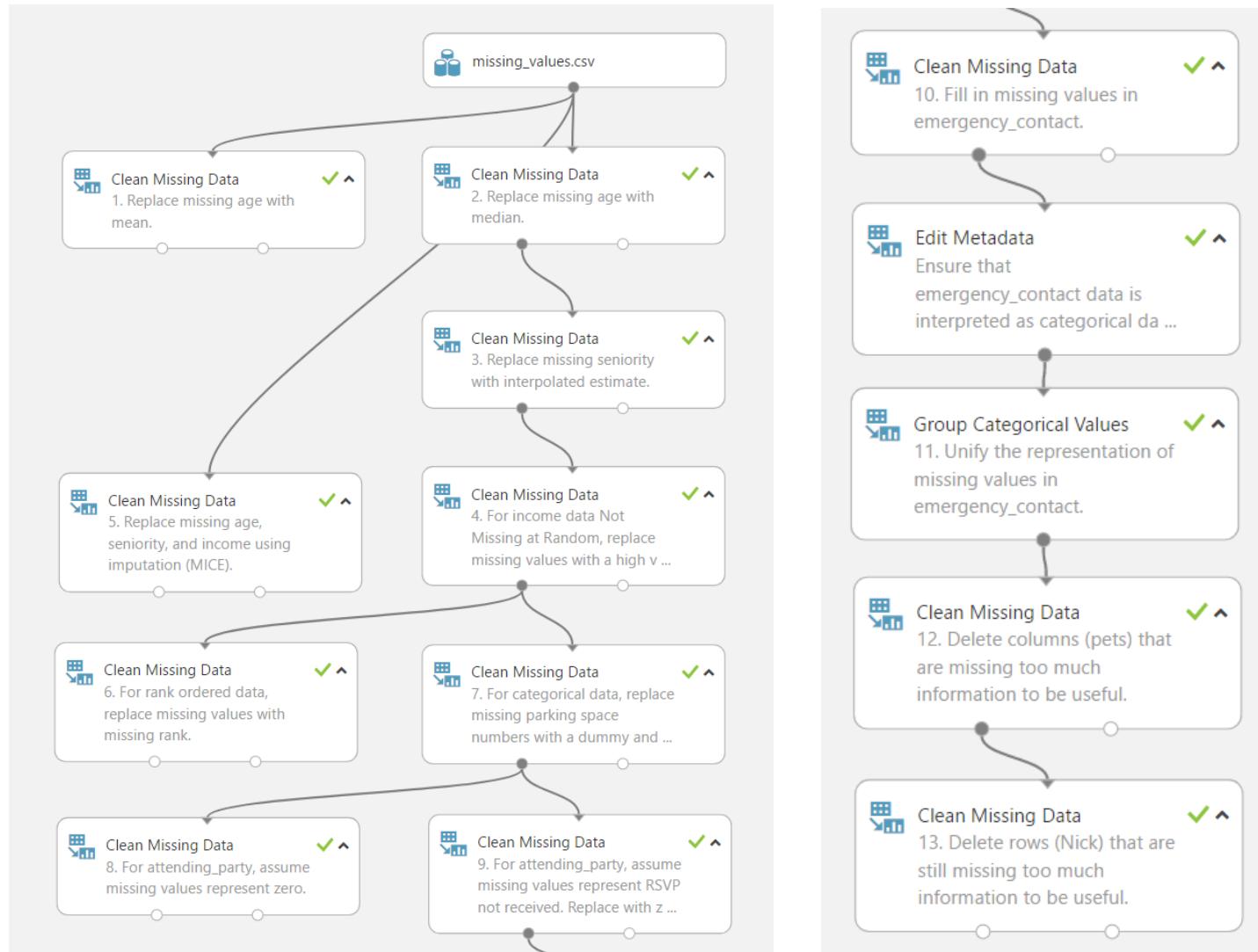
Column 0	age	years_seniority	income	parking_space	attending_party	entree	pets	emergency_contact
Tony	48	27		1	5	shrimp		Pepper
Donald	67	25	86	10	2	beef		Jane
Henry	69	21	95	6	1	chicken	62	Janet
Janet	62	21	110	3	1	beef		Henry
Nick		17		4				NA
Bruce	37	14	63		1	veggie		n/a
Steve	83		77	7	1	chicken		None
Clint	27	9	118	9		shrimp	3	empty
Wanda	19	7	52	2	2	shrimp		-
Natasha	26	4	162	5	3			*****
Carol		3	127	11	1	veggie	1	null
Mandy	44	2	68	8	1	chicken		

Example of missing values dataset CSV file

missing_values.csv

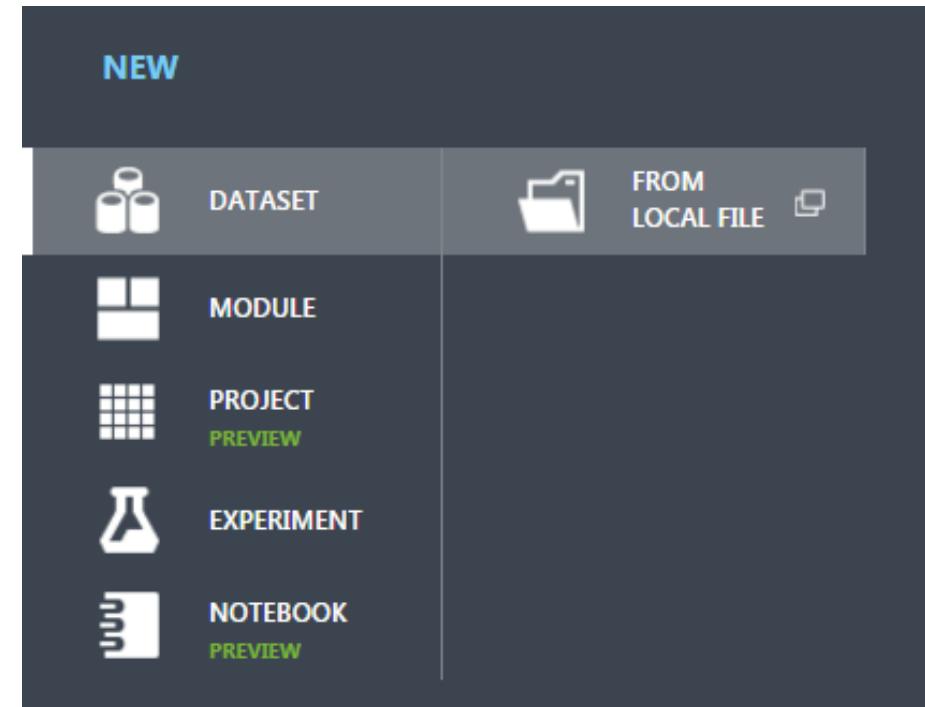
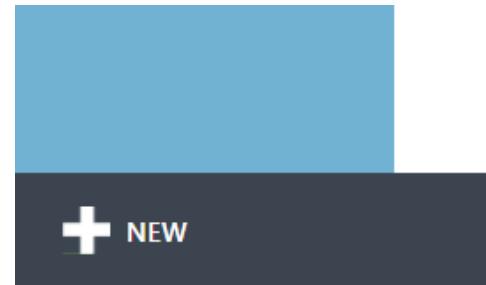
	A	B	C	D	E	F	G	H	I
1		age	years_seniority	income	parking_space	attending_party	entree	pets	emergency_contact
2	Tony	48	27		1		5 shrimp		Pepper
3	Donald	67	25	86	10		2 beef		Jane
4	Henry	69	21	95	6		1 chicken	62	Janet
5	Janet	62	21	110	3		1 beef		Henry
6	Nick		17		4				
7	Bruce	37	14	63			1 veggie		NA
8	Steve	83		77	7		1 chicken		n/a
9	Clint	27	9	118	9		shrimp	3	None
10	Wanda	19	7	52	2		2 shrimp		empty
11	Natasha	26	4	162	5		3		-
12	Carol		3	127	11		1 veggie	1	""
13	Mandy	44	2	68	8		1 chicken		null

Experiment: Methods for handling missing values



Import local data file to Azure ML Studio Dataset

1. Select Datasets tab from menu
2. Click New (+) at the button left corner
3. Click “From local file”



4. Choose file and type description

Upload a new dataset x

SELECT THE DATA TO UPLOAD:

missing_values.csv

This is the new version of an existing dataset

ENTER A NAME FOR THE NEW DATASET:

missing_values.csv

SELECT A TYPE FOR THE NEW DATASET:

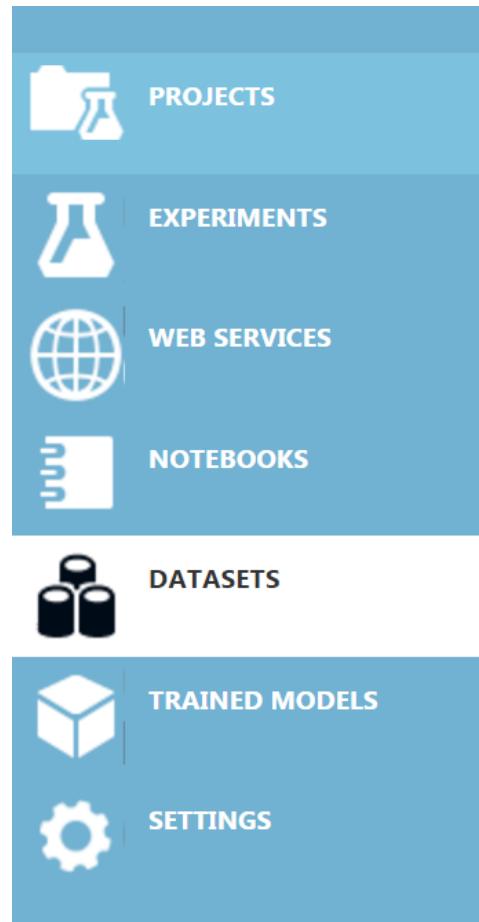
Generic CSV File with a header (.csv) ▾

PROVIDE AN OPTIONAL DESCRIPTION:

sample of missing data

✓

Saved datasets list

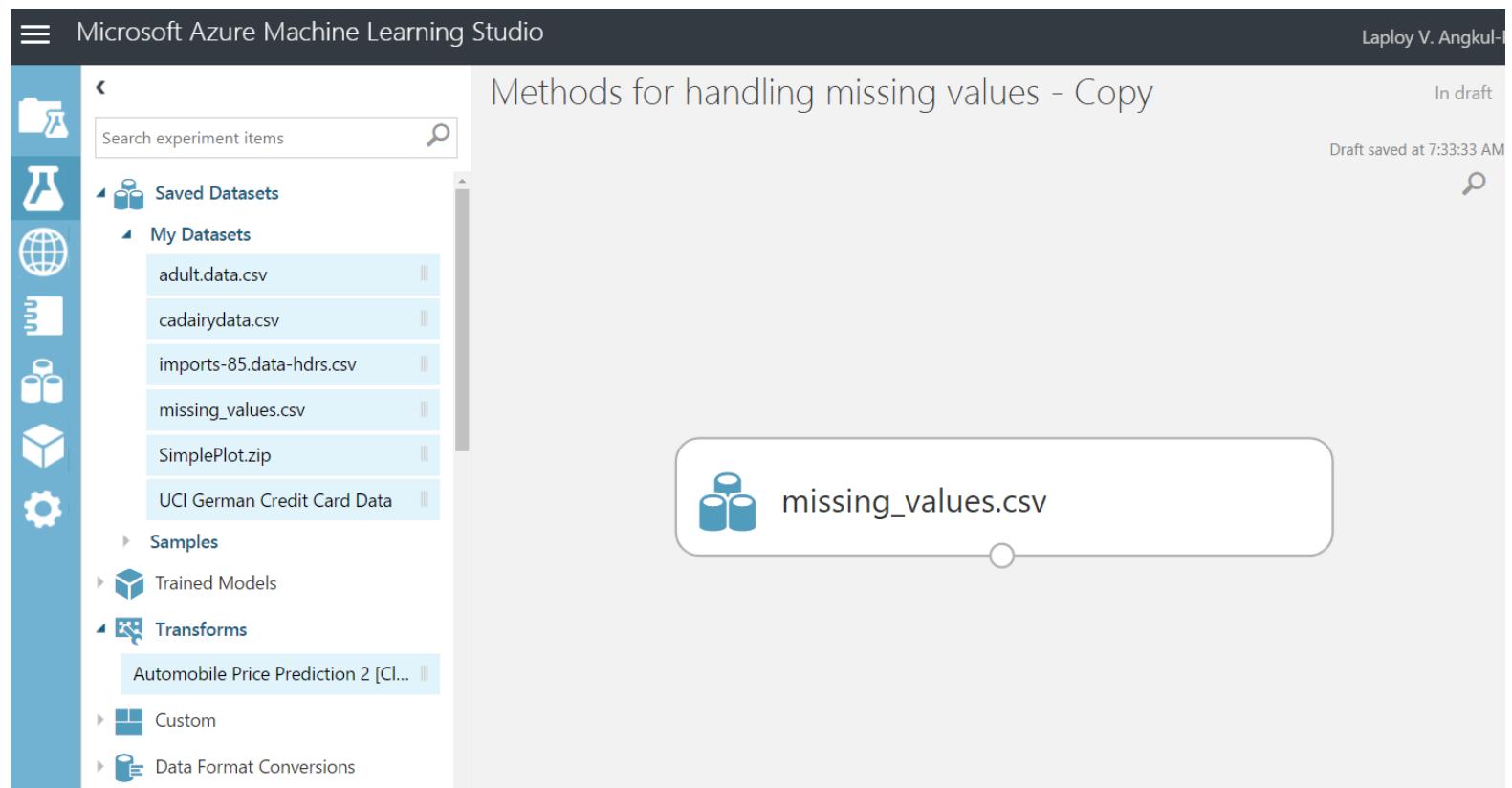


datasets

MY DATASETS SAMPLES

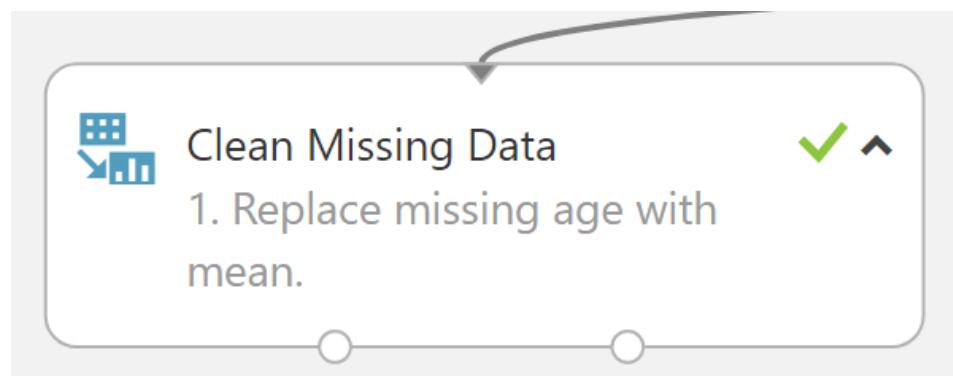
NAME	SUBMITTED BY	DESCRIPTION
missing_values.csv	Brandon Rohrer	A set of example cases of missing values tha...
imports-85.data-hdrs.csv	laploy	1985 Auto Imports Database
adult.data.csv	laploy	Adult Census Income Binary Classification d...
SimplePlot.zip	laploy	simple first R script to experiment with in Az..
cadairydata.csv	laploy	California dairy production and pricing data...
UCI German Credit Card Da...	laploy	

- Create new blank Experiment
- Select missing_values.csv from Saved Datasets
- Drag & drop into canvas



Replace missing values with the mean

- Change project name to “Methods for handling missing values”
- Drag & drop Clean Missing Data module
- Select column age
- Configure “Cleaning mode” to Replace with mean
- Comment = 1. Replace missing age with mean.
- Run/Visualize



Properties Project

▲ Clean Missing Data

Columns to be cleaned

Selected columns:
Column names: age

Launch column selector

Minimum missing value ...

0

Maximum missing value...

1

Cleaning mode

Replace with mean ▾

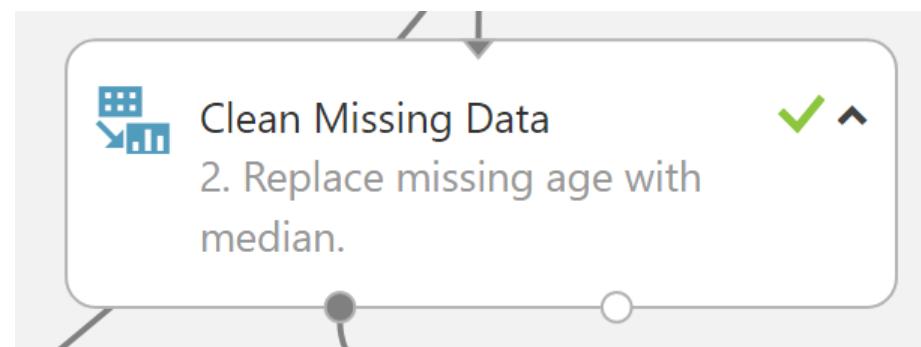
Cols with all missing val...

Remove ▾

Generate missing va...

Replace missing values with the median

- Drag & drop Clean Missing Data module
- Select column age
- Configure “Cleaning mode” to Replace with median
- Comment = 2. Replace missing age with median.
- Run/Visualize



Properties Project

Clean Missing Data

Columns to be cleaned

Selected columns:

Column names: age

Launch column selector

Minimum missing value ...

0

Maximum missing value...

1

Cleaning mode

Replace with median ▾

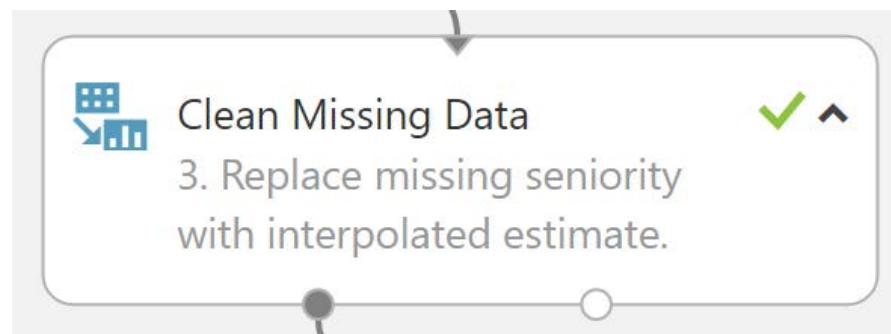
Cols with all missing val...

Remove ▾

Generate missing va...

Replace missing values with an interpolated estimate

- Drag & drop Clean Missing Data module
- Select column year_seniority
- Configure “Cleaning mode” to Custom substitution value
- Set Replacement value to 11.5
- Comment = 3. Replace missing seniority with interpolated estimate.
- Run/Visualize



Properties Project

Clean Missing Data

Columns to be cleaned

Selected columns:

Column names: years_seniority

Launch column selector

Minimum missing value ratio

0

Maximum missing value ratio

1

Cleaning mode

Custom substitution value

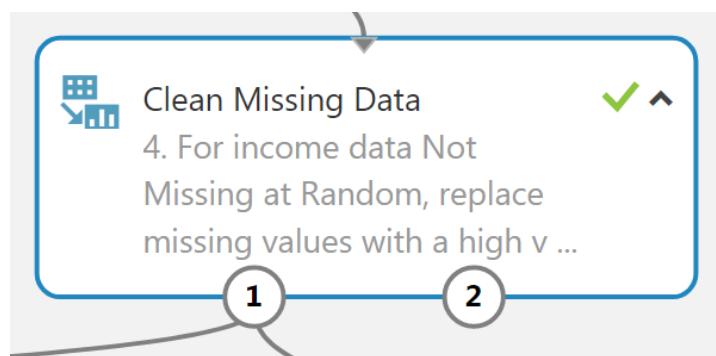
Replacement value

11.5

Generate missing value indica..

Replace missing values with a constant

- Drag & drop Clean Missing Data module
- Select column income
- Configure “Cleaning mode” to Custom substitution value
- Set Replacement value to 250
- Comment = 4. For income data Not Missing at Random, replace missing values with a high value.
- Run/Visualize



Properties Project

Clean Missing Data

Columns to be cleaned

Selected columns:

Column names: income

Launch column selector

Minimum missing value ratio

0

Maximum missing value ratio

1

Cleaning mode

Custom substitution value

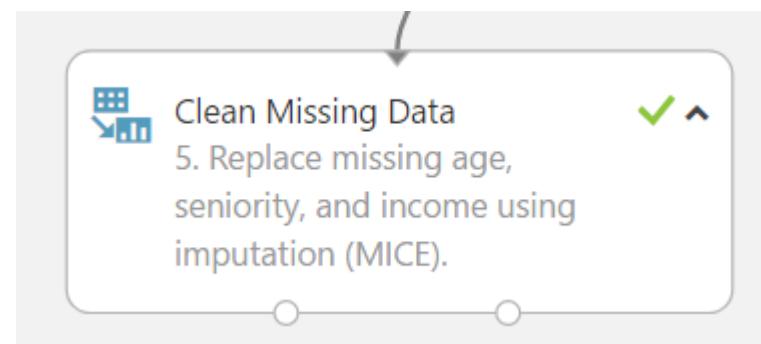
Replacement value

250

Generate missing value indica..

Replace missing values using imputation

- Drag & drop Clean Missing Data module
- Select column years_seniority.age.income
- Configure “Cleaning mode” to Replace using MICE
- Cols with all missing values = Remove
- Number of iterations = 5
- Comment = 5. Replace missing age, seniority, and income using imputation (MICE).
- Run/Visualize



Properties Project

◀ Clean Missing Data

Columns to be cleaned

Selected columns:
Column names:
years_seniority,age,income

Launch column selector

Minimum missing value ratio
0

Maximum missing value ratio
1

Cleaning mode
Replace using MICE

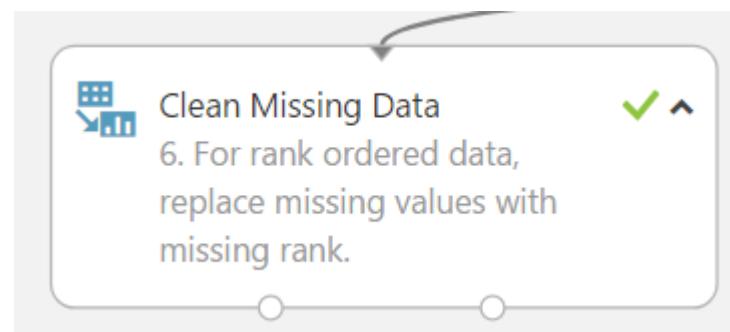
Cols with all missing values
Remove

Generate missing value indic...

Number of iterations
5

Replace missing values with a missing rank

- Drag & drop Clean Missing Data module
- Select column parking_space
- Configure “Cleaning mode” to Custom substitution value
- Replacement value = 12
- Comment = 6. For rank ordered data, replace missing values with missing rank.
- Run/Visualize



Properties Project

Clean Missing Data

Columns to be cleaned

Selected columns:

Column names: parking_space

Launch column selector

Minimum missing value ratio

0

Maximum missing value ratio

1

Cleaning mode

Custom substitution value

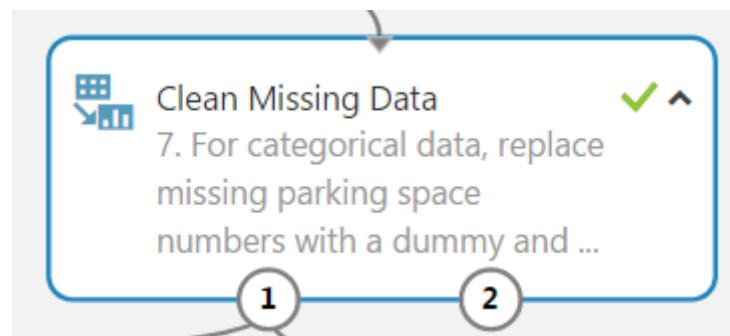
Replacement value

12

Generate missing value indicato...

Replace missing values with a dummy

- Drag & drop Clean Missing Data module
- Select column parking_space
- Configure “Cleaning mode” to Custom substitution value
- Replacement value = -99
- Comment = 7. For categorical data, replace missing parking space numbers with a dummy and include a missing values indicator column
- Run/Visualize



Properties Project

Clean Missing Data

Columns to be cleaned

Selected columns:

Column names: parking_space

Launch column selector

Minimum missing value ratio

0

Maximum missing value ratio

1

Cleaning mode

Custom substitution value

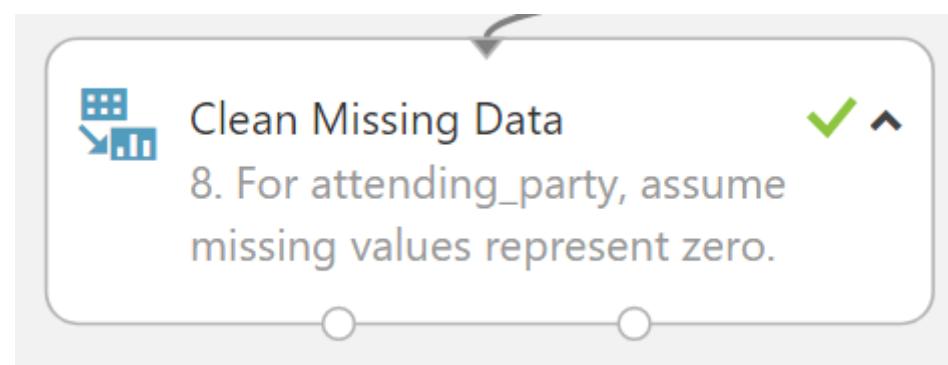
Replacement value

-99

Generate missing value indicato...

Replace missing values with 0

- Drag & drop Clean Missing Data module
- Select column attending_party
- Configure “Cleaning mode” to Custom substitution value
- Replacement value = 0
- Comment = 8. For attending_party, assume missing values represent zero.
- Run/Visualize



Properties Project

▲ Clean Missing Data

Columns to be cleaned

Selected columns:
Column names: attending_party

Launch column selector

Minimum missing value ratio

0

Maximum missing value ratio

1

Cleaning mode

Custom substitution value

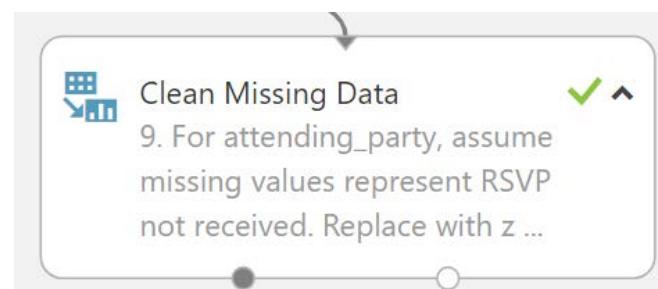
Replacement value

0

Generate missing value indicato...

Create an indicator variable for "missing."

- Drag & drop Clean Missing Data module
- Select column attending_party
- Configure “Cleaning mode” to Custom substitution value
- Check Generate missing value indication column
- Replacement value = 0
- Comment = 9. For attending_party, assume missing values represent RSVP not received. Replace with zero, but add a missing value indicator column.
- Run/Visualize



Properties Project

▲ Clean Missing Data

Columns to be cleaned

Selected columns:
Column names: attending_party

Launch column selector

Minimum missing value ratio

0

Maximum missing value ratio

1

Cleaning mode

Custom substitution value

Replacement value

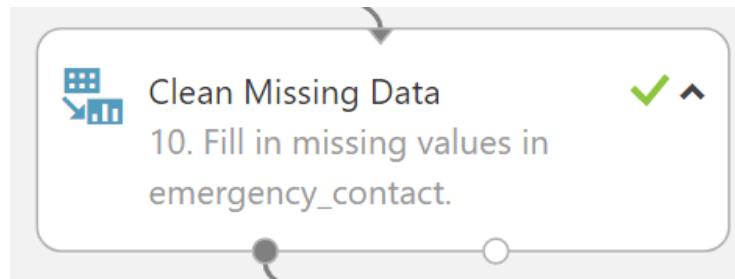
0

Generate missing value indicator column

This screenshot shows the 'Clean Missing Data' properties pane. It includes sections for 'Selected columns', 'Minimum missing value ratio' (set to 0), 'Maximum missing value ratio' (set to 1), 'Cleaning mode' (set to 'Custom substitution value'), 'Replacement value' (set to 0), and a checked checkbox for 'Generate missing value indicator column'.

Replace missing values with a string

- Drag & drop Clean Missing Data module
- Select column emergency_contact
- Configure “Cleaning mode” to Custom substitution value
- Replacement value = no
- Comment = 10. Fill in missing values in emergency_contact.
- Run/Visualize



Properties Project

Clean Missing Data

Columns to be cleaned

Selected columns:

Column names: emergency_contact

Launch column selector

Minimum missing value ratio

0

Maximum missing value ratio

1

Cleaning mode

Custom substitution value

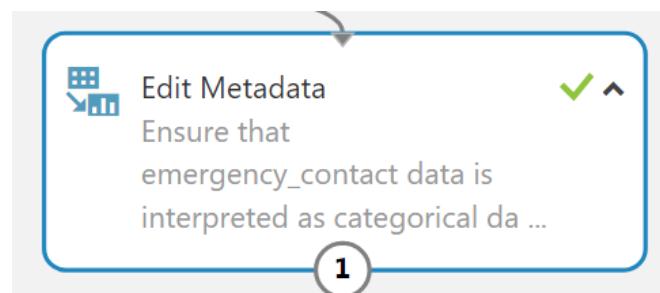
Replacement value

no

Generate missing value indicator column

Change metadata to categorical data

- Drag & drop Edit Metadata
- Select column emergency_contact
- Configure “Cleaning mode” to Custom substitution value
- Data type = String
- Categorical = Make categorical
- Comment = Ensure that emergency_contact data is interpreted as categorical data.
- Run/Visualize



Properties Project

◀ Edit Metadata

Column

Selected columns:
Column names: emergency_contact

Launch column selector

Data type

String

Categorical

Make categorical

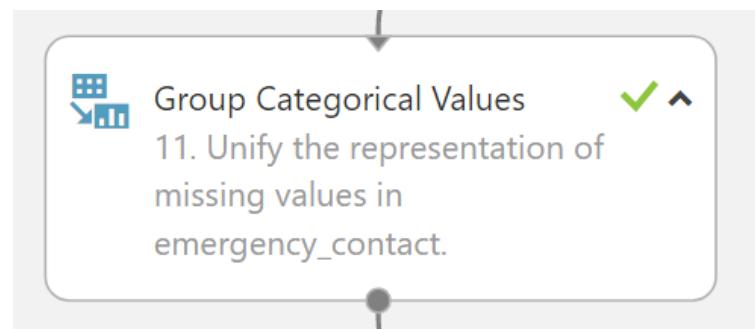
Fields

Unchanged

New column names

Add an indicator variable showing which strings are considered "missing."

- Drag & drop Group Categorical Values
- Select column emergency_contact
- Configure “Cleaning mode” to Custom substitution value
- Output mode = Append
- Default level name = present
- New number of levels = 2
- Name of new level 1 = absent
- Comma-separate list of level to map to new level 1 = no,NA,n/a,None,_,"""",empty,null
- Comment = 11. Unify the representation of missing values in emergency_contact.
- Run/Visualize



Properties Project

◀ Group Categorical Values

Selected columns

Selected columns:
Column names: emergency_contact

Launch column selector

Output mode

Append

Default level name

present

New number of levels

2

Name of new level 1

absent

Comma-separated list of old levels to map..

no,NA,n/a,None,_,"""",empty,null

Delete columns that are missing too many values to be useful

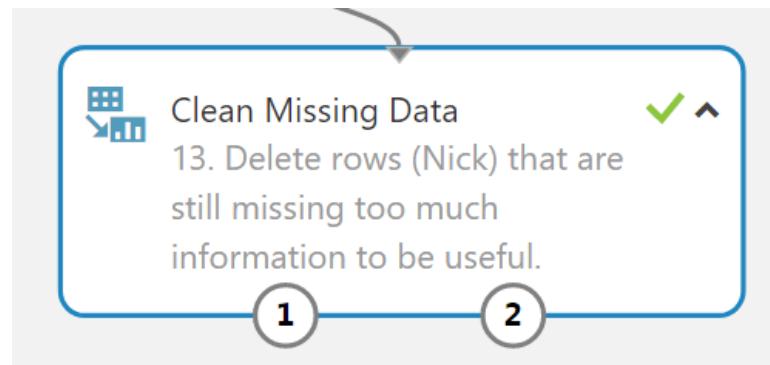
- Drag & drop Clean Missing Data
- Select column pets
- Configure “Cleaning mode” to Remove entire column
- Comment = 12. Delete columns (pets) that are missing too much information to be useful.
- Run/Visualize

The screenshot shows the configuration interface for the 'Clean Missing Data' component. At the top, there are tabs for 'Properties' and 'Project'. Below the tabs, the component name 'Clean Missing Data' is displayed with a comment: '12. Delete columns (pets) that are missing too much information to be useful.' To the right of the component, there is a 'Launch column selector' button. The configuration area includes the following settings:

- Selected columns:** Column names: pets
- Minimum missing value ratio:** .5
- Maximum missing value ratio:** 1
- Cleaning mode:** Remove entire column

Delete rows that are missing critical values

- Drag & drop Clean Missing Data
- Select column entree.emergency_contact
- Configure “Cleaning mode” to Remove entire row
- Comment = 13. Delete rows (Nick) that are still missing too much information to be useful.
- Run/Visualize



Properties Project

▲ Clean Missing Data

Columns to be cleaned

Selected columns:
Column names:
entree,emergency_contact

Launch column selector

Minimum missing value ratio

0

Maximum missing value ratio

1

Cleaning mode

Remove entire row

Final result

Column 0	age	years_seniority	income	parking_space	attending_party	entree	emergency_contact	parking_space_IsMissing
	48	27	250	1	5	shrimp	Pepper	false
	67	25	86	10	2	beef	Jane	false
	69	21	95	6	1	chicken	Janet	false
	62	21	110	3	1	beef	Henry	false
	37	14	63	-99	1	veggie	NA	true
	83	12	77	7	1	chicken	n/a	false
	27	9	118	9	0	shrimp	None	false
	19	7	52	2	2	shrimp	empty	false
	46	3	127	11	1	veggie	""	false
	44	2	68	8	1	chicken	null	false

More information

Clean Missing Data: Specifies how to handle the values missing from a dataset

<https://msdn.microsoft.com/library/azure/d2c5ca2f-7323-41a3-9b7e-da917c99f0c4>

This Experiment

<https://gallery.cortanaintelligence.com/Experiment/Missing-values>

PYTHON INTRODUCTION



In this session

- What is Python?
- What is Anaconda?
- Popularity rank
- Why use Anaconda in Machine Learning?
- Anaconda installation
- PyCharm installation
- Hello world
- Basic calculation
- Variable assignment
- Basic Operator
- Data Structure (Tuple, List, Matrix, Dictionaries)
- If Statement
- For Loop
- Function

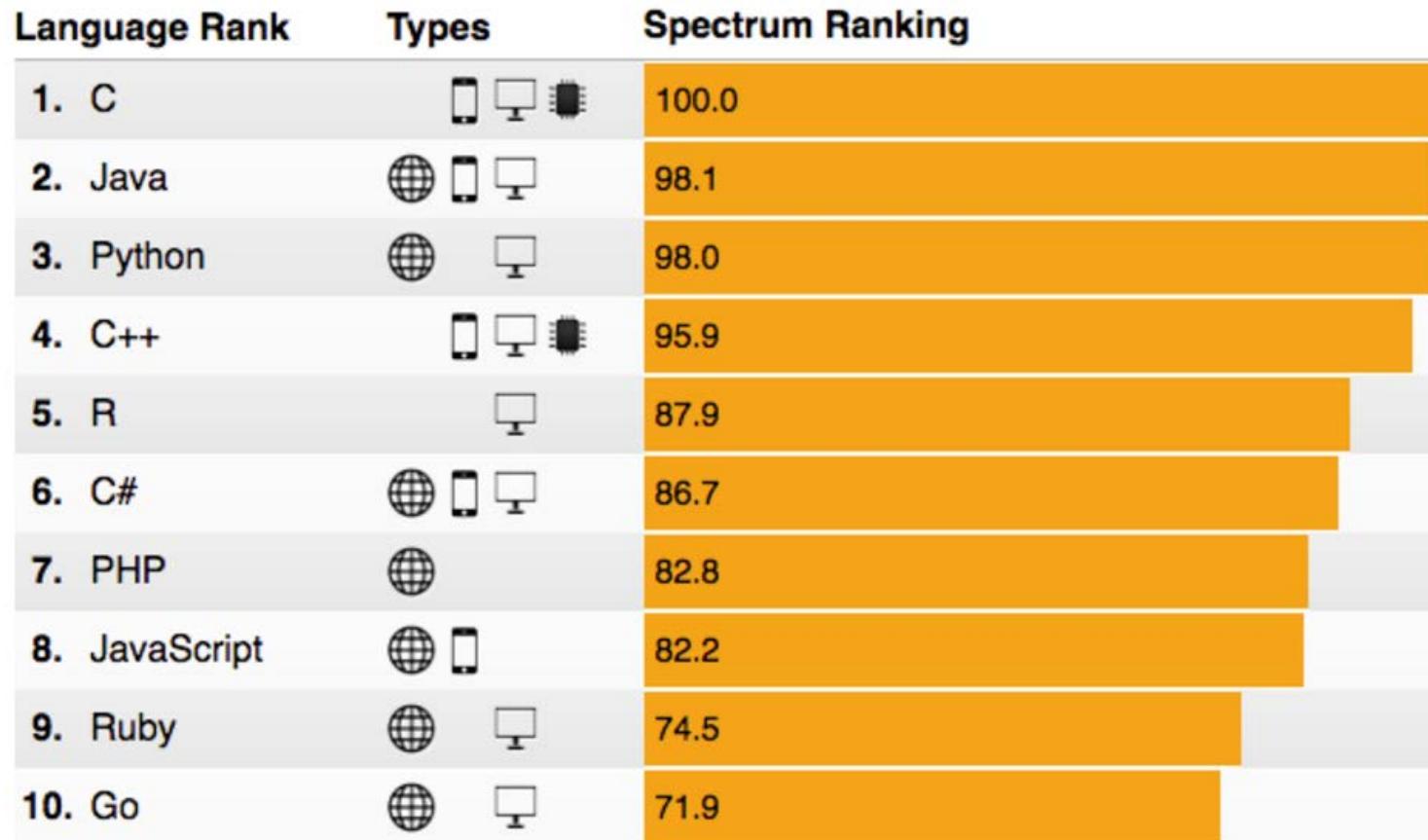
What is Python?

- Computer language
- Interpreter
- Multi-paradigm: (OOP, imperative, functional, procedural)
- Typing: dynamic
- Statistical and graphics
- Linear and nonlinear modelling
- Age 26 (C# 17)
- Free Software (GNU project)
- Linux, Windows and MacOS
- One of the most powerful ML language
- Tool for ML exploration
- Good for building a production model
- Supported in Azure ML Studio

What is Anaconda?

- Anaconda is a python and R distribution
- Provide everything you need for data science "out of the box"
- The core python language
- 100+ python "packages" (libraries)
- Spyder and Jupyter ready
- conda: Anaconda's own package manager
- Microsoft Azure ML Studio fully supports

Popularity rank



Source: The 2016 Top Programming Languages

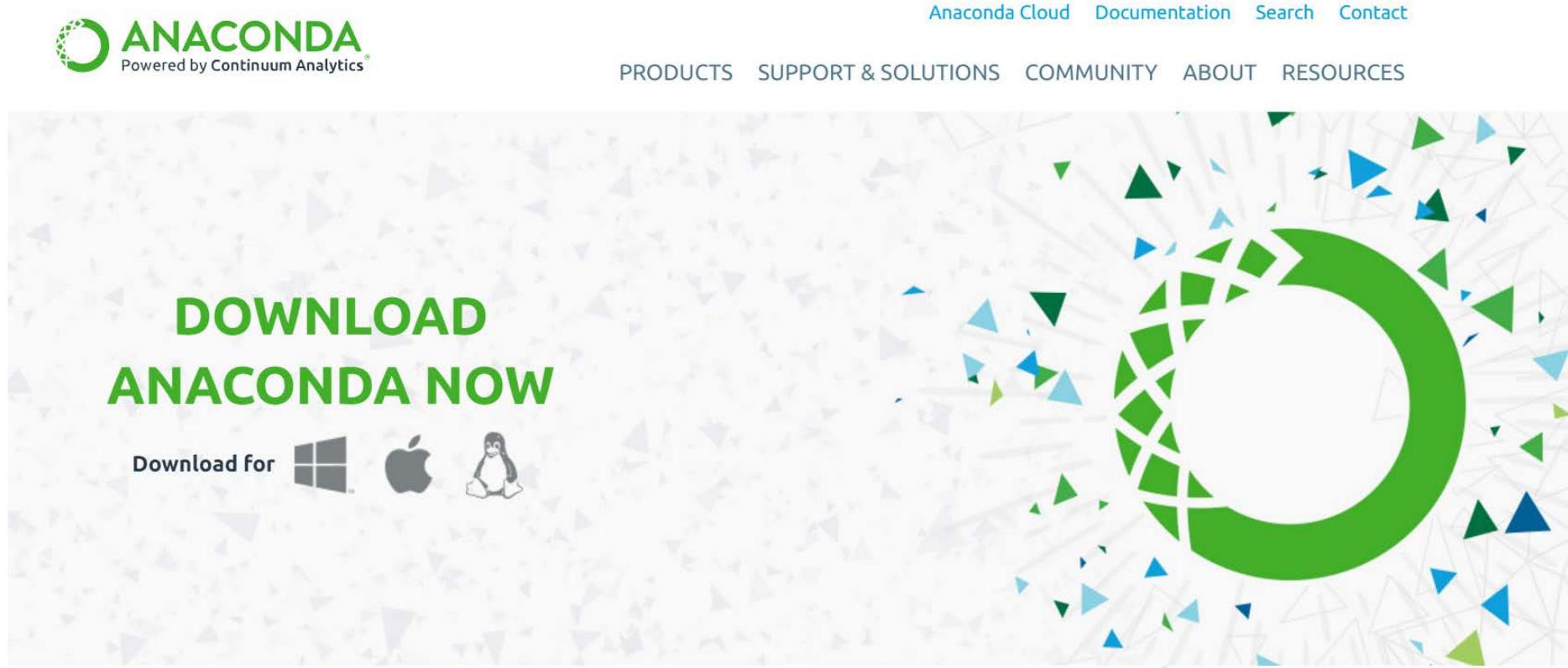
<http://spectrum.ieee.org/computing/software/the-2016-top-programming-languages>

Why use Anaconda in Machine Learning?

- 720+ data science packages
 - visualizations
 - machine learning
 - deep learning
 - Big Data
 - tensor calculation
- Native access
 - HDFS
 - Amazon S3
- Distributed computing
- GPU supercharged
- Agile & fast experimentation data science
- Use Microsoft Excel® to perform predictive analytics
- Query and transform Big Data
- Easily deploy production
- Spyder & Jupiter build-in

Anaconda installation

<https://www.continuum.io/downloads>



Anaconda installation

1. Click Python 3.6 version Python 3.6 version
2. Click Run to launch the installer. [64-BIT INSTALLER \(437M\)](#)
3. Click Next.
4. Read the licensing terms and click I Agree.
5. Select an install for “Just Me”
6. Select a destination folder to install Anaconda and click Next.
7. Choose whether to add Anaconda to your PATH environment variable.
8. Choose whether to register Anaconda as your default Python 3.6.
9. Click Install.
10. Click Next.

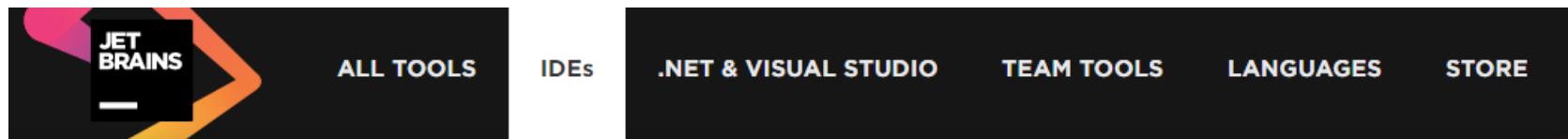
Anaconda installation



11. After a successful installation you will see the “Thanks for installing Anaconda” image:

PyCharm installation

<https://www.jetbrains.com/>



Check out our IDEs

IntelliJ IDEA

The most intelligent Java IDE

Rider

A cross-platform .NET IDE
based on IntelliJ platform and
ReSharper

PyCharm

Python IDE for professional
developers

DataGrip

Many databases, one tool

WebStorm

The smartest JavaScript IDE

RubyMine

The most intelligent Ruby IDE

Toolbox App

A control panel for your tools and projects

PyCharm installation

Click Professional download

The screenshot shows the PyCharm download page on the JetBrains website. At the top, there's a navigation bar with links for ALL TOOLS, IDEs, .NET & VISUAL STUDIO, TEAM TOOLS, LANGUAGES, STORE, SUPPORT, and WE ARE JETBRAINS. Below the navigation bar, the page title is "PyCharm". To the right of the title, it says "Coming in 2017.2 EAP" and links for "What's New", "Features", and "Docs & Demos". A large PyCharm logo is centered on the page. Below the logo, the text "Version: 2017.1.3", "Build: 171.4424.42", and "Released: May 23, 2017" is displayed. There are links for "System requirements" and "Installation Instructions". On the right side, there are two sections: "Download PyCharm" with options for Windows, macOS, and Linux, and "Professional" and "Community" sections. The "Professional" section describes it as a full-featured IDE for Python & Web development, offers a "Free trial", and has a blue "DOWNLOAD" button. The "Community" section describes it as a lightweight IDE for Python & Scientific development, is labeled as "Free, open-source", and has a black "DOWNLOAD" button.

PyCharm

Coming in 2017.2 EAP

What's New Features Docs & Demos

Version: 2017.1.3

Build: 171.4424.42

Released: May 23, 2017

System requirements

Installation Instructions

Download PyCharm

Windows macOS Linux

Professional

Full-featured IDE for Python & Web development

DOWNLOAD

Free trial

Community

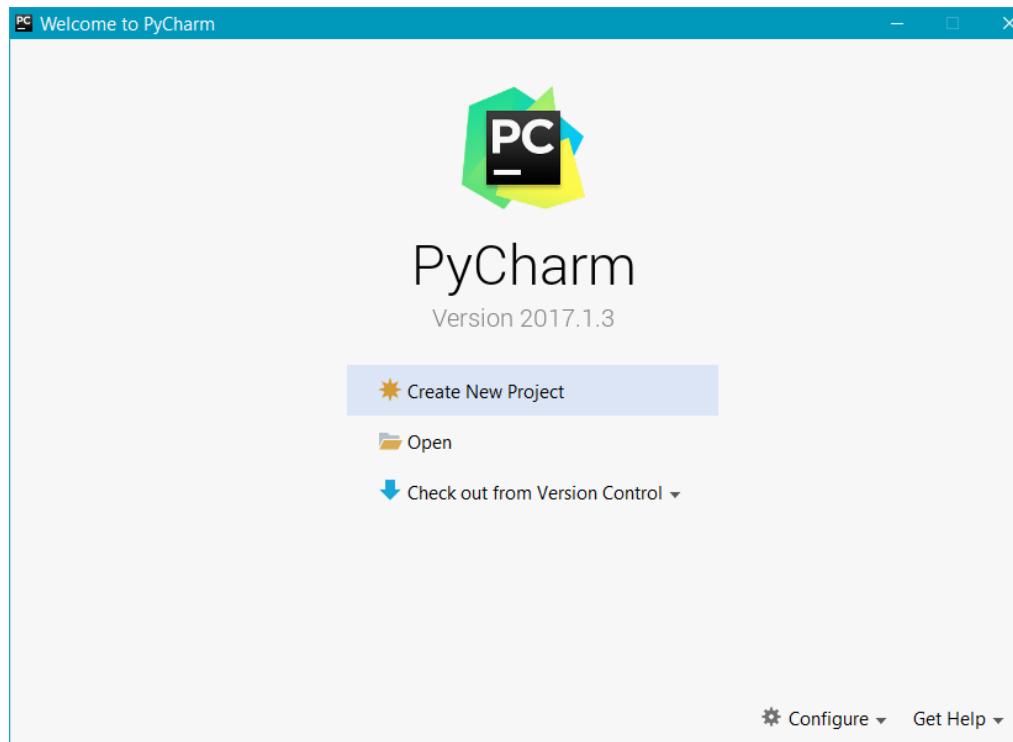
Lightweight IDE for Python & Scientific development

DOWNLOAD

Free, open-source

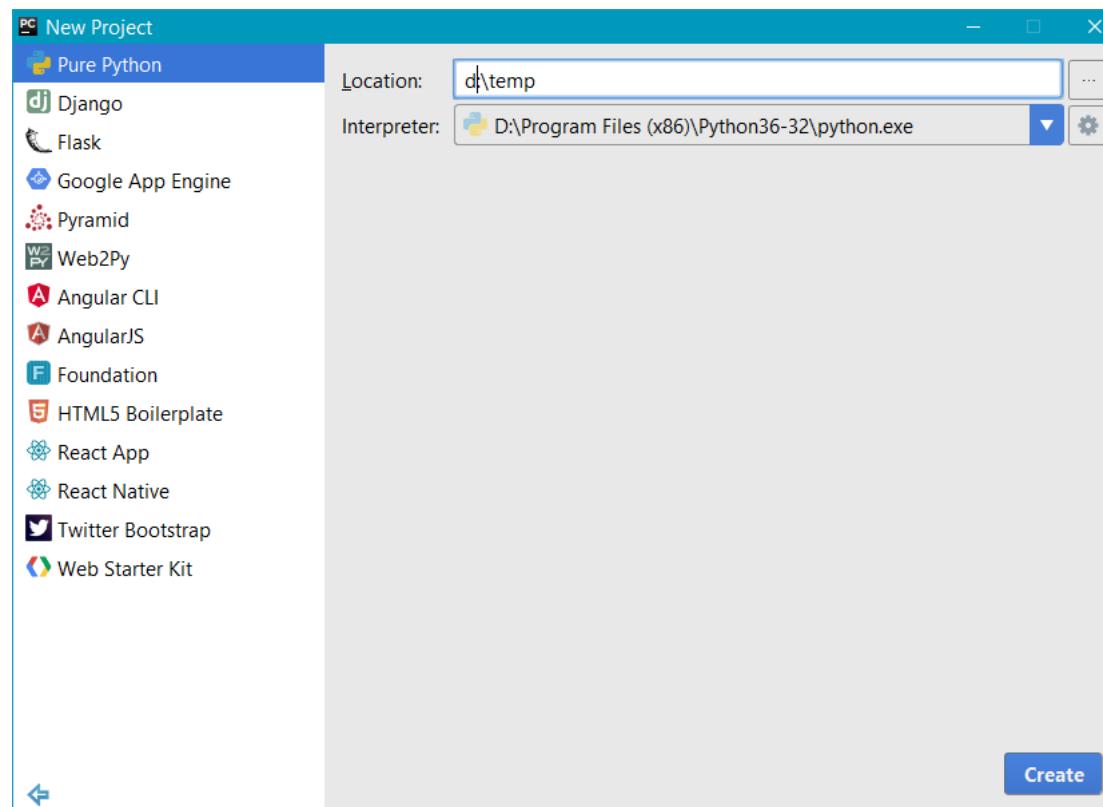
PyCharm installation

Install with all default setting and run



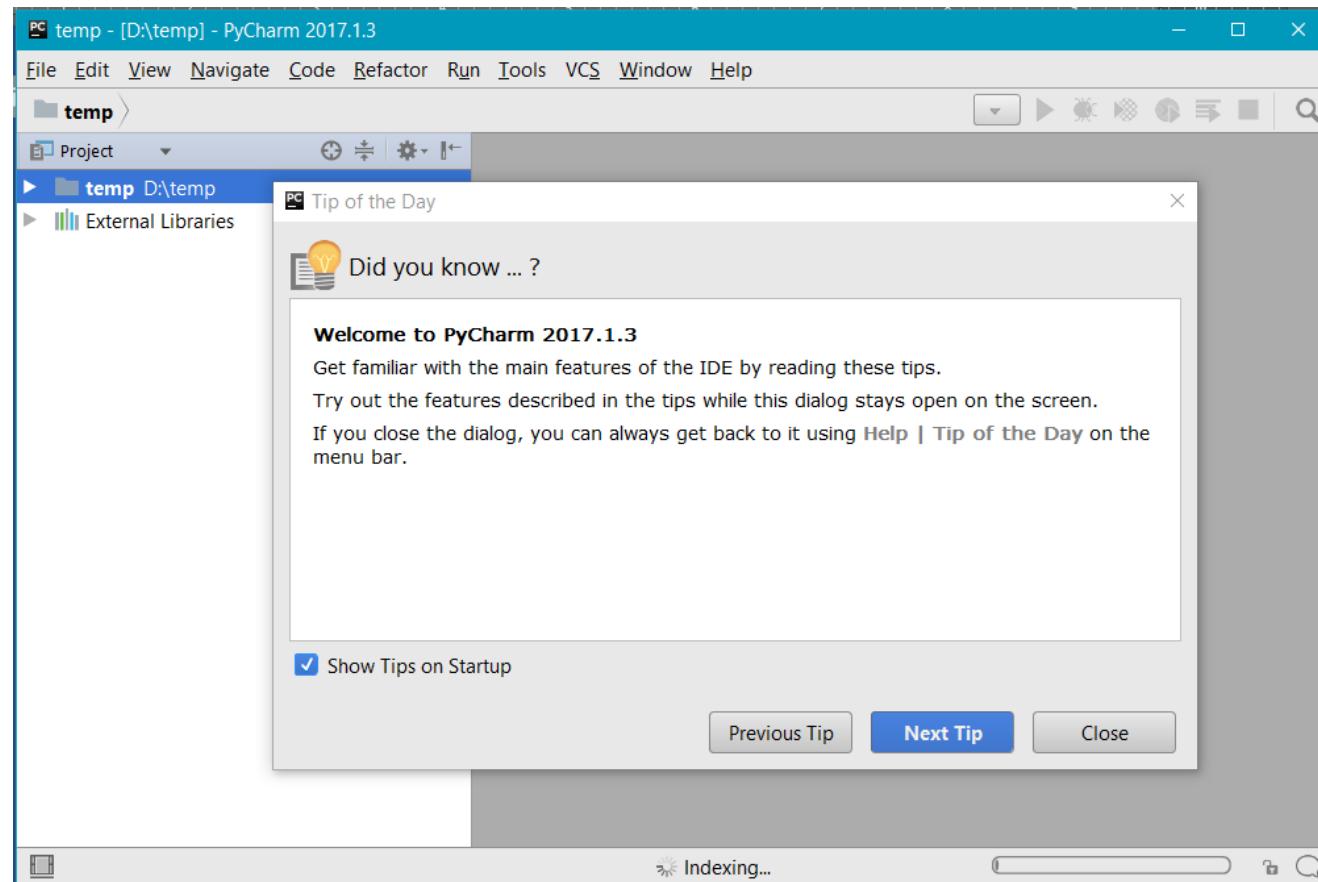
Hello world

1. Create new folder c:\temp (or d:\temp)
2. In Location box type c:\temp (or d:\temp)



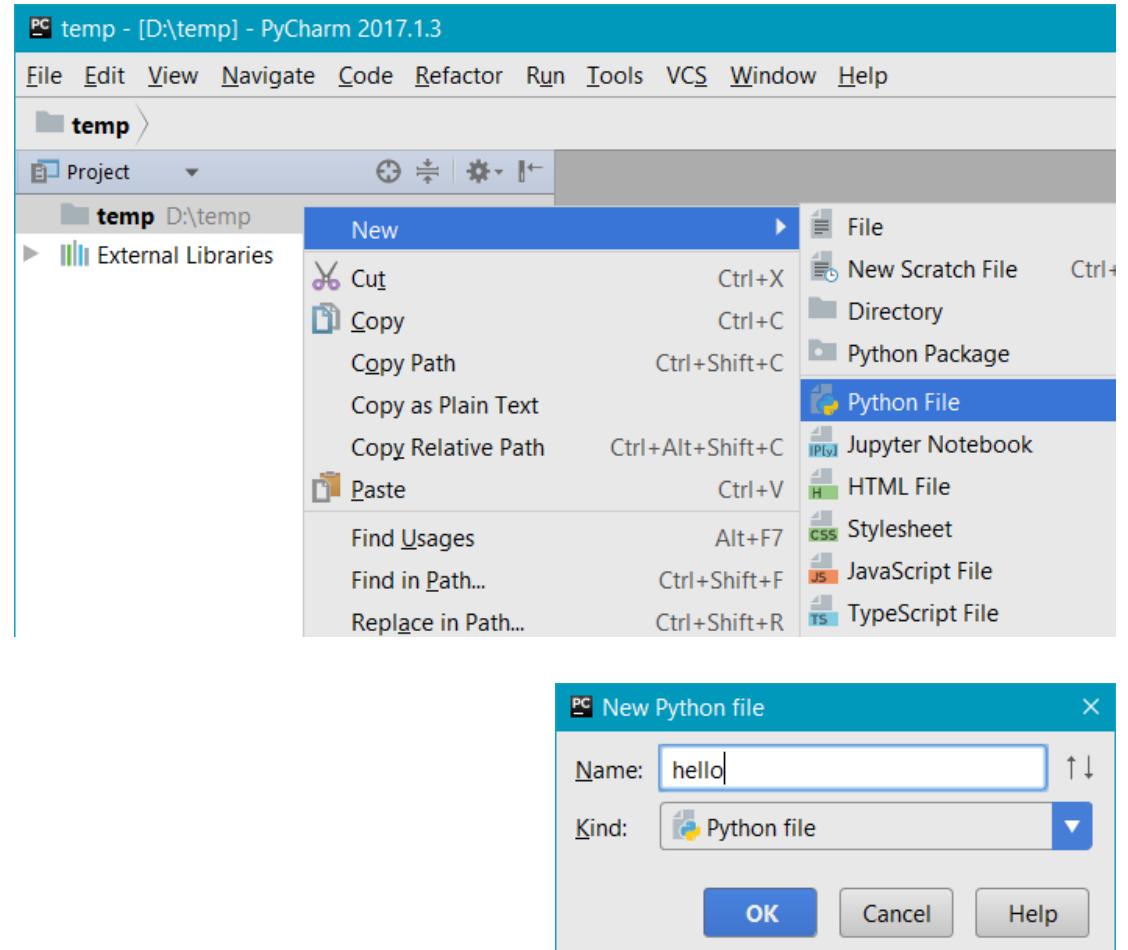
Hello world

3. Click create



Hello world

4. Right click at temp D:\temp
5. Click New
6. Click Python File
7. In Name box type hello
8. Click OK



Hello world

The screenshot shows the PyCharm 2017.1.3 IDE interface. The project navigation bar at the top indicates the current file is `hello.py`. The left sidebar shows a project structure with a single file named `hello.py` under a folder named `temp`. The code editor window displays the following Python code:

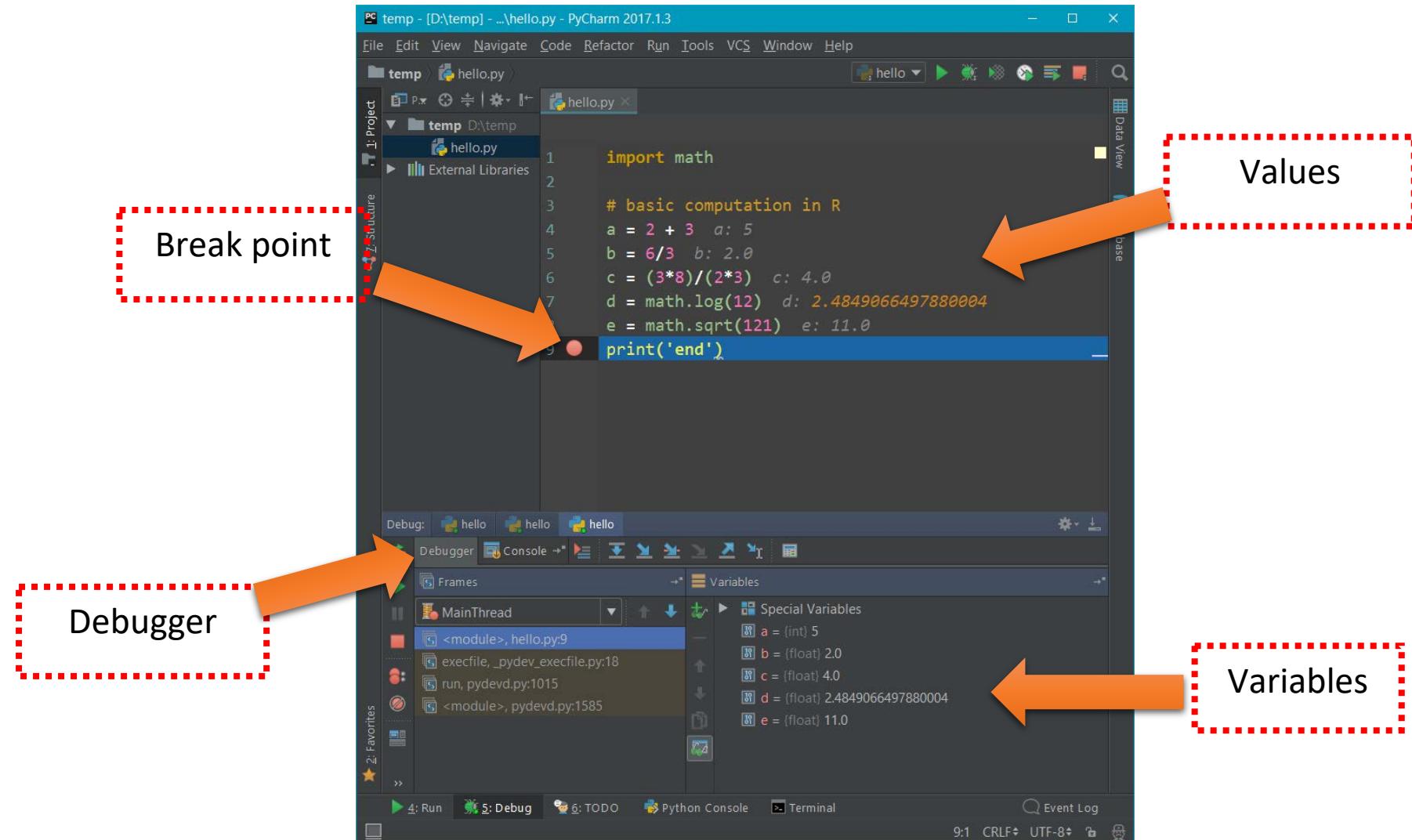
```
1 print('Hello, World!')
```

A context menu is open over the first line of code, listing various actions such as Copy Reference, Paste, Find Usages, Run 'hello', and Save 'hello'. The bottom panel shows the run output, which includes the command used to run the script (`"D:\Program Files (x86)\Python36-32\python.exe" D:/temp/hello.py`), the program's output (`Hello, World!`), and the message `Process finished with exit code 0`.

Basic computation

```
1 import math
2
3 # basic computation in R
4 a = 2 + 3 a: 5
5 b = 6/3 b: 2.0
6 c = (3*8)/(2*3) c: 4.0
7 d = math.log(12) d: 2.4849066497880004
8 e = math.sqrt(121) e: 11.0
9 ● print('end')
```

Debugging in PyCharm



Variable declaration

```
1      # variable assignment & declaration
2      a = 1
3      b = 2
4      c = a + b
5      k = (a * 2) + (b * 3)
6      o = 1; p = 'dog'; q = 3
7      x = y = z = 1           # multiple
8      l, m, n = 1, 2, 'cat'
9  ● print('end')
10
```

Basic data type

```
1 # Basic data type
2 name = 'laploy'          # character
3 who = name + ' v.'       # string concat
4 price = 1500             # integer
5 kilo = 1.5                # floating point
6 x = price + kilo        # automatic type casting
7 yes = True                # bool
8 gen = 123                 # integer
9 gen = 'hello'             # change to string
10 print('end')
```

Basic Operator

```
1 # Basic Operators
2 # Arithmetic
3 a = (2 + 3) * 2 a: 10
4 x, y = 2, 3 x: 2 y: 3
5 b = (x + y) * 2 b: 10
6 c = (2 * 3) / 2 c: 3.0
7 d = 2 / (3 * 4) d: 0.16666666666666666
8 e = 3 * (4/2) e: 6.0
9 # tuple and operator +
10 v = (1, 2, 3) v: <class 'tuple'>: (1, 2, 3)
11 t = (2, 2, 1) t: <class 'tuple'>: (2, 2, 1)
12 f = v + t f: <class 'tuple'>: (1, 2, 3, 2, 2, 1)
13 # relation operator
14 x1 = a > c x1: True
15 x2 = a < d x2: False
16 x3 = a == b x3: True
17 x4 = a != b x4: False
18 ● print('end')
```

Data Structure : List

```
1 L1 = [] # An empty list L1: <class 'list'>: []
2 # Four items: indexes 0..3
3 L2 = [123, 'abc', 1.23, {}] L2: <class 'list'>: [123, 'abc', 1.23, {}]
4 # Nested sublist
5 L3 = ['Bob', 40.0, ['dev', 'mgr']] L3: <class 'list'>: ['Bob', 40.0, ['dev', 'mgr']]
6 L4 = list('spam') L4: <class 'list'>: ['s', 'p', 'a', 'm']
7 L5 = list(range(-4, 4)) L5: <class 'list'>: [-4, -3, -2, -1, 0, 1, 2, 3]
8 x = [1, 2, 3] x: <class 'list'>: [1, 2, 3]
9 y = [1, 2, 3] y: <class 'list'>: [1, 2, 3]
10 z = x + y z: <class 'list'>: [1, 2, 3, 1, 2, 3]
11 # List comprehensions
12 a1 = [v for v in 'SPAM'] a1: <class 'list'>: ['S', 'P', 'A', 'M']
13 a2 = [v * 4 for v in 'cat'] a2: <class 'list'>: ['cccc', 'aaaa', 'tttt']
14 ● print('end')
```

Data Structure : Matrix

```
1 # Matrix
2 matrix = [[1, 2, 3],  matrix: <class 'list'>: [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
3             [4, 5, 6],
4             [7, 8, 9]]
5 a = matrix[1]  # get a row  a: <class 'list'>: [4, 5, 6]
6 b = matrix[1][2]  # get row 1 column 2  b: 6
7 c = [[1, 2, 3],  c: <class 'list'>: [[1, 2, 3], [12, 13, 14], [7, 8, 9]]
8             [4, 5, 6],
9             [7, 8, 9]]
10 c[1] = [12, 13, 14]
11 d = [[1, 2, 3],  d: <class 'list'>: [[1, 2, 3], [], [7, 8, 9]]
12             [4, 5, 6],
13             [7, 8, 9]]
14 d[1] = []
15 print('end')
```

Data structure: Dictionary

```
1 # Dictionary
2 a = {'name': 'loy', 'age': 20, 'gender': 'M'} a: {'name': 'loy', 'age': 20, 'gender': 'M'}
3 b = a['name'] b: 'Loy'
4 c = len(a) c: 3
5 d = 'name' in a d: True
6 e = list(a.keys()) e: <class 'list': ['name', 'age', 'gender']
7 f = list(a.values()) f: <class 'list': [20, 'M', 'loy']
8 g = {'name': 'loy', 'age': 20, 'gender': 'M'} g: {'name': 'loy', 'age': 20, 'gender': 'M'}
9 h = g['name'] = 'lap' h: 'lap'
10 i = {'name': 'loy', 'age': 20, 'gender': 'M'} i: {'name': 'loy', 'age': 20, 'gender': 'M'}
11 # delete entry
12 del i['name']
13 j = {'name': 'loy', 'age': 20, 'gender': 'M'} j: {'name': 'loy', 'age': 20, 'gender': 'M'}
14 # add entry
15 l = j['year'] = 3 l: 3
16 print('end')
```

If statement

```
1 # If Statement
2 x = 1
3 if x == 1:
4     print('same')
5 elif x > 1:
6     print('bigger')
7 else:
8     print('smaller')
9
10 print('end')
```

Loop statement

```
1      # For Loop statement
2      string = "Hello World"
3      s = ''
4      for x in string:
5          s += x + ' '
6      # while loop
7      start, end = 0, 5
8      while start <= end:
9          print('*')
10         start += 1
11     print('end')
```

Function Basic

```
1 # Function
2 def add1(a1, b1):
3     r1 = a1 + b1
4     return r1
5
6 a = add1(2, 4) a: 6
7
8 # anonymous function
9 add2 = lambda a2, b2: a2 + b2
10
11 b = add2(10, 20) b: 30
12
13 print('end')
```

More information

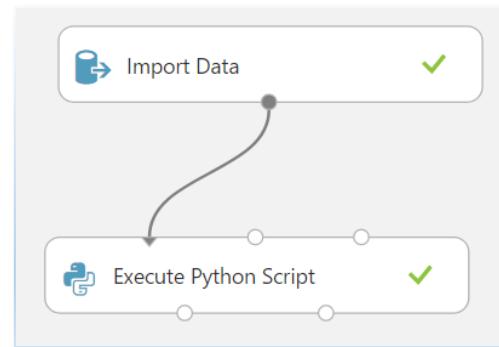
Official Python tutorial

<https://docs.python.org/3/tutorial/>

PyCharm 2017 Quick Start Guide

<https://www.jetbrains.com/help/pycharm/quick-start-guide.html>

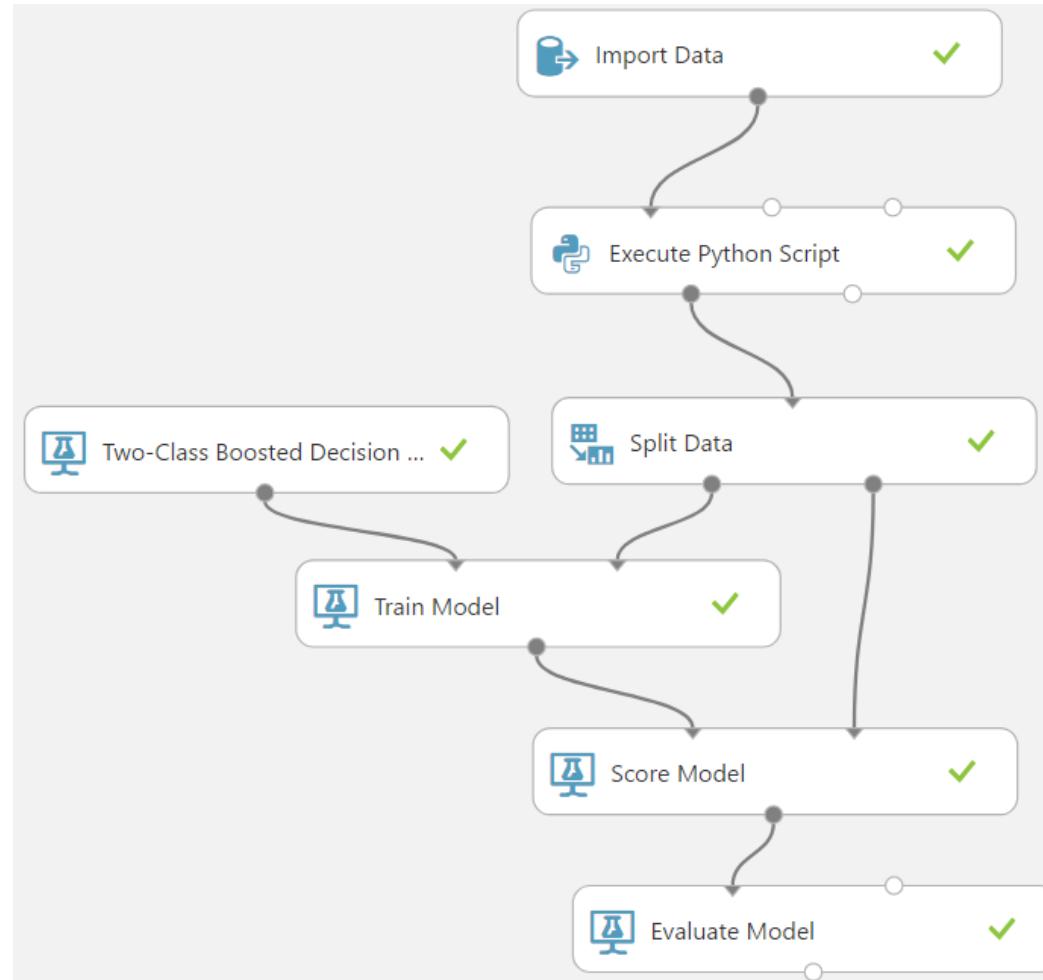
ADDING COLUMN NAME USING PYTHON SCRIPT



In this session:

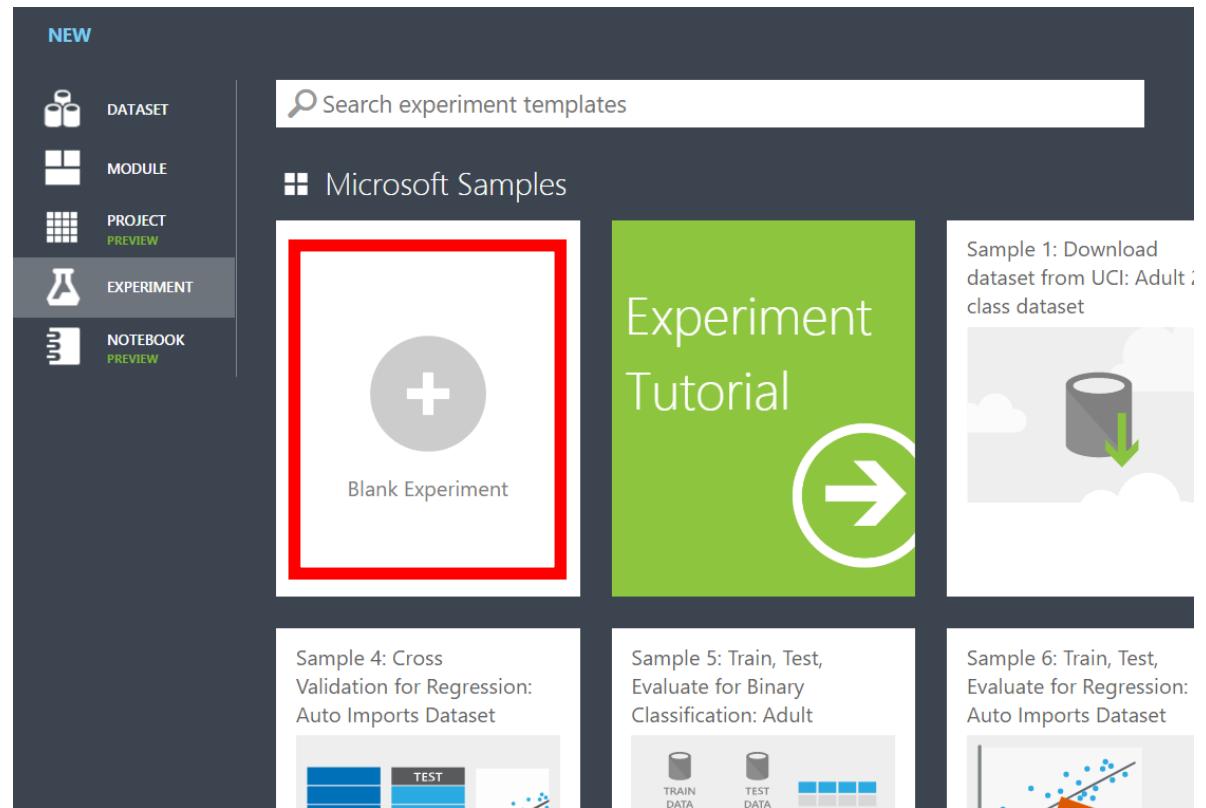
- Create Adult Col name python experiment
- Insert Import data module to experiment's canvas
- Add python module to canvas
- Enter Python script to the module
- Run and debug script
- Make a copy of Adult Income experiment
- Add Python script to Experiment
- More information

Final result experiment



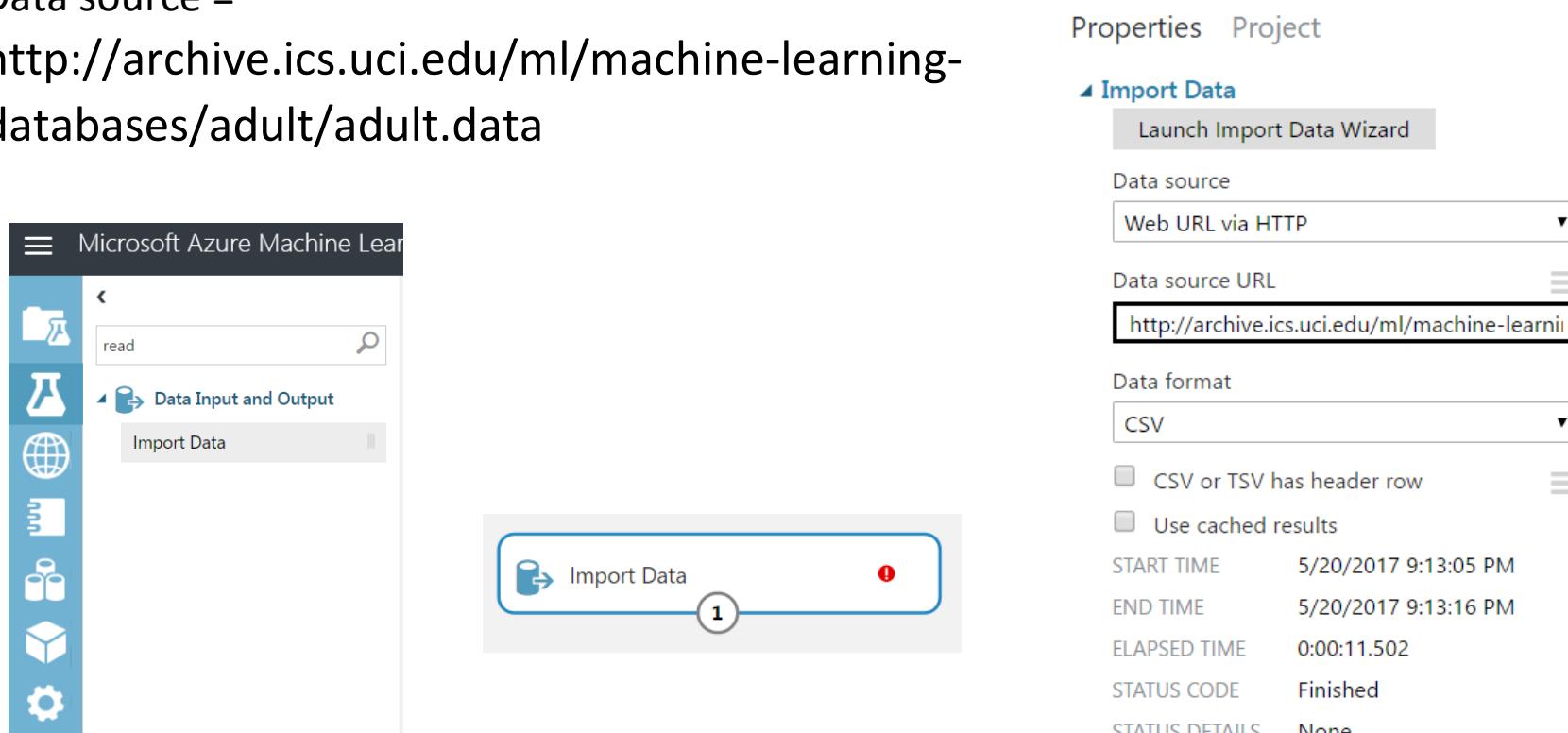
Create Adult Col name python experiment

- Create new Blank Experiment
- Name = Python col name



Insert Import data module to experiment's canvas

- Click Data Input and Output
- Drag & drop Import Data
- Data source =
<http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data>

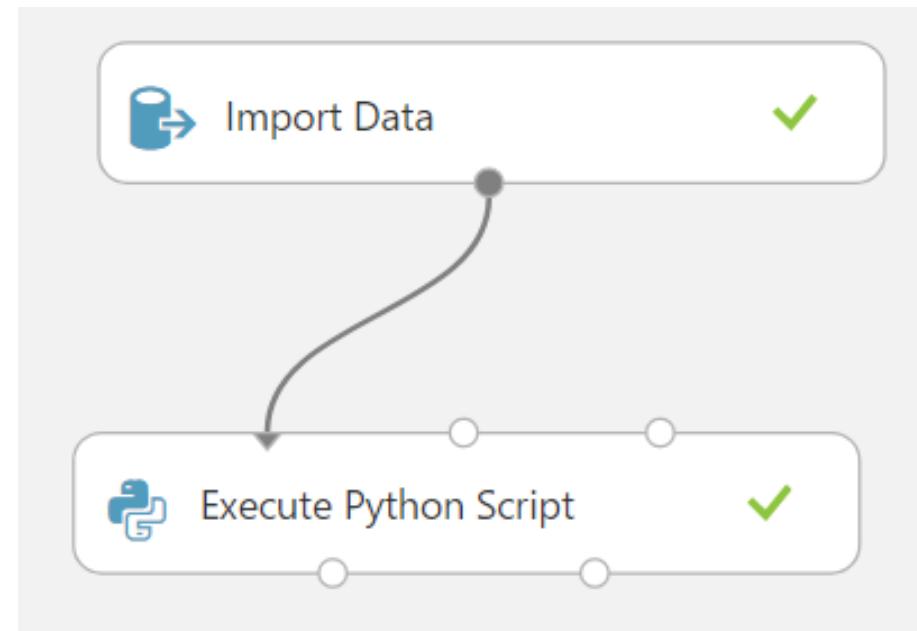


The screenshot shows the Microsoft Azure Machine Learning studio interface. On the left, there is a sidebar with various icons: a folder, a flask, a globe, a document, a gear, and a search bar containing 'read'. Below these are sections for 'Data Input and Output' and 'Import Data'. A large button labeled 'Import Data' with a blue arrow icon is highlighted with a red circle containing the number '1'. To the right, the 'Import Data' properties pane is open, showing the following details:

Properties		Project	
Import Data			
Launch Import Data Wizard			
Data source		Web URL via HTTP	
Data source URL		http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data	
Data format		CSV	
<input type="checkbox"/> CSV or TSV has header row			
<input type="checkbox"/> Use cached results			
START TIME	5/20/2017 9:13:05 PM		
END TIME	5/20/2017 9:13:16 PM		
ELAPSED TIME	0:00:11.502		
STATUS CODE	Finished		
STATUS DETAILS	None		

Add python module to canvas

- Add Execute Python Script to canvas
- Connect Import Data module to Execute Python Script



Enter Python script to the module

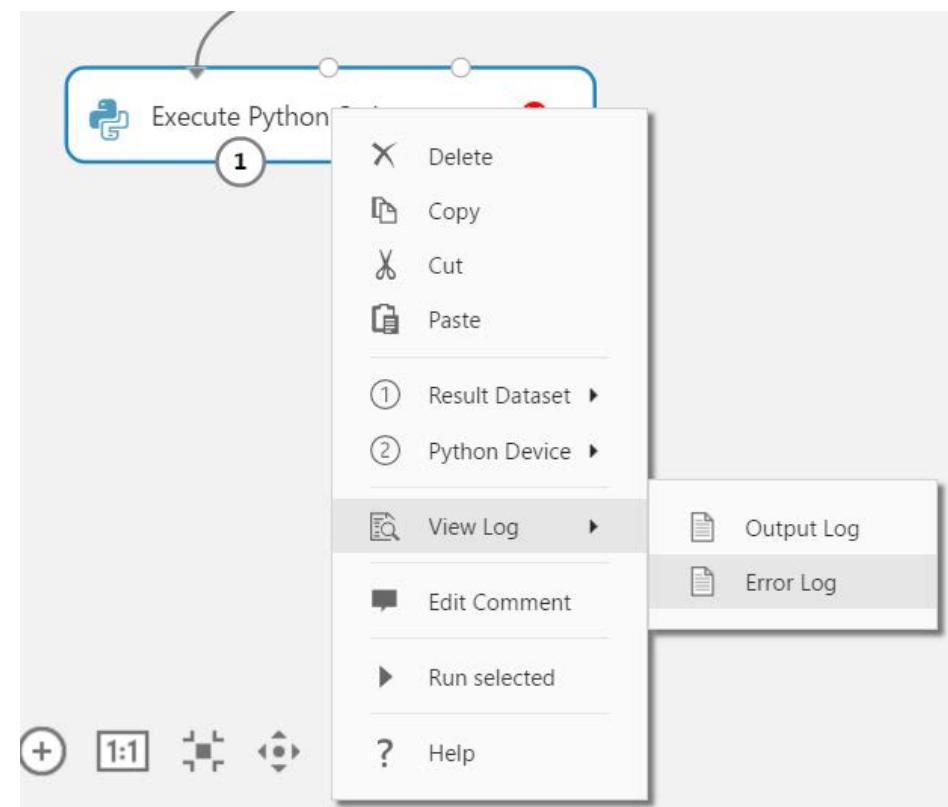
Python script

- Add Python script to Execute Python Script module
- Press 

```
1 #import pandas as pd
2 def azureml_main(dataframe1 = None, dataframe2 = None):
3     dataframe1.columns = [
4         'age',
5         'workclass',
6         'fnlwgt',
7         'education',
8         'education-num',
9         'marital-status',
10        'occupation',
11        'relationship',
12        'race',
13        'sex',
14        'capital-gain',
15        'capital-loss',
16        'hours-per-week',
17        'native-country',
18        'income'
19    ]
20    return dataframe1,
```

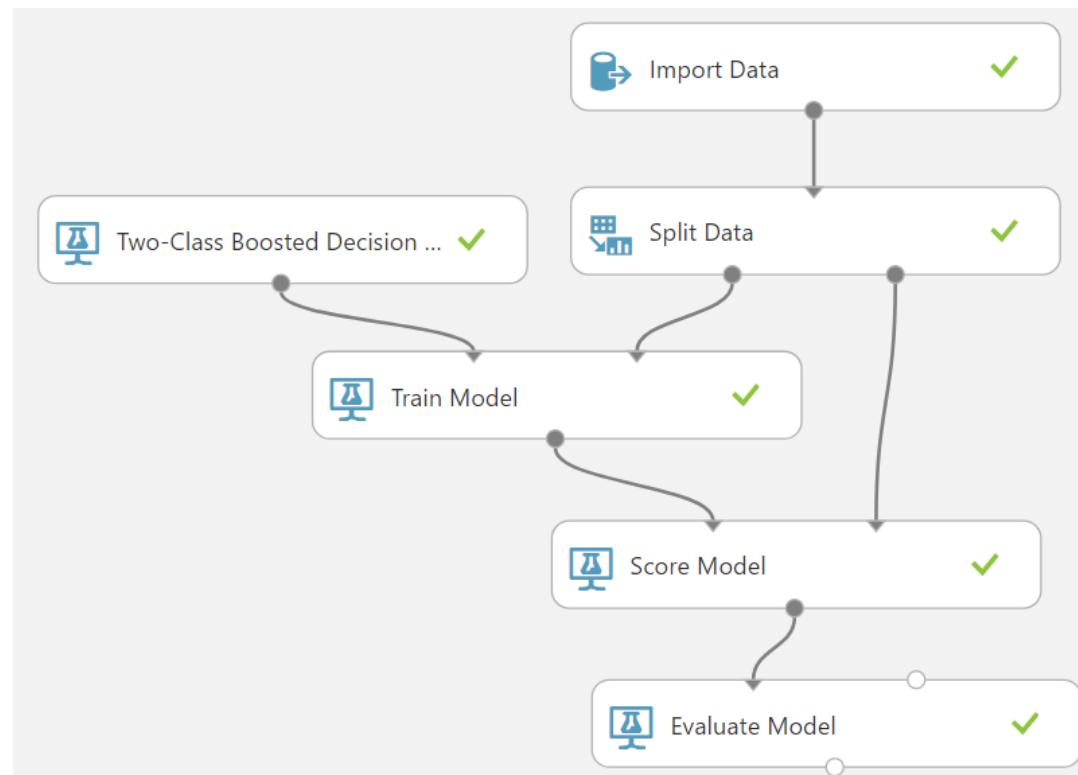
Run and debug script

- Run Experiment
- Debugging Python script
 - View Error Log
 - Edit script
 - Re-Run Experiment
- Visualize the result
- Save the experiment



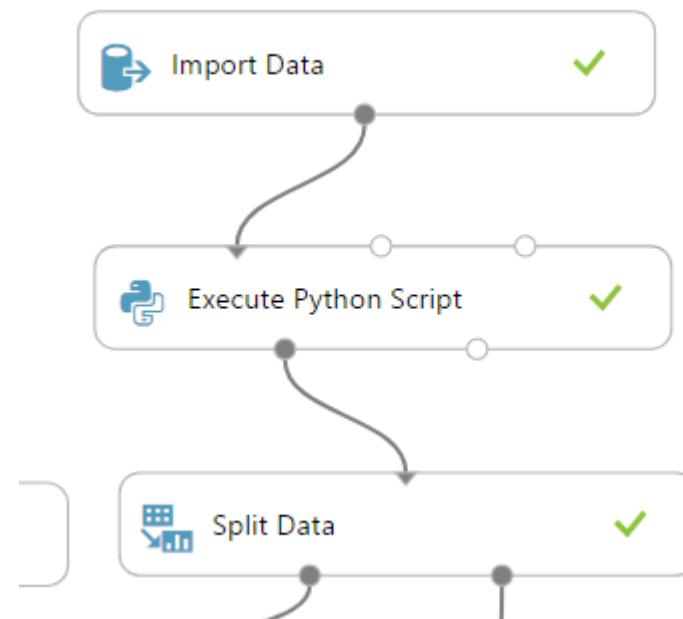
Make a copy of Adult Income experiment

- Open experiment Adult Income
- Save as Adult Income python



Add Python script to Experiment

- Add module Execute Python Script
- Remove connection between Import Data & Split Data
- Connect Import Data to Execute Python Script module
- Enter Script
- Run and debug
- Visualize
- Save experiment



More information

Execute Python machine learning scripts in Azure Machine Learning Studio

<https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-execute-python-scripts>

This experiment

<https://gallery.cortanaintelligence.com/Experiment/Python-Col-name>

PYTHON FEATURE ENGINEERING



In this session

- What is the Feature?
- What is Feature Engineering?
- The process of Feature Engineering
- Where is FE in ML?
- Preparing for experiment
- Adding family size feature
- Adding Age*Class and Fare per person feature
- Adding Deck feature
- Adding Title feature

What is the Feature?

What is the Feature?

- A piece of information
- Might be useful for prediction
- Any useful attribute to the model
- Is measurable property
- Feature is input; label is output.
- Is one column of the data

What is Feature Engineering?

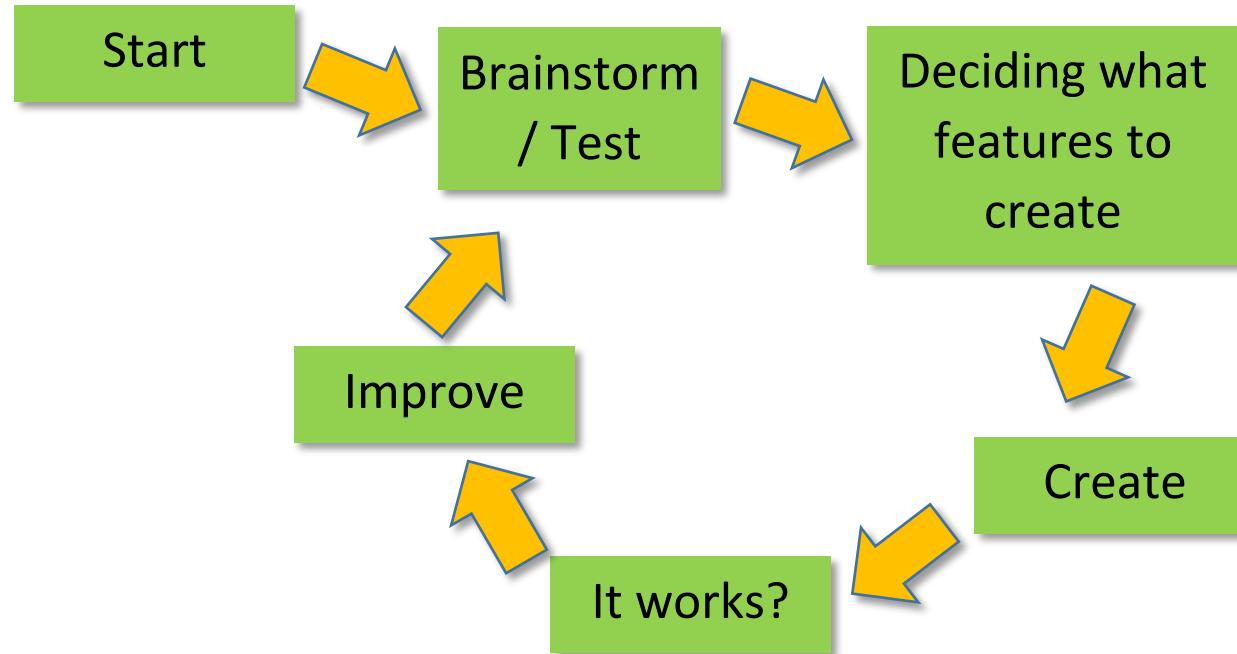
What is Feature Engineering?

- Is the method if find X for input
- Is “Data Science”
- Is difficult
- Is expensive
- Is time-consuming
- Is require expert knowledge in domain
- Is applied machine learning
- Is Yak shaving



The process of feature engineering

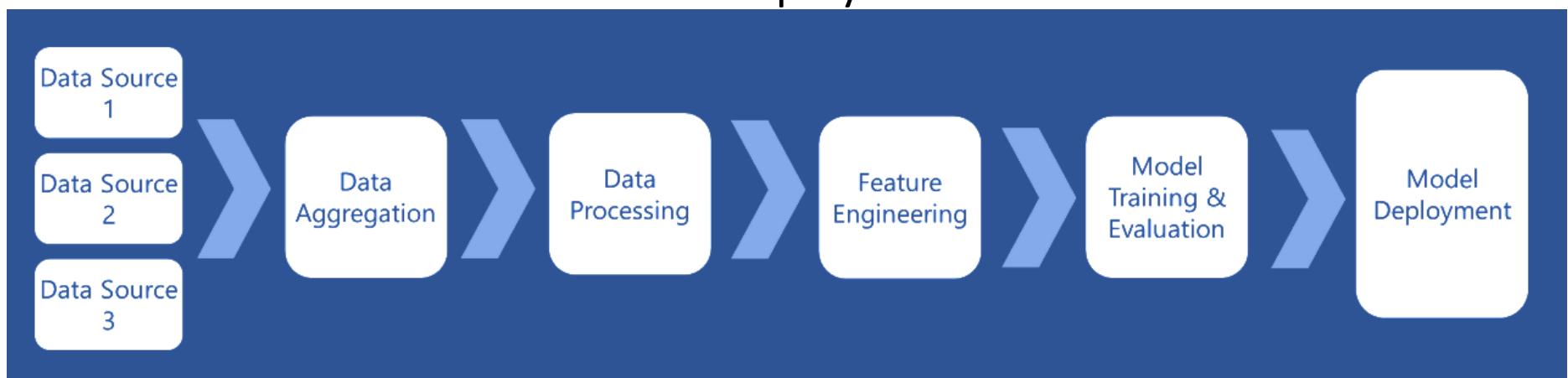
The process of feature engineering



Where is FE in ML?

Where is FE in ML?

- Data sources
- Data aggregation
- Data Processing
- Feature Engineering
- Model Training & Evaluation
- Model Deployment



Preparing for experiment

Preparing for experiment

1. Go to <https://github.com/laploy/ML>
2. Right click TitanicData.csv and save link as to c:\temp
3. Open Pycharm
4. Create New project name = c:\temp\fe
5. Right click project / click Add... / New Python file
6. File name = 100 familySize

Add family size feature File name = 100 familySize

```
9     import pandas as pd
10
11     df = pd.read_csv('d:\\temp\\TitanicData.csv')
12     print(list(df))
13     # Create Family Size
14     df['Family_Size'] = df['SibSp']+df['Parch']
15     print(df[['SibSp', 'Parch', 'Family_Size']].head(10))
16     df.to_csv("d:\\temp\\output.csv")
17     print("end")
18     print("****")
```

	SibSp	Parch	Family_Size
0	1	0	1
1	1	0	1
2	0	0	0
3	1	0	1
4	0	0	0
5	0	0	0
6	0	0	0
7	3	1	4
8	0	2	2
9	1	0	1

Add Age*Class and Fare per person feature

File name = 101 ageClass

```
9     import pandas as pd
10
11     df = pd.read_csv('d:\\temp\\TitanicData.csv')
12     print(list(df))
13     # Create age per class
14     df['Age*Class'] = df['Age']*df['Pclass']
15     print(df[['PassengerId', 'Age', 'Age*Class']].head(10))
16     # Create fare per person
17     df['Family_Size'] = df['SibSp']+df['Parch']
18     df['Fare_Per_Person'] = df['Fare']/(df['Family_Size']+1)
19     print(df[['PassengerId', 'Family_Size', 'Fare_Per_Person']].head(10))
20     print("end")
```

	PassengerId	Age	Age*Class
0	1	22.0	66.0
1	2	38.0	38.0
2	3	26.0	78.0
3	4	35.0	35.0
4	5	35.0	105.0
5	6	NaN	NaN
6	7	54.0	54.0
7	8	2.0	6.0
8	9	27.0	81.0
9	10	14.0	28.0

	PassengerId	Family_Size	Fare_Per_Person
0	1	1	3.62500
1	2	1	35.64165
2	3	0	7.92500
3	4	1	26.55000
4	5	0	8.05000
5	6	0	8.45830
6	7	0	51.86250
7	8	4	4.21500
8	9	2	3.71110
9	10	1	15.03540

Add Deck feature

File name = 102 addDeck

```

9     import pandas as pd
10
11
12     # function to extract title from name
13     def get_deck(main, sub):
14         if type(main) != str: return float('nan')
15         for s in sub:
16             if main.find(s) != -1:
17                 return s
18         return float('nan')
19
20     cabin_list = ['A', 'B', 'C', 'D', 'E', 'F', 'T', 'G', 'Unknown']
21
22     # Turning cabin number into Deck
23     df = pd.read_csv('d:\\temp\\TitanicData.csv')
24     print(list(df))
25     df['Deck'] = df['Cabin'].map(lambda x: get_deck(x, cabin_list))
26     print(df[['PassengerId', 'Cabin', 'Deck']].head(10))
27     print("end")

```

	PassengerId	Cabin	Deck
0	1	NaN	NaN
1	2	C85	C
2	3	NaN	NaN
3	4	C123	C
4	5	NaN	NaN
5	6	NaN	NaN
6	7	E46	E
7	8	NaN	NaN
8	9	NaN	NaN
9	10	NaN	NaN
end			

Adding Title feature

File name = 103 addTitle

```
9     import pandas as pd
10    import numpy as np
11
12    title_list = ['Mrs', 'Mr', 'Master', 'Miss', 'Major', 'Rev',
13                  'Dr', 'Ms', 'Mlle', 'Col', 'Capt', 'Mme', 'Countess',
14                  'Don', 'Jonkheer']
15
16
17    # function to extract title from name
18    def get_title(main, sub):
19        for s in sub:
20            if main.find(s) != -1:
21                return s
22        return np.nan
```

Adding Title feature

```
25     # function to replacing all titles with mr, mrs, miss, master
26     def replace_titles(x):
27         title = x['Title']
28         if title in ['Don', 'Major', 'Capt', 'Jonkheer', 'Rev', 'Col']:
29             return 'Mr'
30         elif title in ['Countess', 'Mme']:
31             return 'Mrs'
32         elif title in ['Mlle', 'Ms']:
33             return 'Miss'
34         elif title == 'Dr':
35             if x['Sex'] == 'Male':
36                 return 'Mr'
37             else:
38                 return 'Mrs'
39         else:
40             return title
```

Adding Title feature

```
43 # here comes the main program
44 df = pd.read_csv('d:\\temp\\TitanicData.csv')
45 print(list(df))
46 df['Title'] = df['Name'].map(lambda x: get_title(x, title_list))
47 df['Title'] = df.apply(replace_titles, axis=1) # 1 = row
48 print(df[['Name', 'Title']].head(10))
49 print("end")
```

	Name	Title
0	Braund, Mr. Owen Harris	Mr
1	Cumings, Mrs. John Bradley (Florence Briggs Th... Heikkinen, Miss. Laina	Mrs Miss
2	Futrelle, Mrs. Jacques Heath (Lily May Peel)	Mrs
3	Allen, Mr. William Henry	Mr
4	Moran, Mr. James	Mr
5	McCarthy, Mr. Timothy J	Mr
6		

More information

Feature engineering in data science

<https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-data-science-create-features>

Source code

<https://github.com/laploy/fe>

MISSING VALUE HANDLING IN PYTHON

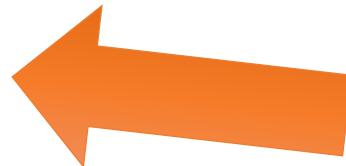
```
import pandas as pd  
import os
```

In this session

14. Replace missing values with the mean
15. Replace missing values with the median
16. Replace missing values with an interpolated estimate
17. Replace missing values with a constant
18. Replace missing values using imputation
19. Replace missing values with a missing rank
20. Replace missing values with a dummy
21. Replace missing values with 0
22. Create an indicator variable for "missing."
23. Replace missing values with a string
24. Add an indicator variable showing which strings are considered "missing."
25. Delete columns that are missing too many values to be useful
26. Delete rows that are missing critical values

We need data that is:

- Relevant
- Connected
- Accurate
- Enough to work with



Example of missing values dataset CSV file

missing_values.csv

	A	B	C	D	E	F	G	H	I
1		age	years_seniority	income	parking_space	attending_party	entree	pets	emergency_contact
2	Tony	48	27		1		5 shrimp		Pepper
3	Donald	67	25	86	10		2 beef		Jane
4	Henry	69	21	95	6		1 chicken	62	Janet
5	Janet	62	21	110	3		1 beef		Henry
6	Nick		17		4				
7	Bruce	37	14	63			1 veggie		NA
8	Steve	83		77	7		1 chicken		n/a
9	Clint	27	9	118	9		shrimp	3	None
10	Wanda	19	7	52	2		2 shrimp		empty
11	Natasha	26	4	162	5		3		-
12	Carol		3	127	11		1 veggie	1	""
13	Mandy	44	2	68	8		1 chicken		null

General commands

```
1 import pandas as pd
2 import os
3
4 # General useful commands
5 os.chdir("d:\\temp")      # change current directory
6 # Read CSV to pandas Data frame
7 df = pd.read_csv('missing_values.csv')
8 print(type(df))          # show data type of df
9 print(list(df))          # show column list
10 print(df)                # show all rows & columns
11 # column projection limit
12 print(df[['name', 'age', 'income']])
13 # row limit
14 print(df[['name', 'age', 'income']].head(n=5))
```

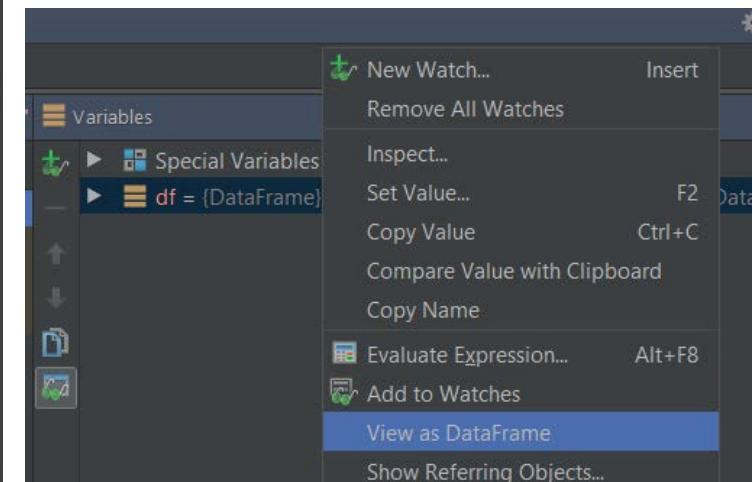
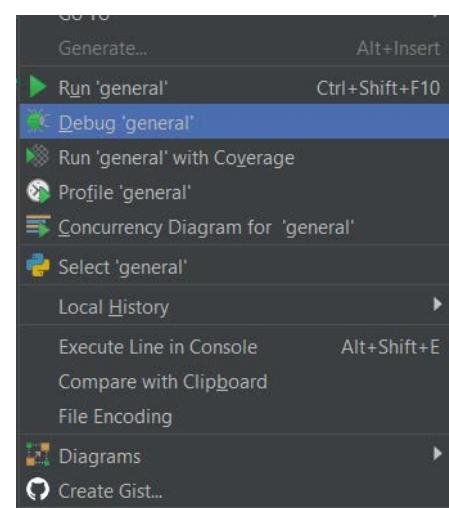
View variable as Data Frame

	name	age	years_senior...	income	parking_sp...	attending_...	entree	pets	emergency...
0	Tony	48.00000	27.00000	nan	1.00000	5.00000	shrimp	nan	Pepper
1	Donald	67.00000	25.00000	86.00000	10.00000	2.00000	beef	nan	Jane
2	Henry	69.00000	21.00000	95.00000	6.00000	1.00000	chicken	62.00000	Janet
3	Janet	62.00000	21.00000	110.00000	3.00000	1.00000	beef	nan	Henry
4	Nick	nan	17.00000	nan	4.00000	nan	nan	nan	nan
5	Bruce	37.00000	14.00000	63.00000	nan	1.00000	veggie	nan	nan
6	Steve	83.00000	nan	77.00000	7.00000	1.00000	chicken	nan	n/a
7	Clint	27.00000	9.00000	118.00000	9.00000	nan	shrimp	3.00000	None
8	Wanda	19.00000	7.00000	52.00000	2.00000	2.00000	shrimp	nan	empty
9	Natasha	26.00000	4.00000	162.00000	5.00000	3.00000	nan	nan	-
10	Carol	nan	3.00000	127.00000	11.00000	1.00000	veggie	1.00000	
11	Mandy	44.00000	2.00000	68.00000	8.00000	1.00000	chicken	nan	null

```

1 + import ...
3
4     # General useful commands
5     os.chdir("d:\temp")      # change
6     # Read CSV to pandas Data frame
7     df = pd.read_csv('missing_values'
8     print(type(df))        # show data type
9     print(list(df))         # show columns
10    print(df)               # show all rows

```



Replace missing values with the mean

```
1 import pandas as pd
2 import os
3
4 # Replace missing values with the mean (age)
5 # column = age
6 os.chdir("d:\\temp")
7 df = pd.read_csv('missing_values.csv')
8 print(type(df))
9 print(list(df))
10 print(df[['name', 'age', 'income']])
11 # replace missing value with mean
12 # and create a new column 'age1'
13 df['age1'] = df[['age']].fillna(df.mean()['age':'age'])
14 print(df[['age', 'age1']])
15 print("end")
16 print("****")
```

age	age1
48.0	48.0
67.0	67.0
69.0	69.0
62.0	62.0
NaN	48.2
37.0	37.0
83.0	83.0
27.0	27.0
19.0	19.0
26.0	26.0
NaN	48.2
44.0	44.0

Replace missing values with the median

```
1 import pandas as pd
2 import os
3
4 # Replace missing values with the median
5 # column = age
6 os.chdir("d:\\temp")
7 df = pd.read_csv('missing_values.csv')
8 print(type(df))
9 print(list(df))
10 print(df[['name', 'age', 'income']])
11 df['age1'] = df[['age']].fillna(df.median()['age':'age'])
12 print(df[['age', 'age1']])
13 print("end")
14 print("****")
```

age	age1
48.0	48.0
67.0	67.0
69.0	69.0
62.0	62.0
NaN	46.0
37.0	37.0
83.0	83.0
27.0	27.0
19.0	19.0
26.0	26.0
NaN	46.0
44.0	44.0

Replace missing values with an interpolated estimate

```
1 import pandas as pd
2 import os
3
4 # Replace missing values with an interpolated estimate
5 # column = years_seniority
6 os.chdir("d:\\temp")
7 df = pd.read_csv('missing_values.csv')
8 print(type(df))
9 print(list(df))
10 print(df[['name', 'age', 'years_seniority']])
11 df['years_seniority1'] = df[['years_seniority']].fillna(11.5)
12 print(df[['years_seniority', 'years_seniority1']])
13 print("end")
14 print("****")
```

years_seniority	years_seniority1
27.0	27.0
25.0	25.0
21.0	21.0
21.0	21.0
17.0	17.0
14.0	14.0
NaN	11.5
9.0	9.0
7.0	7.0
4.0	4.0
3.0	3.0
2.0	2.0

Replace missing values with a constant

```
1 import pandas as pd
2 import os
3
4 # Replace missing values with a constant
5 # column = income
6 os.chdir("d:\\temp")
7 df = pd.read_csv('missing_values.csv')
8 print(type(df))
9 print(list(df))
10 print(df[['name', 'age', 'income']])
11 df['income1'] = df[['income']].fillna(250)
12 print(df[['income', 'income1']])
13 print("end")
14 print("****")
```

income	income1
NaN	250.0
86.0	86.0
95.0	95.0
110.0	110.0
NaN	250.0
63.0	63.0
77.0	77.0
118.0	118.0
52.0	52.0
162.0	162.0
127.0	127.0
68.0	68.0

Replace missing values using imputation (MICE)

The screenshot shows a Jupyter Notebook interface with several files listed in the top navigation bar: bayesian_ridge_regr..., common.py, mice.py, and solver.py. Below the files, a code cell contains the following Python code:

```
1 import mice
2 import pandas as pd
3 import os
4 import numpy as np
5
6 os.chdir("d:\\temp")
7 df = pd.read_csv('missing_values.csv')
8 df2 = df[['age', 'years_seniority', 'income']]
9 a = np.array(df2)
10 x = mice.MICE().complete(a)
11 print(df2)
12 print('-----')
13 print(x)
```

To the right of the code cell, a data frame is displayed with columns 'age', 'years_seniority', and 'income'. The data consists of 13 rows of numerical values.

	age	years_seniority	income
0	48.	27.	91.01255587
1	67.	25.	86.
2	69.	21.	95.
3	62.	21.	110.
4	49.20878949	17.	90.94503833
5	37.	14.	63.
6	83.	26.37257548	77.
7	27.	9.	118.
8	19.	7.	52.
9	26.	4.	162.
10	25.13935223	3.	127.
11	44.	2.	68.

Replace missing values with a missing rank

```
1 import pandas as pd
2 import os
3
4 # Replace missing values with a missing rank
5 # column = parking_space
6 os.chdir("d:\\temp")
7 df = pd.read_csv('missing_values.csv')
8 print(type(df))
9 print(list(df))
10 print(df[['name', 'age', 'parking_space']])
11 # Missing one might be 12
12 df['park1'] = df[['parking_space']].fillna(12)
13 print(df[['parking_space', 'park1']])
14 print("end")
15 print("****")
```

	parking_space	park1
	1.0	1.0
	10.0	10.0
	6.0	6.0
	3.0	3.0
	4.0	4.0
	NaN	12.0
	7.0	7.0
	9.0	9.0
	2.0	2.0
	5.0	5.0
	11.0	11.0
	8.0	8.0

Replace missing values with a dummy

```
1 import pandas as pd
2 import os
3
4 # Replace missing values with a dummy
5 # column = parking_space
6 os.chdir("d:\\temp")
7 df = pd.read_csv('missing_values.csv')
8 print(type(df))
9 print(list(df))
10 print(df[['name', 'age', 'parking_space']])
11 # dummy is -99
12 df['park1'] = df[['parking_space']].fillna(-99)
13 print(df[['parking_space', 'park1']])
14 print("end")
15 print("****")
```

	parking_space	park1
	1.0	1.0
	10.0	10.0
	6.0	6.0
	3.0	3.0
	4.0	4.0
	NaN	-99.0
	7.0	7.0
	9.0	9.0
	2.0	2.0
	5.0	5.0
	11.0	11.0
	8.0	8.0

Replace missing values with 0

```
1 import pandas as pd
2 import os
3
4 # Replace missing values with 0
5 # column = attending_party
6 os.chdir("d:\\temp")
7 df = pd.read_csv('missing_values.csv')
8 print(type(df))
9 print(list(df))
10 print(df[['name', 'age', 'attending_party']])
11 df['party'] = df[['attending_party']].fillna(0)
12 print(df[['attending_party', 'party']])
13 print("end")
```

attending_party	party
5.0	5.0
2.0	2.0
1.0	1.0
1.0	1.0
NaN	0.0
1.0	1.0
1.0	1.0
NaN	0.0
2.0	2.0
3.0	3.0
1.0	1.0
1.0	1.0

Create an indicator variable for "missing"

```
1 import pandas as pd
2 import os
3
4 # Create an indicator variable for "missing"
5 # column = pets
6 os.chdir("d:\\temp")
7 df = pd.read_csv('missing_values.csv')
8 df['pets1'] = df[['pets']].fillna(0)
9 df['pets2'] = df[['pets1']].isin([0])
10 print(df[['name', 'pets', 'pets1', 'pets2']])
11 print("end")
```

	name	pets	pets1	pets2
1	Tony	NaN	0.0	True
2	Donald	NaN	0.0	True
3	Henry	62.0	62.0	False
4	Janet	NaN	0.0	True
5	Nick	NaN	0.0	True
6	Bruce	NaN	0.0	True
7	Steve	NaN	0.0	True
8	Clint	3.0	3.0	False
9	Wanda	NaN	0.0	True
10	Natasha	NaN	0.0	True
11	Carol	1.0	1.0	False
12	Mandy	NaN	0.0	True

Replace missing values with a string

```
1 import pandas as pd
2 import os
3
4 # Replace missing values with a string
5 # column = emergency_contact
6 os.chdir("d:\\temp")
7 df = pd.read_csv('missing_values.csv')
8 df['e1'] = df[['emergency_contact']].fillna('no')
9 print(df[['emergency_contact', 'e1']])
10 print("end")
```

emergency_contact	e1
Pepper	Pepper
Jane	Jane
Janet	Janet
Henry	Henry
NaN	no
NaN	no
n/a	n/a
None	None
empty	empty
..	..
null	null

Add an indicator variable showing which strings are considered "missing."

```
1 import pandas as pd
2 import os
3
4 # Add an indicator variable showing which
5 # strings are considered "missing."
6 # column = emergency_contact
7 os.chdir("d:\\temp")
8 df = pd.read_csv('missing_values.csv')
9 k = ['NA', 'n/a', 'None', 'empty', '_', "", 'null']
10 df['e1'] = df[['emergency_contact']].isin(k)
11 print(df[['name', 'emergency_contact', 'e1']])
12 print("end")
```

	name	emergency_contact	e1
0	Tony	Pepper	False
1	Donald	Jane	False
2	Henry	Janet	False
3	Janet	Henry	False
4	Nick	NaN	False
5	Bruce	NaN	False
6	Steve	n/a	True
7	Clint	None	True
8	Wanda	empty	True
9	Natasha	-	True
10	Carol	""	True
11	Mandy	null	True

Delete columns that are missing too many values to be useful

```
1 import pandas as pd
2 import os
3
4 # Delete columns that are missing too
5 # many values to be useful
6 # column = pets
7 os.chdir("d:\\temp")
8 df = pd.read_csv('missing_values.csv')
9 del df['pets']
10 print(df)
11 print("end")
```

Delete rows that are missing critical values

```
1 import pandas as pd
2 import os
3
4 # Delete row that are missing too
5 # many values to be useful
6 os.chdir("d:\\temp")
7 df = pd.read_csv('missing_values.csv')
8 df = df.dropna(how='any')    # 'all'
9 print(df)
10 print("end")
```

More information on Missing value handling in Python

Pandas 0.20.1 documentation: Working with missing data

https://pandas.pydata.org/pandas-docs/stable/missing_data.html

Source code

<https://github.com/laploy/ML/blob/master/missing%20value%20python%20msvs.zip>

R SCRIPT INTRODUCTION



In this session

- What is R?
- R current popularity rank
- Why use R language in Machine Learning?
- R Script interpreter installation
- R Studio installation
- Hello world
- Basic calculation
- Variable assignment
- Basic Operator
- Data Structure (Array, Matrix, List, Data Frame)
- If Statement
- For Loop
- Basic plotting

What is R?

- Computer language
- Interpreter
- Multi-paradigm: (OOP, imperative, functional, procedural)
- Typing: dynamic
- Good for: Statistical and graphics
- Origin: New Zealand
- Age: 23 (C# 17)
- Free Software (GNU project)
- Linux, Windows and MacOS
- One of the most powerful ML language
- Tool for ML exploration
- NOT for building a production model
- Supported in Azure ML Studio

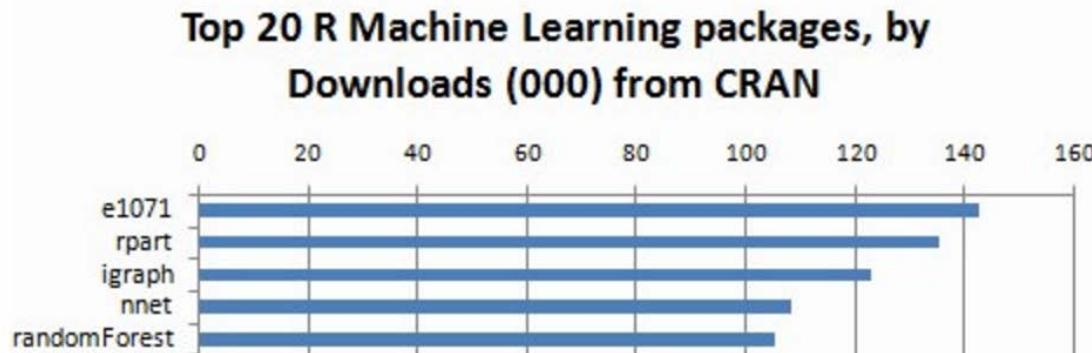
R current popularity rank



Source: The 2016 Top Programming Languages
<http://spectrum.ieee.org/static/interactive-the-top-programming-languages-2016>



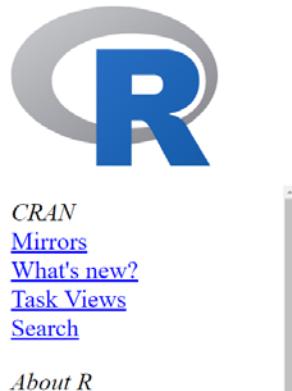
Why use R language in Machine Learning?



- Microsoft Azure ML support
- Free and open source
- Data Scientist's tools of trade
- Simple syntax
- Large community
- Over 7,800 package listed on CRAN
- Good ML packages (e1071, caret, etc.)
- Visualizations
- Full-set tools

R Script interpreter installation

1. Go to CRAN website <https://cran.rstudio.com>



The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

2. Click download R for Windows

3. Click base

R for Windows

Subdirectories:

- [base](#) (circled in red)
- [contrib](#)
- [old contrib](#)
- [Rtools](#)

Binaries for base distribution (managed by Duncan Murdoch). Want to [install R for the first time](#).
Binaries of contributed CRAN packages (for R >= 3.5.0). There is also information on [third party software](#).
Binaries of contributed CRAN packages for older versions of R (managed by Uwe Ligges).
Tools to build R and R packages (managed by Duncan Murdoch). Want to build your own packages on Windows, or

R Script interpreter installation

4. Click Download R 3.4.0 for Windows (76 megabytes, 32/64 bit)



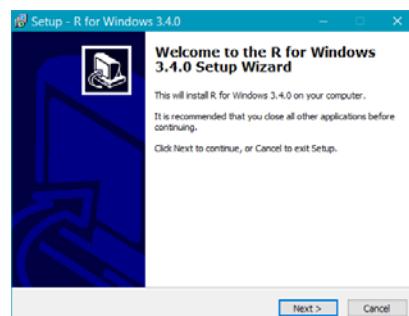
[CRAN](#)
[Mirrors](#)
[What's new?](#)
[Task Views](#)
[Search](#)

R-3.4.0 for Windows (32/64 bit)

[Download R 3.4.0 for Windows](#) (76 megabytes, 32/64 bit)
[Installation and other instructions](#)
[New features in this version](#)

If you want to double-check that the package you have downloaded matches the package you can compare the [md5sum](#) of the .exe to the [fingerprint](#) on the master server. You will md5sum for windows: both [graphical](#) and [command line versions](#) are available.

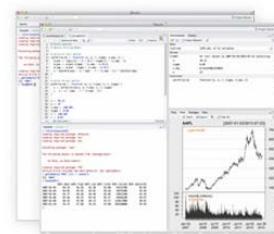
When downloading done, open downloaded file to run setup



R Studio installation

Go to R Studio download page

<https://www.rstudio.com/products/rstudio/download/>



Choose Your Version of RStudio

RStudio is a set of integrated tools designed to console, syntax-highlighting editor that support tools for plotting, viewing history, debugging and RStudio features.

Click RStudio Desktop FREE download

RStudio Desktop
Open Source License

FREE

R Studio installation

1. Click RStudio 1.0.143 - Windows Vista/7/8/10

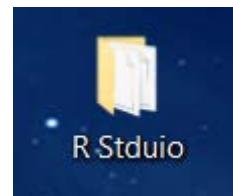
Installers for Supported Platforms

Installers

- [RStudio 1.0.143 - Windows Vista/7/8/10](#)
- [RStudio 1.0.143 - Mac OS X 10.6+ \(64-bit\)](#)
- [RStudio 1.0.143 - Ubuntu 12.04+/Debian 8+ \(32-bit\)](#)
- [RStudio 1.0.143 - Ubuntu 12.04+/Debian 8+ \(64-bit\)](#)
- [RStudio 1.0.143 - Fedora 19+/RedHat 7+/openSUSE 13.1+ \(32-bit\)](#)
- [RStudio 1.0.143 - Fedora 19+/RedHat 7+/openSUSE 13.1+ \(64-bit\)](#)

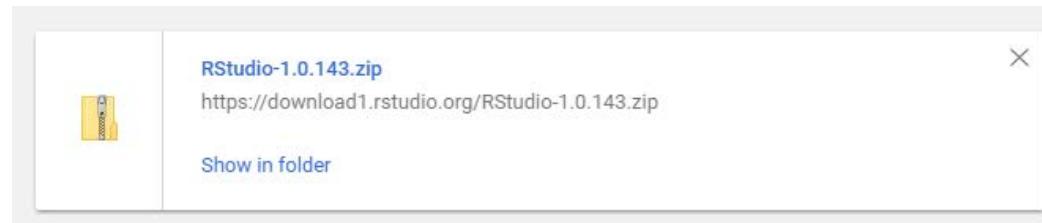


2. Create a folder R Studio on the desktop



R Studio installation

3. Open Zip file

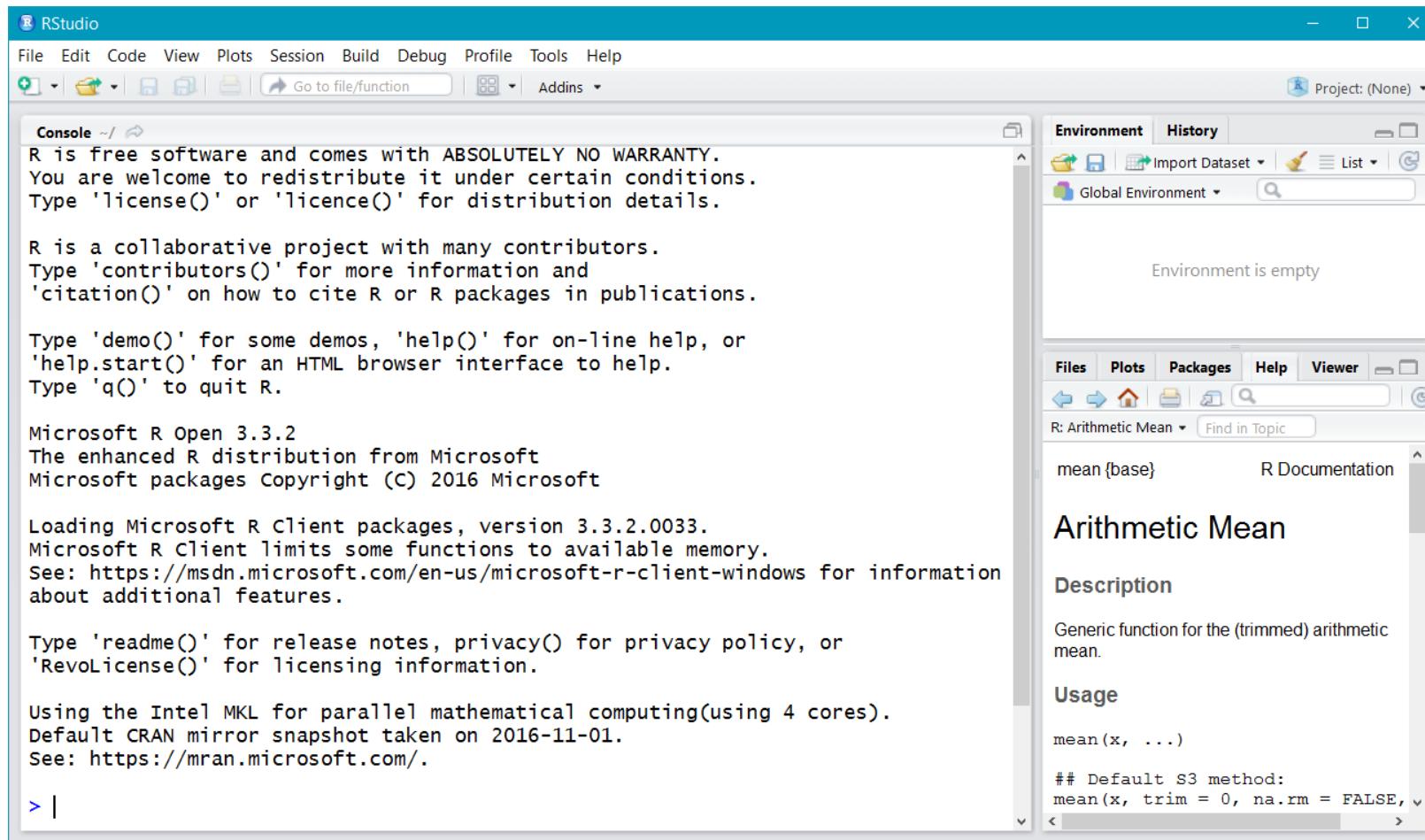


4. Drag & drop items from zip file to folder R Studio

Name	Type	Compressed size	Size
📁 bin	File folder		
📁 R	File folder		
📁 resources	File folder		
📁 www	File folder		
📁 www-symbolmaps	File folder		
📄 COPYING	File	12 KB	35 KB
📄 INSTALL	File	3 KB	6 KB
📄 NOTICE	File	47 KB	167 KB
📄 README.md	MD File	1 KB	2 KB
📄 SOURCE	File	1 KB	1 KB
📄 VERSION	File	1 KB	1 KB

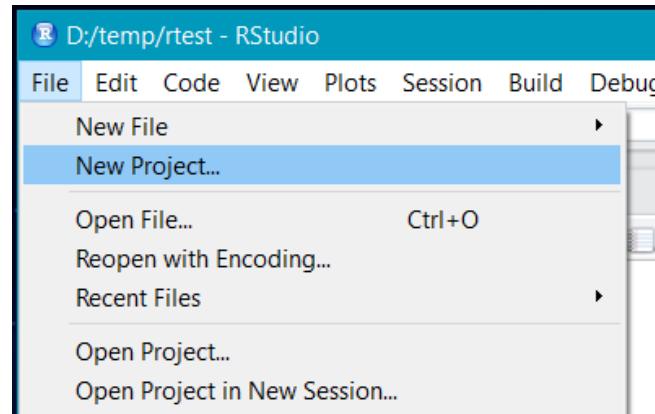
R Studio installation

5. Click icon C:\Desktop\R Studio\bin\rstudio.exe

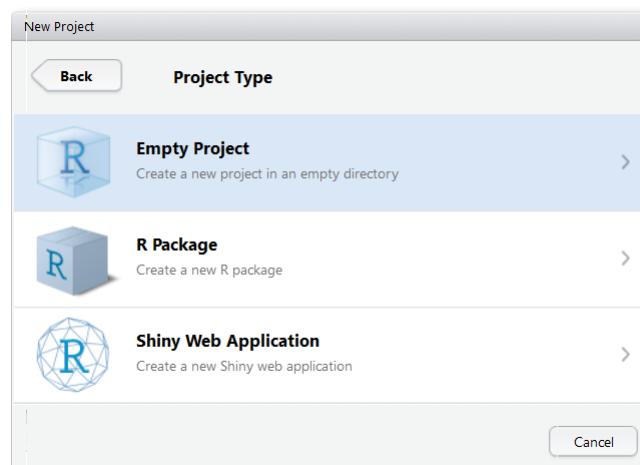


Hello world

Create project

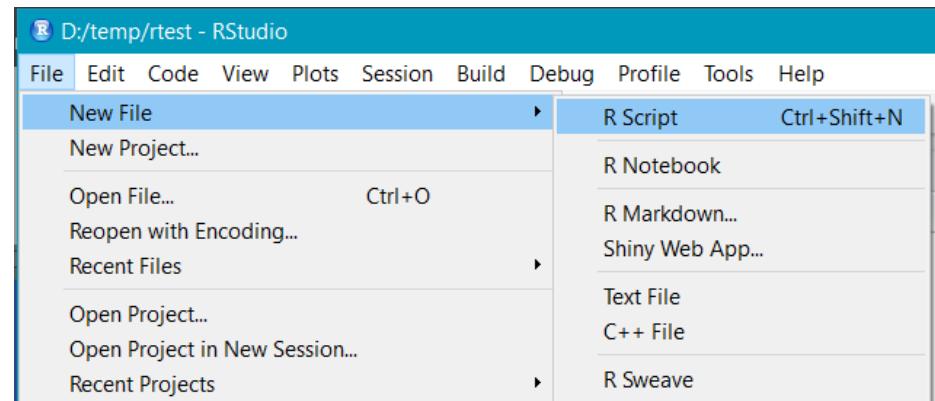


Click empty Project

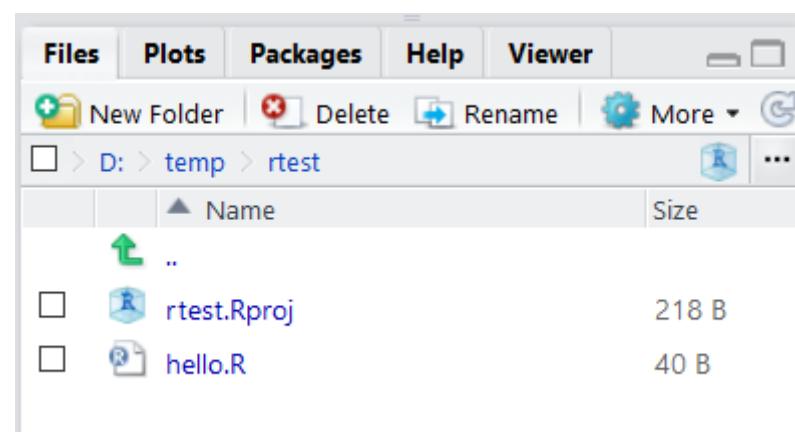


Hello world

Add R Script file to project

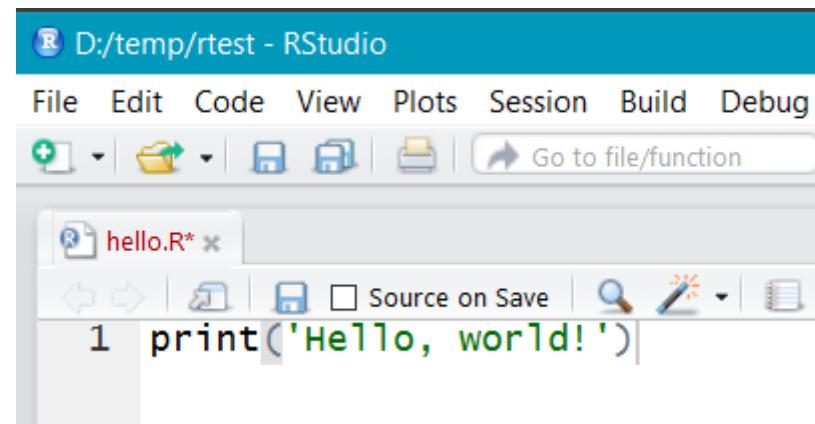


Save as hello.R

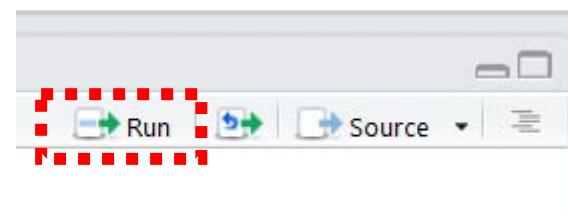


Hello world

Enter code



Click Run button to run Script



Basic calculation

The screenshot shows the RStudio interface with the following components:

- File Bar:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Toolbar:** Includes icons for file operations like Open, Save, Print, and Run, along with Go to file/function and Addins dropdown.
- Script Editor:** hello.R x, containing R code:

```
1 # Basic computations in R
2 2+3
3 6/3
4 (3*8)/(2*3)
5 log(12)
6 sqrt(121)
7 print('end')
```
- Console:** D:/temp/rtest/
- ```
> source('D:/temp/rtest/hello.R', echo=TRUE)

> # Basic computations in R
> 2+3
[1] 5

> 6/3
[1] 2

> (3*8)/(2*3)
[1] 4

> log(12)
[1] 2.484907

> sqrt(121)
[1] 11

> print('end')
[1] "end"
```
- Environment Tab:** Shows Global Environment with message: "Environment is empty".
- Files Tab:** Shows a project structure: D:/temp/rtest with files rtest.Rproj and hello.R.

## Variable assignment

The screenshot shows the RStudio interface with the following components:

- File Editor:** Displays the script file `hello.R` containing the following R code:

```
1 # Variable assignment
2 a <- 10
3 b <- 20
4 c <- a + b
5 c -> y
6 (y) + (a * b) -> m
```
- Console:** Shows the output of running the script:

```
> source('D:/temp/rtest/hello.R', echo=TRUE)
> # Variable assignment
> a <- 10
>
> b <- 20
>
> c <- a + b
>
> c -> y
>
> (y) + (a * b) -> m
> |
```
- Environment View:** Shows the global environment with the following variables:

| Name | Type    | Length | Size | Value |
|------|---------|--------|------|-------|
| a    | numeric | 1      | 48 B | 10    |
| b    | numeric | 1      | 48 B | 20    |
| c    | numeric | 1      | 48 B | 30    |
| m    | numeric | 1      | 48 B | 230   |
| y    | numeric | 1      | 48 B | 30    |
- Files View:** Shows the directory structure:

| Name        | Size  | Modified               |
|-------------|-------|------------------------|
| ..          |       |                        |
| rtest.Rproj | 218 B | May 31, 2017, 8:37 AM  |
| hello.R     | 78 B  | May 31, 2017, 10:36 AM |

## Basic data type

The screenshot shows the RStudio interface. The top window is the 'Script Editor' titled 'hello.R\*', containing the following R code:

```
1 # Basic data type
2 rm(list = ls()) # clear objects
3 name <- 'laploy' # character
4 who <- paste(name, 'v.') # string concat
5 price <- 1500 # numeric
6 x <- 1; y <- 2; z <- 3
7 # c() = combind function
8 v <- c(x,y,z) # double vector
9 bar <- 6:11 # integer vactor
10 e <- exp(1) # double form function
11
12 |
```

The bottom window is the 'Environment' browser, showing the global environment with the following variables:

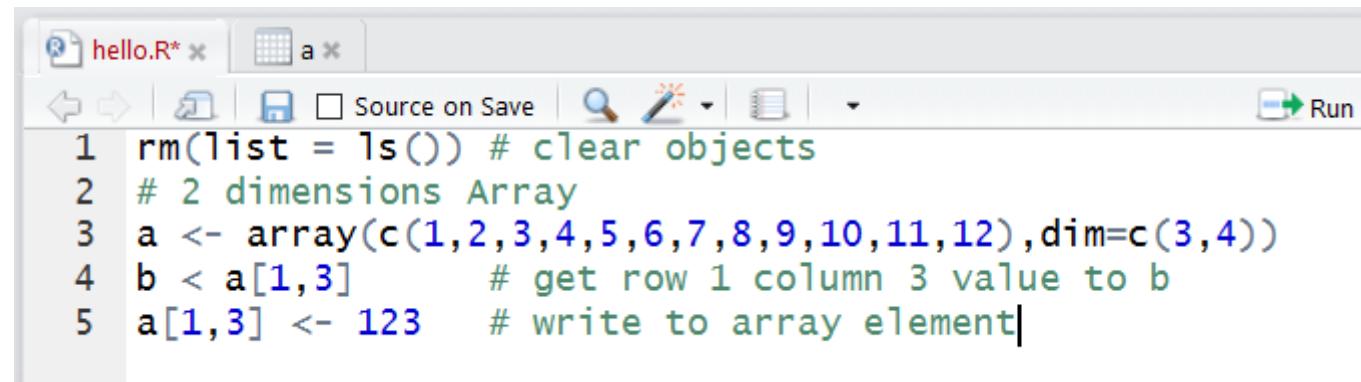
| Name  | Type      | Length | Size  | Value                    |
|-------|-----------|--------|-------|--------------------------|
| bar   | integer   | 6      | 72 B  | int [1:6] 6 7 8 9 10 ... |
| e     | numeric   | 1      | 48 B  | 2.71828182845905         |
| name  | character | 1      | 96 B  | "laploy"                 |
| price | numeric   | 1      | 48 B  | 1500                     |
| v     | numeric   | 3      | 72 B  | num [1:3] 1 2 3          |
| who   | character | 1      | 104 B | "laploy v."              |
| x     | numeric   | 1      | 48 B  | 1                        |
| y     | numeric   | 1      | 48 B  | 2                        |
| z     | numeric   | 1      | 48 B  | 3                        |

## Basic Operator

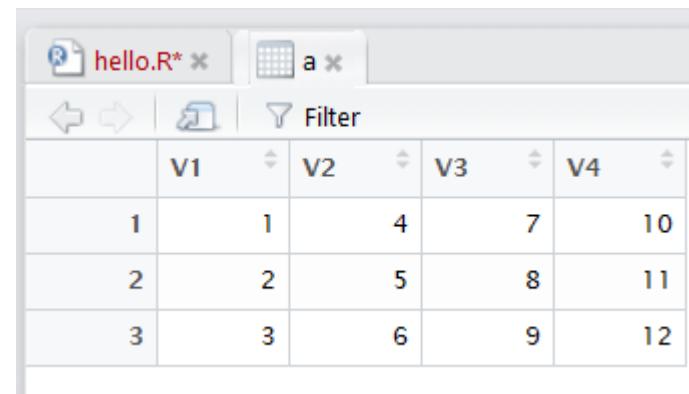
```
1 rm(list = ls()) # clear objects
2 # Arithmetic Operators
3 v <- c(1,2,3)
4 t <- c(2,2,1)
5 a <- v + t # Add two vectors
6 s <- v - t # subtracts
7 m <- v * t # multiply
8 d <- v / t # divide
9 r <- v %% t # remainder
10 e <- v ^ t # exponent
11 # Relation operators
12 g <- v > t # is greater?
13 b <- v < t # is less?
14 b <- v == t # is equal?
15 f <- v != t # is NOT equal?
```

## Data Structure

### Array



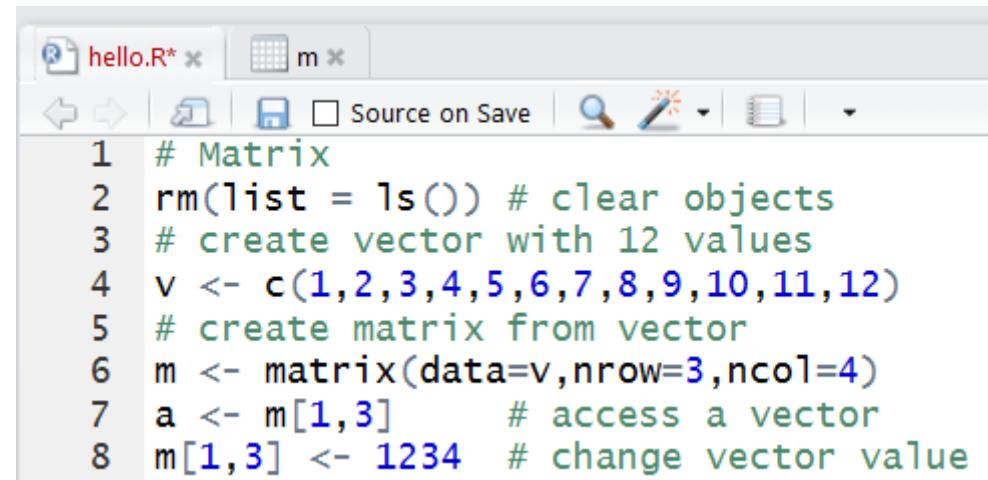
```
1 rm(list = ls()) # clear objects
2 # 2 dimensions Array
3 a <- array(c(1,2,3,4,5,6,7,8,9,10,11,12),dim=c(3,4))
4 b <- a[1,3] # get row 1 column 3 value to b
5 a[1,3] <- 123 # write to array element|
```



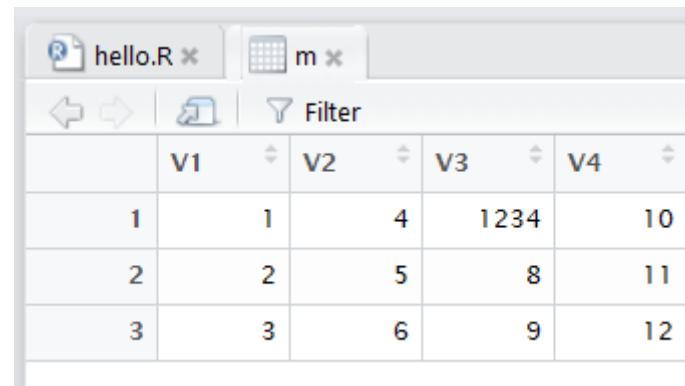
|   | V1 | V2 | V3 | V4 |
|---|----|----|----|----|
| 1 | 1  | 4  | 7  | 10 |
| 2 | 2  | 5  | 8  | 11 |
| 3 | 3  | 6  | 9  | 12 |

## Data Structure

### Matrix



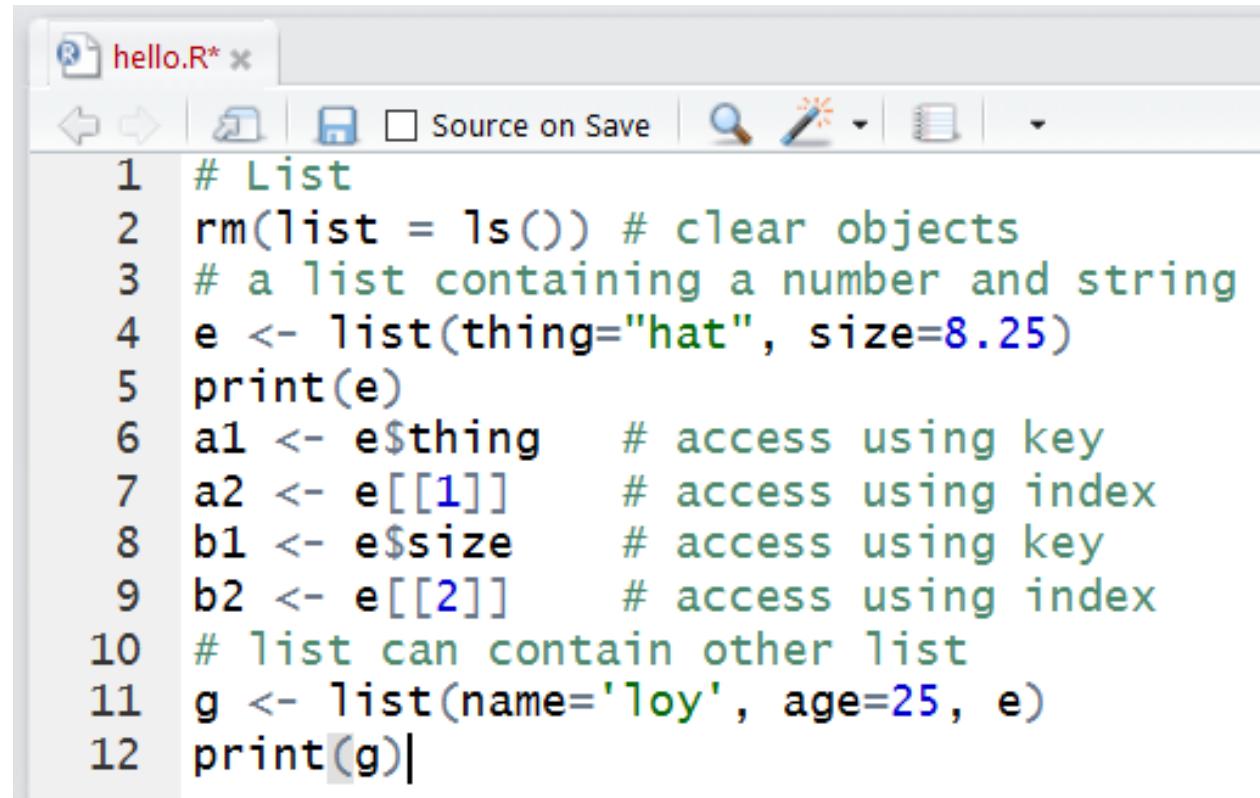
```
1 # Matrix
2 rm(list = ls()) # clear objects
3 # create vector with 12 values
4 v <- c(1,2,3,4,5,6,7,8,9,10,11,12)
5 # create matrix from vector
6 m <- matrix(data=v,nrow=3,ncol=4)
7 a <- m[1,3] # access a vector
8 m[1,3] <- 1234 # change vector value
```



|   | v1 | v2 | v3   | v4 |
|---|----|----|------|----|
| 1 | 1  | 4  | 1234 | 10 |
| 2 | 2  | 5  | 8    | 11 |
| 3 | 3  | 6  | 9    | 12 |

## Data Structure

### List



```
hello.R* x
1 # List
2 rm(list = ls()) # clear objects
3 # a list containing a number and string
4 e <- list(thing="hat", size=8.25)
5 print(e)
6 a1 <- e$thing # access using key
7 a2 <- e[[1]] # access using index
8 b1 <- e$size # access using key
9 b2 <- e[[2]] # access using index
10 # list can contain other list
11 g <- list(name='Loy', age=25, e)
12 print(g)|
```

## Data Structure

### Data Frame

|   | name  | age | gender |
|---|-------|-----|--------|
| 1 | Loy   | 19  | M      |
| 2 | Jim   | 17  | M      |
| 3 | Bo    | 22  | F      |
| 4 | Alice | 12  | F      |
| 5 | Tan   | 24  | M      |

---

```
1 # Data frame
2 rm(list = ls()) # clear objects
3 # create name vector variable
4 name <- c('Loy', 'Jim', 'Bo', 'Alice', 'Tan')
5 # create age vector variable
6 age <- c(19, 17, 22, 12, 24)
7 # create gender vector variable
8 gender <- c('M', 'M', 'F', 'F', 'M')
9 # create data frame from vector
10 student <- data.frame(name,age,gender)
11 student$gender == 'F' # look for female student
12 student$age > 20 # look for student older than 20
```

## If Statement

```
1 # If Statement
2 rm(list = ls()) # clear objects
3 x <- 1
4 if (x == 1){
5 print('same')
6 } else if (x > 1){
7 print('bigger')
8 } else {
9 print('smaller')
10 }
11 # ifelse function
12 a = c(5,7,2,9)
13 ifelse(a %% 2 == 0,"even","odd")
```

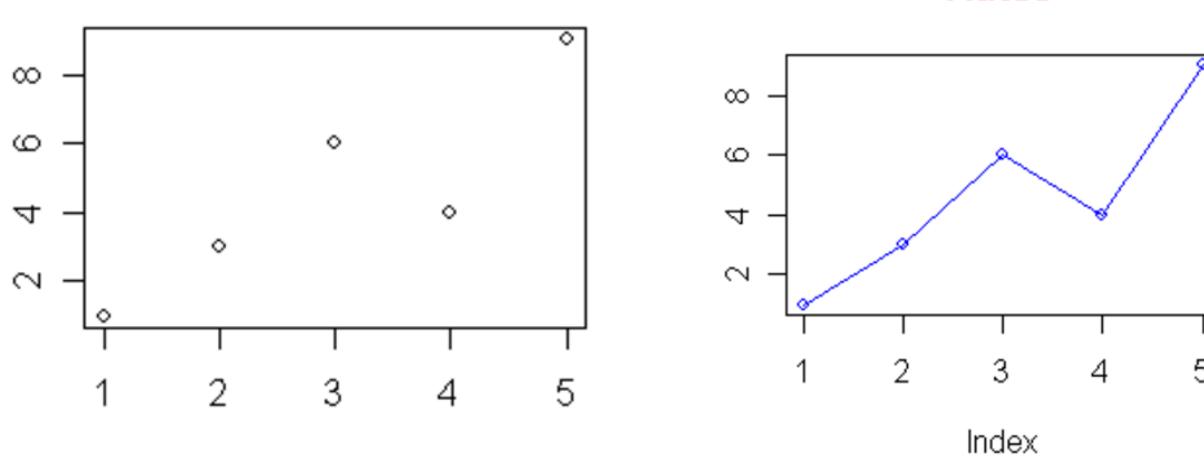
```
Console D:/temp/rtest/
> source('D:/temp/rtest/hello.R')
[1] "same"
>
>
> ifelse(a %% 2 == 0,"even","odd")
[1] "odd" "odd" "even" "odd"
> |
```

## For Loop

```
1 # For Loop
2 x <- c(2,5,3,9,8,11,6)
3 # iterate through elements
4 for (v in x) {
5 print(v)
6 }
7 # count even element
8 count <- 0
9 for (val in x) {
10 if(val %% 2 == 0) count = count+1
11 }
12 print(count)
```

## Basic plotting

```
1 # Line Charts
2 # Define the cars vector with 5 values
3 cars <- c(1, 3, 6, 4, 9)
4 # Graph the cars vector with all defaults
5 plot(cars)
6 -----
7 # Graph cars using blue points overlayed by a line
8 plot(cars, type="o", col="blue")
9 # Create a title with a red, bold/italic font
10 title(main="Autos", col.main="red", font.main=4)
```



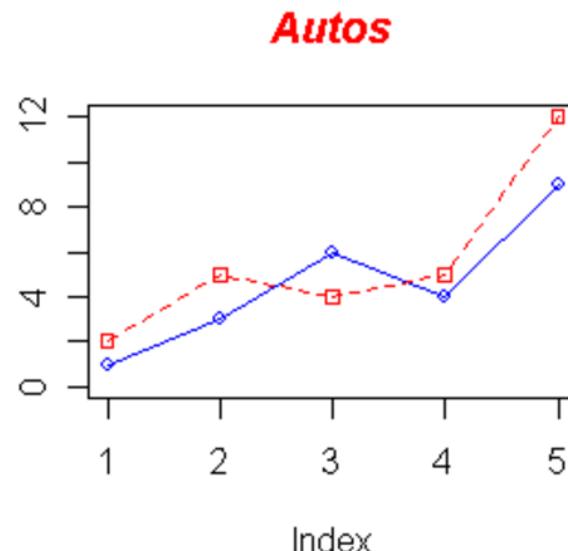
## Basic plotting

```
Define 2 vectors
cars <- c(1, 3, 6, 4, 9)
trucks <- c(2, 5, 4, 5, 12)

Graph cars using a y axis that ranges from 0 to 12
plot(cars, type="o", col="blue", ylim=c(0,12))

Graph trucks with red dashed line and square points
lines(trucks, type="o", pch=22, lty=2, col="red")

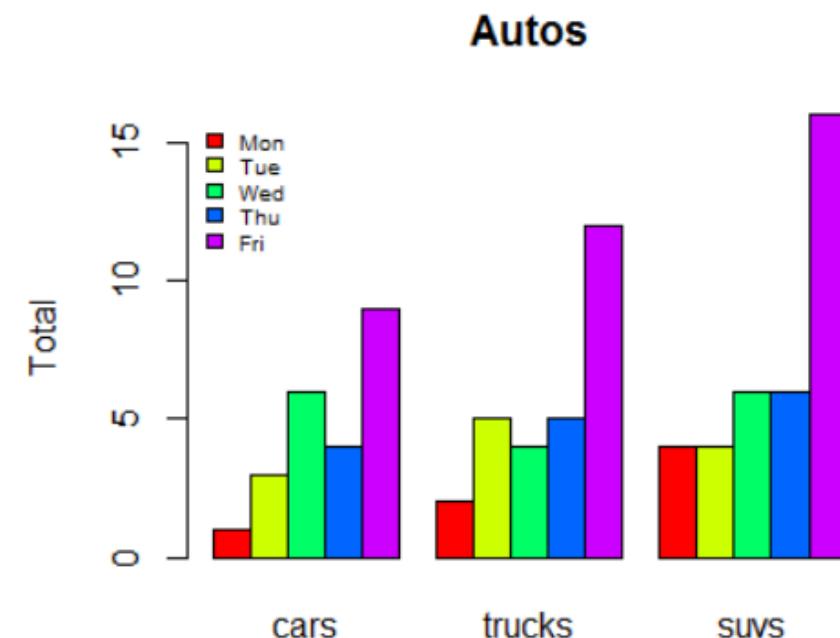
Create a title with a red, bold/italic font
title(main="Autos", col.main="red", font.main=4)
```



## Basic plotting

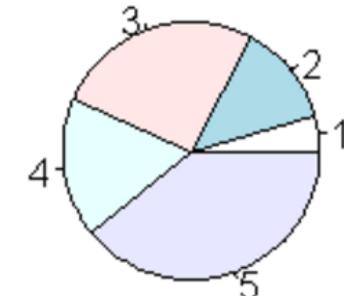
```
1 # Read values from tab-delimited autos.dat
2 autos_data <- read.table("d:/temp/autos.dat", header=T, sep="\t")
3 # Graph autos with adjacent bars using rainbow colors
4 barplot(as.matrix(autos_data), main="Autos", ylab= "Total", beside=TRUE, col=rainbow(5))
5 # Place the legend at the top-left corner with no frame
6 # using rainbow colors
7 legend("topleft", c("Mon", "Tue", "Wed", "Thu", "Fri"), cex=0.6, bty="n", fill=rainbow(5))
```

|   | cars | trucks | SUVS |
|---|------|--------|------|
| 1 | 1    | 2      | 4    |
| 2 | 3    | 5      | 4    |
| 3 | 6    | 4      | 6    |
| 4 | 4    | 5      | 6    |
| 5 | 9    | 12     | 16   |



## Basic plotting

```
1 # Define cars vector with 5 values
2 cars <- c(1, 3, 6, 4, 9)
3 # Create a pie chart for cars
4 pie(cars)
5 -----
6 pie(cars, main="Cars", col=rainbow(length(cars)),
7 labels=c("Mon", "Tue", "Wed", "Thu", "Fri"))
```



**Cars**



**Cars**

More information

## More information on R Script Basic

A Complete Tutorial to learn Data Science in R from Scratch

<https://www.analyticsvidhya.com/blog/2016/02/complete-tutorial-learn-data-science-scratch/>

# IMPORT DATASET



## In this session

- Import dataset from local CVS file to R data frame (R Studio)
- Add column name Using R code (R Studio)
- Import dataset from internet (R Studio)
- Import dataset from internet (Azure ML Studio)
- Add column name Using R code (Azure ML Studio)
- Data Visualization (Azure ML Studio)

## Import dataset from local CVS file to R data frame (R Studio)

adult100.csv

|    | A   | B                | C                     | D                 | E     | F      |
|----|-----|------------------|-----------------------|-------------------|-------|--------|
| 1  | age | workclass        | marital-status        | occupation        | race  | sex    |
| 2  | 39  | State-gov        | Never-married         | Adm-clerical      | White | Male   |
| 3  | 50  | Self-emp-not-inc | Married-civ-spouse    | Exec-managerial   | White | Male   |
| 4  | 38  | Private          | Divorced              | Handlers-cleaners | White | Male   |
| 5  | 53  | Private          | Married-civ-spouse    | Handlers-cleaners | Black | Male   |
| 6  | 28  | Private          | Married-civ-spouse    | Prof-specialty    | Black | Female |
| 7  | 37  | Private          | Married-civ-spouse    | Exec-managerial   | White | Female |
| 8  | 49  | Private          | Married-spouse-absent | Other-service     | Black | Female |
| 9  | 52  | Self-emp-not-inc | Married-civ-spouse    | Exec-managerial   | White | Male   |
| 10 | 31  | Private          | Never-married         | Prof-specialty    | White | Female |
| 11 | 42  | Private          | Married-civ-spouse    | Exec-managerial   | White | Male   |

## Import dataset from local CVS file to R data frame (R Studio)

```
getwd() # get working directory
setwd("c:/temp") # set working directory
list.files() # list file in current directory
d1 <- read.csv("adult100.csv", header = FALSE)
str(d1) # show structure of d1

'data.frame': 99 obs. of 6 variables:
 $ age : int 39 50 38 53 28 37 49 52 31 42 ...
 $ workclass : Factor w/ 7 levels "?","Federal-gov",...: 7 6 4 4 4 4 4 6 4 4 ...
 $ marital-status: Factor w/ 6 levels "Divorced","Married-AF-spouse",...: 5 3 1 3 3 3 4 3 5 3 ...
 $ occupation : Factor w/ 13 levels "?","Adm-clerical",...: 2 4 6 6 9 4 8 4 9 4 ...
 $ race : Factor w/ 5 levels "Amer-Indian-Eskimo",...: 5 5 5 3 3 5 3 5 5 5 ...
 $ sex : Factor w/ 2 levels "Female","Male": 2 2 2 2 1 1 2 1 2 1 ...

typeof(d1) # show data type of d1
class(d1) # show class of d1
dim(d1) # show dimension of d1
```

## Import dataset from local CVS file to R data frame (R Studio)

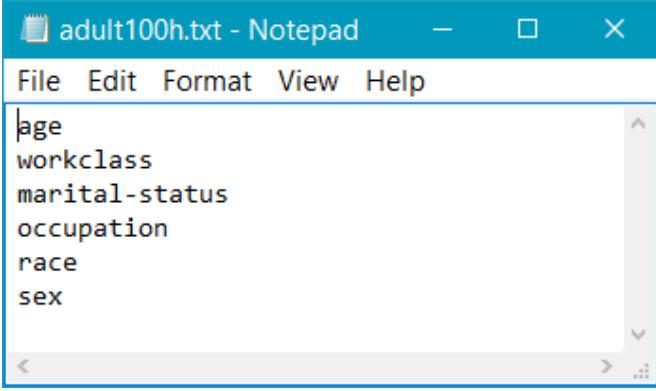
|                       |                                         |
|-----------------------|-----------------------------------------|
| ncol(d1)              | # number of columns                     |
| nrow(d1)              | # number of rows                        |
| d1                    | # show all data rows                    |
| head(d1)              | # preview data in d1                    |
| head(d1, n = 10)      | # show top 10 rows                      |
| tail(d1)              | # preview data in d1                    |
| tail(d1, n = 10)      | # show bottom 10 rows                   |
| d1[1:3]               | # show column 1 to column 3             |
| head(d1[1:3], n = 10) | # show top 10 rows column 1 to column 3 |

## Add column name Using R code (R Studio)

Adult100n.csv

|   | A  | B                | C                  | D                 | E     | F      |
|---|----|------------------|--------------------|-------------------|-------|--------|
| 1 | 39 | State-gov        | Never-married      | Adm-clerical      | White | Male   |
| 2 | 50 | Self-emp-not-inc | Married-civ-spouse | Exec-managerial   | White | Male   |
| 3 | 38 | Private          | Divorced           | Handlers-cleaners | White | Male   |
| 4 | 53 | Private          | Married-civ-spouse | Handlers-cleaners | Black | Male   |
| 5 | 28 | Private          | Married-civ-spouse | Prof-specialty    | Black | Female |
| 6 | 37 | Private          | Married-civ-spouse | Exec-managerial   | White | Female |
| 7 | 49 | Private          | Married-civ-spouse | Other-service     | Black | Female |

Adult100h.txt



The screenshot shows a Windows Notepad window titled "adult100h.txt - Notepad". The window contains the following text:

```
age
workclass
marital-status
occupation
race
sex
```

## Add column name Using R code (R Studio)

```
ls() # print all object in workspace
rm(list=ls()) # Clear R workspace
d1 <- read.csv("adult100n.csv", header = FALSE) # import dataset without column
typeof, structure, class, preview d1
d2 <- readLines("adult100h.txt") # import column name
typeof, structure, class, preview d2
colnames(d1) <- d2 # update d1 column names
preview d1
```

## Import dataset from internet (R Studio)

UCI Machine Learning Repository: Adult Data Set

home page

<https://archive.ics.uci.edu/ml/datasets/adult>

Description

<https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.names>

Data Set

<http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data>

Import dataset from internet (R Studio)



[About](#) [Citation Policy](#) [Donate a Data Set](#) [Contact](#)

Search

Repository  Web 

[View ALL Data Sets](#)

## Machine Learning Repository

Center for Machine Learning and Intelligent Systems

## Adult Data Set

Download: [Data Folder](#), [Data Set Description](#)

**Abstract:** Predict whether income exceeds \$50K/yr based on census data. Also known as "Census Income" dataset.



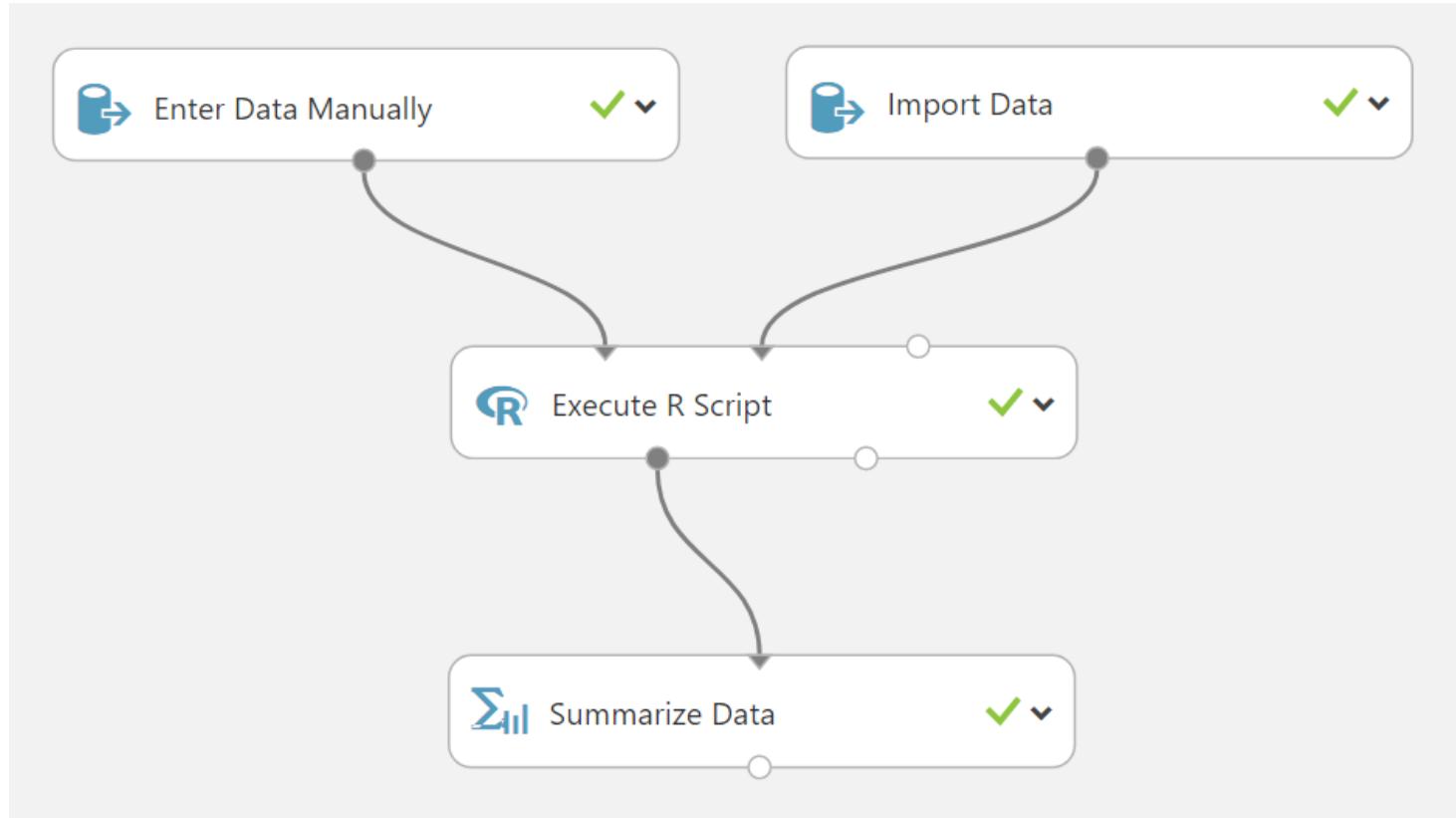
|                                   |                      |                              |       |                            |            |
|-----------------------------------|----------------------|------------------------------|-------|----------------------------|------------|
| <b>Data Set Characteristics:</b>  | Multivariate         | <b>Number of Instances:</b>  | 48842 | <b>Area:</b>               | Social     |
| <b>Attribute Characteristics:</b> | Categorical, Integer | <b>Number of Attributes:</b> | 14    | <b>Date Donated</b>        | 1996-05-01 |
| <b>Associated Tasks:</b>          | Classification       | <b>Missing Values?</b>       | Yes   | <b>Number of Web Hits:</b> | 899430     |

## Import dataset from internet (R Studio)

```
rm(list=ls()) # Clear R workspace
u <- "http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data"
d1 <- read.csv(url(u), header = FALSE) # import dataset
View(d1) # invoke spreadsheet-style data viewer on a matrix-like R object
d2 <- readLines("cencol.txt") # import column name
colnames(d1) <- d2 # update d1 column names
```

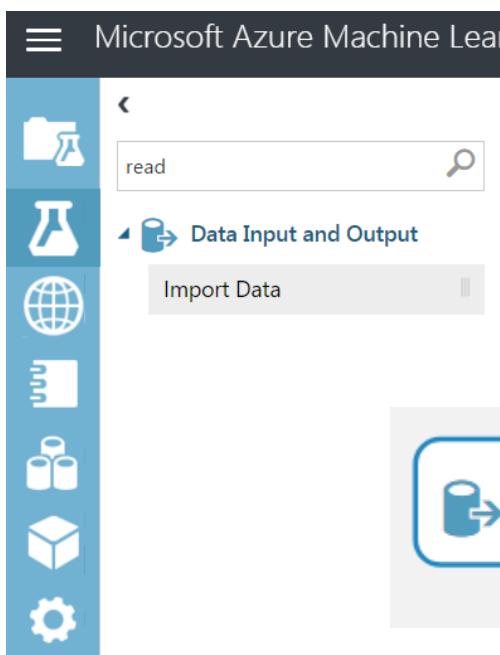
## Import dataset from internet (Azure ML Studio)

This experiment is completed



## Import dataset from internet (Azure ML Studio)

- Open Microsoft Azure Machine Learning Studio
- Create New blank experiment name = R add col name
- Click Data Input and Output
- Drag & drop Import Data
- Set properties



<http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data>

Properties Project

### Import Data

Launch Import Data Wizard

Data source

Web URL via HTTP

Data source URL

http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data

Data format

CSV

CSV or TSV has header row

Use cached results

START TIME 5/20/2017 9:13:05 PM

END TIME 5/20/2017 9:13:16 PM

ELAPSED TIME 0:00:11.502

STATUS CODE Finished

STATUS DETAILS None

A detailed view of the "Import Data" step properties. It shows the selected "Web URL via HTTP" as the data source, with the URL "http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data" entered. The "Data format" is set to "CSV". There are two unchecked checkboxes: "CSV or TSV has header row" and "Use cached results". The step's start time is 5/20/2017 9:13:05 PM, end time is 5/20/2017 9:13:16 PM, and elapsed time is 0:00:11.502. The status is "Finished" with no details provided.

## Import dataset from internet (Azure ML Studio)

### Right-click result dataset (Dataset) at Import Data Visualize

| rows  | columns          | Col1    | Col2      | Col3      | Col4                  | Col5              | Col6          | Col7      | Col8      | Col9      |
|-------|------------------|---------|-----------|-----------|-----------------------|-------------------|---------------|-----------|-----------|-----------|
| 32562 | 15               |         |           |           |                       |                   |               |           |           |           |
|       |                  | view as | histogram | histogram | histogram             | histogram         | histogram     | histogram | histogram | histogram |
| 39    | State-gov        | 77516   | Bachelors | 13        | Never-married         | Adm-clerical      | Not-in-family | White     |           |           |
| 50    | Self-emp-not-inc | 83311   | Bachelors | 13        | Married-civ-spouse    | Exec-managerial   | Husband       | White     |           |           |
| 38    | Private          | 215646  | HS-grad   | 9         | Divorced              | Handlers-cleaners | Not-in-family | White     |           |           |
| 53    | Private          | 234721  | 11th      | 7         | Married-civ-spouse    | Handlers-cleaners | Husband       | Black     |           |           |
| 28    | Private          | 338409  | Bachelors | 13        | Married-civ-spouse    | Prof-specialty    | Wife          | Black     |           |           |
| 37    | Private          | 284582  | Masters   | 14        | Married-civ-spouse    | Exec-managerial   | Wife          | White     |           |           |
| 49    | Private          | 160187  | 9th       | 5         | Married-spouse-absent | Other-service     | Not-in-family | Black     |           |           |

## Add column name Using R instructions (Azure ML Studio)

- Drag & drop Enter Data Manually from Data Input and Output to canvas
- Set properties

The screenshot shows the Azure ML Studio interface. On the left, there's a sidebar with icons for Data, Data Input and Output, and Model. The 'Data Input and Output' section is expanded, showing the 'Enter Data Manually' component. A blue callout box highlights this component with the number '1'. The main workspace shows the 'Enter Data Manually' component on the canvas. To the right, the 'Properties' pane is open, showing the 'Enter Data Manually' settings under the 'Data' tab. The 'DataFormat' is set to 'CSV' and 'HasHeader' is checked. The 'Data' table lists the columns: 1 column\_name, 2 age, 3 workclass, 4 fnlwgt, 5 education, 6 education-num. Below the component, execution details are shown: START TIME 5/20/2017 9:13:02 PM, END TIME 5/20/2017 9:13:04 PM, ELAPSED TIME 0:00:02.453, STATUS CODE Finished, STATUS DETAILS None. To the far right, a vertical list of column names is displayed, corresponding to the numbers in the 'Data' table: 1 column\_name, 2 age, 3 workclass, 4 fnlwgt, 5 education, 6 education-num, 7 marital-status, 8 occupation, 9 relationship, 10 race, 11 sex, 12 capital-gain, 13 capital-loss, 14 hours-per-week, 15 native-country, 16 income.

| Column Number | Column Name    |
|---------------|----------------|
| 1             | column_name    |
| 2             | age            |
| 3             | workclass      |
| 4             | fnlwgt         |
| 5             | education      |
| 6             | education-num  |
| 7             | marital-status |
| 8             | occupation     |
| 9             | relationship   |
| 10            | race           |
| 11            | sex            |
| 12            | capital-gain   |
| 13            | capital-loss   |
| 14            | hours-per-week |
| 15            | native-country |
| 16            | income         |

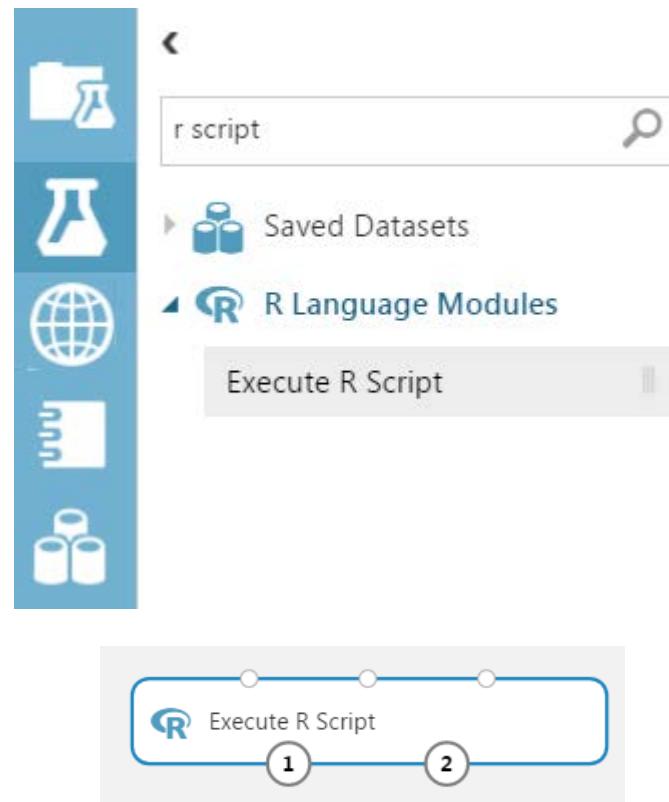
## Add column name Using R instructions (Azure ML Studio)

### Right-click at Enter Data manually dataset (Dataset) Visualize



## Add column name Using R instructions (Azure ML Studio)

- Add Execute R Script module
- Configure R Script module
- Enter R Script



Properties Project

### Execute R Script

#### R Script

```
1 # Map 1-based optional input port
2 dataset1 <- maml.mapInputPort(1)
3 dataset2 <- maml.mapInputPort(2)
4
5 # Contents of optional Zip port
6 # source("src/yourfile.R");
```

#### Random Seed

#### R Version

CRAN R 3.1.0

START TIME 5/20/2017 9:13:20 PM

END TIME 5/20/2017 9:13:33 PM

ELAPSED TIME 0:00:13.093

STATUS CODE Finished

STATUS DETAILS None

## Add column name Using R instructions (Azure ML Studio)

```
Map 1-based optional input ports to variables
dataset1 <- maml.mapInputPort(1) # class: data.frame
dataset2 <- maml.mapInputPort(2) # class: data.frame

Sample operation
colnames(dataset2) <- c(dataset1['column_name'])$column_name;
data.set = dataset2;

Select data.frame to be sent to the output Dataset port
maml.mapOutputPort("data.set");
```

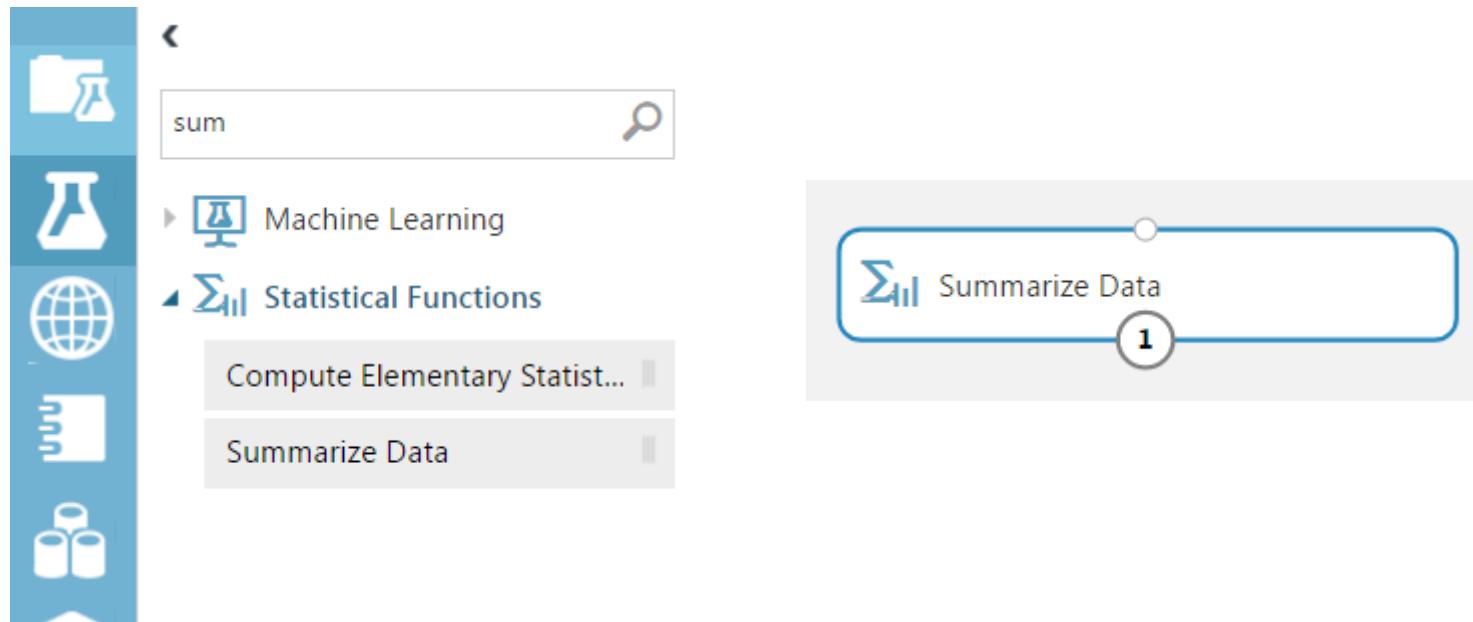
## Add column name Using R instructions (Azure ML Studio)

### Visualize at the output of Execute R Script module

| age | workclass        | fnlwgt | education | education-num | marital-status     | occupation        | relationship  | race  | sex    | capital-gain | ca | lo |
|-----|------------------|--------|-----------|---------------|--------------------|-------------------|---------------|-------|--------|--------------|----|----|
| 39  | State-gov        | 77516  | Bachelors | 13            | Never-married      | Adm-clerical      | Not-in-family | White | Male   | 2174         | 0  |    |
| 50  | Self-emp-not-inc | 83311  | Bachelors | 13            | Married-civ-spouse | Exec-managerial   | Husband       | White | Male   | 0            | 0  |    |
| 38  | Private          | 215646 | HS-grad   | 9             | Divorced           | Handlers-cleaners | Not-in-family | White | Male   | 0            | 0  |    |
| 53  | Private          | 234721 | 11th      | 7             | Married-civ-spouse | Handlers-cleaners | Husband       | Black | Male   | 0            | 0  |    |
| 28  | Private          | 338409 | Bachelors | 13            | Married-civ-spouse | Prof-specialty    | Wife          | Black | Female | 0            | 0  |    |
| 37  | Private          | 284582 | Masters   | 14            | Married-civ-spouse | Exec-managerial   | Wife          | White | Female | 0            | 0  |    |

## Data Visualization (Azure ML Studio)

- Add Summarize data module
- Link to Execute R Script module



## Data Visualization (Azure ML Studio)

### Summarize dataset visualization

| Feature        | Count | Unique Value Count | Missing Value Count | Min   | Max     | Mean          | Mean Deviation | 1st Quartile | Median | 3rd Quartile |
|----------------|-------|--------------------|---------------------|-------|---------|---------------|----------------|--------------|--------|--------------|
| age            | 32561 | 73                 | 1                   | 17    | 90      | 38.581647     | 11.189182      | 28           | 37     | 48           |
| workclass      | 30725 | 9                  | 1837                |       |         |               |                |              |        |              |
| fnlwgt         | 32561 | 21648              | 1                   | 12285 | 1484705 | 189778.366512 | 77608.21854    | 117827       | 178356 | 237051       |
| education      | 32561 | 17                 | 1                   |       |         |               |                |              |        |              |
| education-num  | 32561 | 16                 | 1                   | 1     | 16      | 10.080679     | 1.903048       | 9            | 10     | 12           |
| marital-status | 32561 | 8                  | 1                   |       |         |               |                |              |        |              |
| occupation     | 30718 | 15                 | 1844                |       |         |               |                |              |        |              |
| relationship   | 32561 | 7                  | 1                   |       |         |               |                |              |        |              |
| race           | 32561 | 6                  | 1                   |       |         |               |                |              |        |              |
| sex            | 32561 | 3                  | 1                   |       |         |               |                |              |        |              |

## More Information

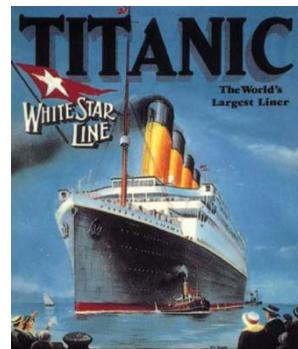
Import your training data into Azure Machine Learning Studio from various data sources

<https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-data-science-import-data>

This experiment

<https://gallery.cortanaintelligence.com/Experiment/R-add-col-name>

# R SCRIPT FEATURE ENGINEERING



## In this session

- What is the Feature?
- What is Feature Engineering?
- The process of Feature Engineering
- Where is FE in ML?
- Preparing for experiment
- Adding family size feature
- Adding Age\*Class and Fare per person feature
- Adding Deck feature
- Adding Title feature

## What is the Feature?

### What is the Feature?

- A piece of information
- Might be useful for prediction
- Any useful attribute to the model
- Is measurable property
- Feature is input; label is output.
- Is one column of the data

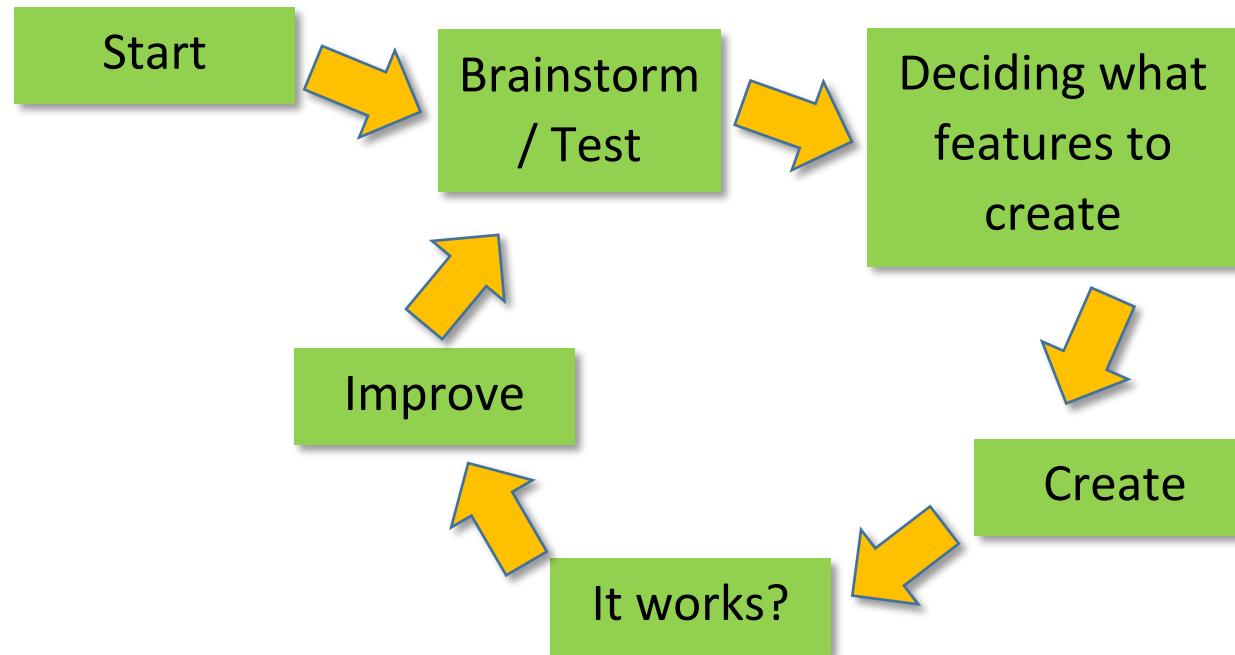
## What is Feature Engineering?

# What is Feature Engineering?

- Is the method if find X for input
- Is “Data Science”
- Is difficult
- Is expensive
- Is time-consuming
- Is require expert knowledge in domain
- Is applied machine learning

## The process of feature engineering

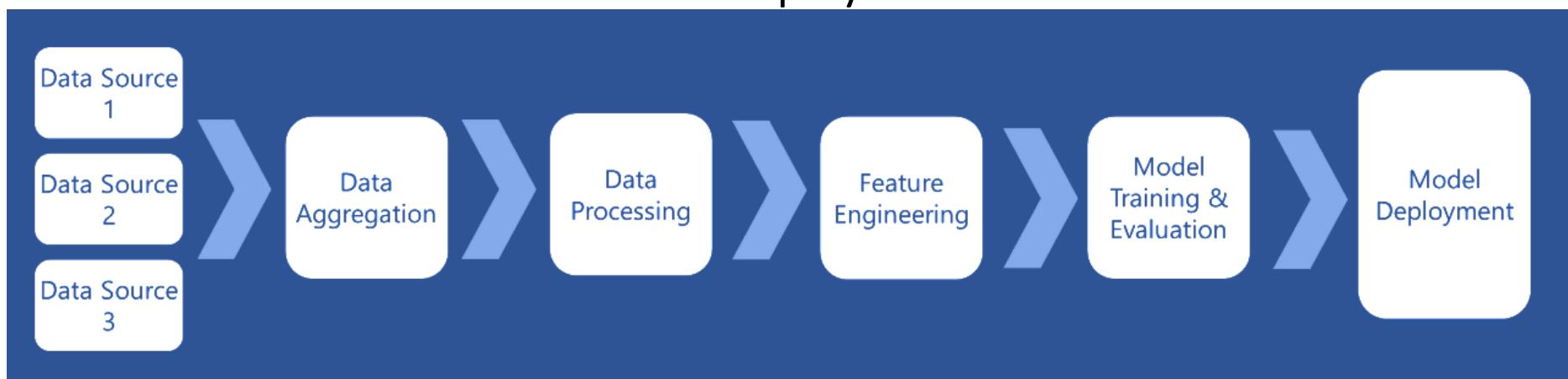
# The process of feature engineering



Where is FE in ML?

## Where is FE in ML?

- Data sources
- Data aggregation
- Data Processing
- Feature Engineering
- Model Training & Evaluation
- Model Deployment



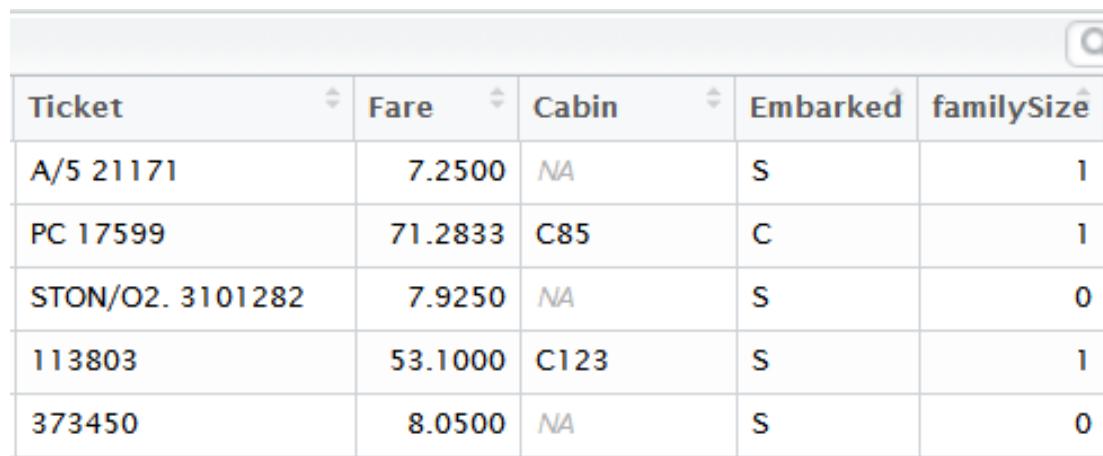
## Preparing for experiment

# Preparing for experiment

7. Go to <https://github.com/laploy/ML>
8. Right click TitanicData.csv and save link as to c:\temp
9. Open R Studio
10. Create New project name = c:\temp\rfe
11. Right click project / click Add... / New R Script file

Add family size feature  
File name = 100 familySize

```
7 rm(list = ls()) # clear work space
8 setwd("d:/temp") # set current work directory
9 # import data file
10 dat <- read.csv("TitanicData.csv", na.strings = "")
11
12 # ----- Main program -----
13 # Create Family Size feature
14 dat$familySize <- dat$SibSp + dat$Parch
```



The screenshot shows a data visualization interface with a search icon at the top right. Below is a table with the following data:

| Ticket           | Fare    | Cabin | Embarked | familySize |
|------------------|---------|-------|----------|------------|
| A/5 21171        | 7.2500  | NA    | S        | 1          |
| PC 17599         | 71.2833 | C85   | C        | 1          |
| STON/O2. 3101282 | 7.9250  | NA    | S        | 0          |
| 113803           | 53.1000 | C123  | S        | 1          |
| 373450           | 8.0500  | NA    | S        | 0          |

## Add Age\*Class and Fare per person feature

File name = 101 ageClass

```
7 rm(list = ls()) # clear work space
8 setwd("d:/temp") # set current work directory
9 # import data file
10 dat <- read.csv("TitanicData.csv", na.strings = "")
11
12 # ----- Main program -----
13 # 1. Create ageClass feature
14 dat$ageClass <- dat$Age * dat$Pclass
15 # 2. Create Family Size feature
16 dat$familySize <- dat$SibSp + dat$Parch
17 # 3. Create fare per person
18 dat$FarePerPerson <- dat$Fare / dat$familySize
```

| Fare    | Cabin | Embarked | ageClass | familySize | FarePerPerson |
|---------|-------|----------|----------|------------|---------------|
| 7.2500  | NA    | S        | 66.00    | 1          | 7.250000      |
| 71.2833 | C85   | C        | 38.00    | 1          | 71.283300     |
| 7.9250  | NA    | S        | 78.00    | 0          | Inf           |
| 53.1000 | C123  | S        | 35.00    | 1          | 53.100000     |
| 8.0500  | NA    | S        | 105.00   | 0          | Inf           |
| 8.4583  | NA    | Q        | NA       | 0          | Inf           |
| 51.8625 | E46   | S        | 54.00    | 0          | Inf           |
| 21.0750 | NA    | S        | 6.00     | 4          | 5.268750      |
| 11.1333 | NA    | S        | 81.00    | 2          | 5.566650      |

## Add Deck feature File name = 102 addDeck

```
7 rm(list = ls()) # clear work space
8 setwd("d:/temp") # set current work directory
9 # import data file
10 dat <- read.csv("TitanicData.csv", na.strings = "")
11 # create Deck feature
12 dat$Deck <- (substr(dat$Cabin,0,1))
13 print('end')
```

| Parch | Ticket           | Fare    | Cabin | Embarked | Deck |
|-------|------------------|---------|-------|----------|------|
| 0     | A/5 21171        | 7.2500  | NA    | S        | NA   |
| 0     | PC 17599         | 71.2833 | C85   | C        | C    |
| 0     | STON/O2. 3101282 | 7.9250  | NA    | S        | NA   |
| 0     | 113803           | 53.1000 | C123  | S        | C    |
| 0     | 373450           | 8.0500  | NA    | S        | NA   |
| 0     | 330877           | 8.4583  | NA    | Q        | NA   |
| 0     | 17463            | 51.8625 | E46   | S        | E    |
| 1     | 349909           | 21.0750 | NA    | S        | NA   |

## Adding Title feature

File name = 103 addTitle

```
7 require(magrittr)
8 require(purrr)
9 rm(list = ls()) # clear work space
10 setwd("d:/temp") # set current work directory
11 # import data file
12 dat <- read.csv("TitanicData.csv", na.strings = "")
13
14 titleList = c('Mrs', 'Mr', 'Master', 'Miss', 'Major', 'Rev',
15 'Dr', 'Ms', 'Mlle', 'Col', 'Capt', 'Mme', 'Countess',
16 'Don', 'Jonkheer')
17
18 - getTitle <- function(name){
19 for(s in titleList)
20 if(regexexpr(pattern=s, name) != -1)
21 return(s)
22 return(NA)
23 }
```

## Adding Title feature

```
25 - replaceTitles <- function(x){
26 title = x['Title']
27 sex = x['Sex']
28 s = sex$Sex
29 t = title>Title
30 if(any(t == c('Don', 'Major', 'Capt', 'Jonkheer', 'Rev', 'Col')))
31 return('Mr')
32 if(any(t == c('Countess', 'Mme')))
33 return('Mrs')
34 if(any(t == c('Mlle', 'Ms')))
35 return('Miss')
36 if(t == 'Dr')
37 if(s == 'male')
38 return('Mr')
39 if(s == 'female')
40 return('Mrs')
41 return(t)
42 }
```

## Adding Title feature

```
44 # ----- Main program -----
45 # Extract title from column Name and create Title column
46 dat>Title <- dat %>% .$Name %>% map(~ getTitle(.x))
47 # Replacing all titles with mr, mrs, miss, master
48 dat$x <- apply(dat[c('Title','Sex')], 1, replaceTitles)
49 print('end')
```

| Sex    | Age   | SibSp | Parch | Ticket            | Fare    | Cabin | Embarked | Title | x   |
|--------|-------|-------|-------|-------------------|---------|-------|----------|-------|-----|
| female |       |       |       |                   |         |       |          |       |     |
| male   | 29.00 | 0     | 0     | W./C. 14263       | 10.5000 | NA    | S        | Mr    | Mr  |
| male   | 22.00 | 0     | 0     | STON/O 2. 3101275 | 7.1250  | NA    | S        | Mr    | Mr  |
| male   | 30.00 | 0     | 0     | 2694              | 7.2250  | NA    | C        | Mr    | Mr  |
| male   | 44.00 | 2     | 0     | 19928             | 90.0000 | C78   | Q        | Dr    | Mr  |
| female | 25.00 | 0     | 0     | 347071            | 7.7750  | NA    | S        | Miss  | Mrs |
| female | 24.00 | 0     | 2     | 250649            | 14.5000 | NA    | S        | Mrs   | Mrs |
| male   | 37.00 | 1     | 1     | 11751             | 52.5542 | D35   | S        | Mr    | Mr  |
| male   | 54.00 | 1     | 0     | 244252            | 26.0000 | NA    | S        | Rev   | Mr  |
| male   | NA    | 0     | 0     | 362316            | 7.2500  | NA    | S        | Mr    | Mr  |
| female | 29.00 | 1     | 1     | 347054            | 10.4625 | G6    | S        | Mrs   | Mrs |

More information

## Feature engineering in data science

<https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-data-science-create-features>

## Source code

<https://github.com/laploy/rfe>

# MISSING VALUE HANDLING IN R

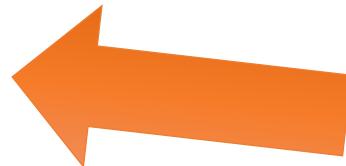


## In this session

27. Replace missing values with the mean
28. Replace missing values with the median
29. Replace missing values with an interpolated estimate
30. Replace missing values with a constant
31. Replace missing values using imputation
32. Replace missing values with a missing rank
33. Replace missing values with a dummy
34. Replace missing values with 0
35. Create an indicator variable for "missing."
36. Replace missing values with a string
37. Add an indicator variable showing which strings are considered "missing."
38. Delete columns that are missing too many values to be useful
39. Delete rows that are missing critical values

We need data that is:

- Relevant
- Connected
- Accurate
- Enough to work with



## Example of missing values dataset

|         | Column 0 | age | years_seniority | income | parking_space | attending_party | entree  | pets | emergency_contact |
|---------|----------|-----|-----------------|--------|---------------|-----------------|---------|------|-------------------|
| Tony    | 48       | 27  |                 | 86     | 1             | 5               | shrimp  |      | Pepper            |
| Donald  | 67       | 25  |                 | 95     | 10            | 2               | beef    |      | Jane              |
| Henry   | 69       | 21  |                 | 110    | 6             | 1               | chicken | 62   | Janet             |
| Janet   | 62       | 21  |                 | 63     | 3             | 1               | beef    |      | Henry             |
| Nick    |          | 17  |                 | 77     | 4             |                 | veggie  |      | NA                |
| Bruce   | 37       | 14  |                 | 118    |               | 1               | chicken |      | n/a               |
| Steve   | 83       |     |                 | 52     | 7             | 1               | shrimp  | 3    | None              |
| Clint   | 27       | 9   |                 | 162    | 9             |                 | shrimp  |      | empty             |
| Wanda   | 19       | 7   |                 | 127    | 2             | 2               | veggie  | 1    | -                 |
| Natasha | 26       | 4   |                 | 68     | 5             | 3               | chicken |      | *****             |
| Carol   |          | 3   |                 |        | 11            | 1               |         |      | null              |
| Mandy   | 44       | 2   |                 |        | 8             | 1               |         |      |                   |



Too many missing data == Swiss cheese

## Example of missing values dataset CSV file

missing\_values.csv

|    | A       | B   | C               | D      | E             | F               | G         | H    | I                 |
|----|---------|-----|-----------------|--------|---------------|-----------------|-----------|------|-------------------|
| 1  |         | age | years_seniority | income | parking_space | attending_party | entree    | pets | emergency_contact |
| 2  | Tony    | 48  | 27              |        | 1             |                 | 5 shrimp  |      | Pepper            |
| 3  | Donald  | 67  | 25              | 86     | 10            |                 | 2 beef    |      | Jane              |
| 4  | Henry   | 69  | 21              | 95     | 6             |                 | 1 chicken | 62   | Janet             |
| 5  | Janet   | 62  | 21              | 110    | 3             |                 | 1 beef    |      | Henry             |
| 6  | Nick    |     | 17              |        | 4             |                 |           |      |                   |
| 7  | Bruce   | 37  | 14              | 63     |               |                 | 1 veggie  |      | NA                |
| 8  | Steve   | 83  |                 | 77     | 7             |                 | 1 chicken |      | n/a               |
| 9  | Clint   | 27  | 9               | 118    | 9             |                 | shrimp    | 3    | None              |
| 10 | Wanda   | 19  | 7               | 52     | 2             |                 | 2 shrimp  |      | empty             |
| 11 | Natasha | 26  | 4               | 162    | 5             |                 | 3         |      | -                 |
| 12 | Carol   |     | 3               | 127    | 11            |                 | 1 veggie  | 1    | ""                |
| 13 | Mandy   | 44  | 2               | 68     | 8             |                 | 1 chicken |      | null              |

## R Studio

The screenshot shows the R Studio interface with the following components:

- Top Bar:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Project Bar:** Untitled1\*, dat\*.
- Data View:** A data frame named "dat" with 12 rows and 9 columns. The columns are: name, age, years\_seniority, income, parking\_space, attending\_party, entree, pets, emergency\_contact. The data includes names like Tony, Donald, Henry, Janet, Nick, Bruce, Steve, Clint, Wanda, Natasha, Carol, and values for their respective attributes.
- Environment View:** Shows the global environment with "dat" defined.
- Help View:** The "Arithmetic Mean" documentation is open, showing the description, usage, and arguments for the mean function.
- Console View:** Displays R code and its output. The code includes reading the data, checking for missing values, and viewing the data frame.

```
R code from the Console
#> # Check the data for missing values
#> sapply(dat, function(x) sum(is.na(x)))
#>
#> View(dat)
#>
```

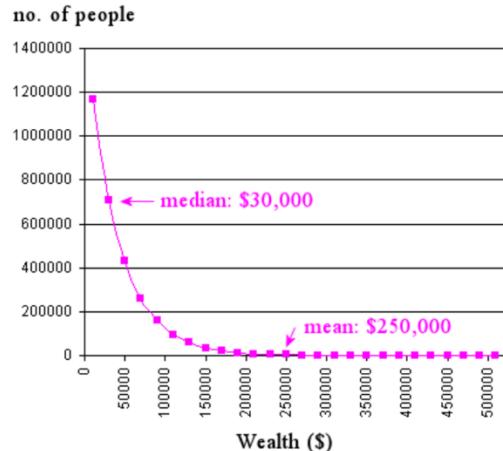
## Data Frame Output (from View(dat))

|    | name    | age | years_seniority | income | parking_space | attending_party | entree    | pets | emergency_contact |
|----|---------|-----|-----------------|--------|---------------|-----------------|-----------|------|-------------------|
| 1  | Tony    | 48  | 27              | NA     | 1             |                 | 5 shrimp  | NA   | Pepper            |
| 2  | Donald  | 67  | 25              | 86     | 10            |                 | 2 beef    | NA   | Jane              |
| 3  | Henry   | 69  | 21              | 95     | 6             |                 | 1 chicken | 62   | Janet             |
| 4  | Janet   | 62  | 21              | 110    | 3             |                 | 1 beef    | NA   | Henry             |
| 5  | Nick    | NA  | 17              | NA     | 4             |                 | NA NA     | NA   | NA                |
| 6  | Bruce   | 37  | 14              | 63     | NA            |                 | 1 veggie  | NA   | NA                |
| 7  | Steve   | 83  | NA              | 77     | 7             |                 | 1 chicken | NA   | n/a               |
| 8  | Clint   | 27  | 9               | 118    | 9             |                 | NA shrimp | 3    | None              |
| 9  | Wanda   | 19  | 7               | 52     | 2             |                 | 2 shrimp  | NA   | empty             |
| 10 | Natasha | 26  | 4               | 162    | 5             |                 | 3 NA      | NA   | -                 |
| 11 | Carol   | NA  | 2               | 127    | 11            |                 | 1 veggie  | 1    | " "               |

## General commands

```
1 rm(list = ls()) # clear work space
2 setwd("d:/temp") # set current work directory
3 sessionInfo() # get session information
4 installed.packages() # list installed packages
5 # import data file
6 dat <- read.csv("missing_values.csv", na.strings = "")
7 str(dat) # show data frame structure
8 # Check the data for missing values
9 sapply(dat, function(x) sum(is.na(x)))
```

## Replace missing values with the mean



| Sample Mean                  | Population Mean          |
|------------------------------|--------------------------|
| $\bar{X} = \frac{\sum X}{n}$ | $\mu = \frac{\sum X}{N}$ |

where  $\sum X$  is sum of all data values

$N$  is number of data items in population

$n$  is number of data items in sample

```

1 # Replace missing values with the mean
2 # column = age
3 # Missing values type = distributed
4 # Formal name = Missing Completely at Random (MCAR)
5 rm(list = ls()) # clear work space
6 setwd("d:/temp") # set current work directory
7 # import data file
8 dat <- read.csv("missing_values.csv", na.strings = "")
9 dat$age.mean <- ifelse(is.na(dat$age),
10 mean(dat$age, na.rm = TRUE),
11 dat$age)

```

## Replace missing values with the median

$$\text{Median} = l + \frac{h}{f} \left( \frac{N}{2} - c \right)$$

Where:

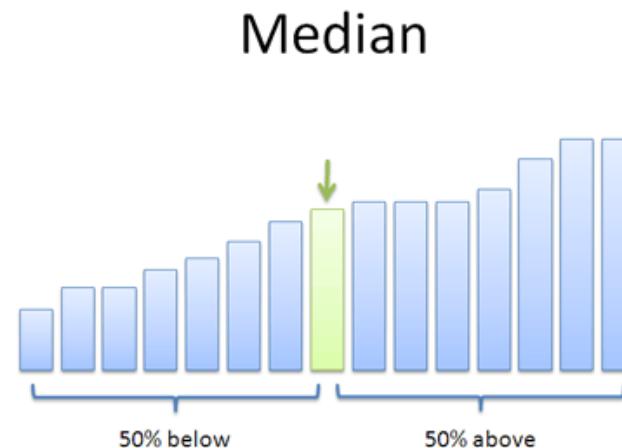
$l$  = lower class boundary of the median class

$h$  = Size of the median class interval

$f$  = Frequency corresponding to the median class

$N$  = Total number of observations i.e. sum of the frequencies

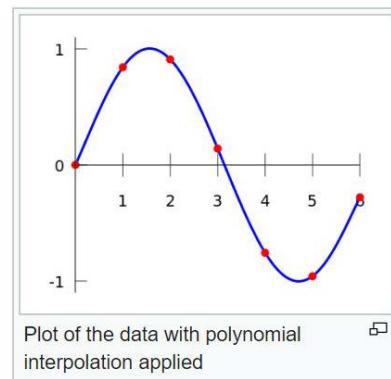
$c$  = Cumulative frequency preceding median class.



```
1 # Replace missing values with the median
2 # column = age
3 # • Another justifiable way to handle missing-at-random data
4 rm(list = ls()) # clear work space
5 setwd("d:/temp") # set current work directory
6 # import data file
7 dat <- read.csv("missing_values.csv", na.strings = "")
8 dat$age.mean <- ifelse(is.na(dat$age),
9 median(dat$age, na.rm = TRUE), dat$age)
```

## Replace missing values with an interpolated estimate

```
1 # Replace missing values with an interpolated estimate
2 # column = years_seniority
3 # the values in this column, years seniority, is ordered,
4 # greatest to least. This structure can be exploited by
5 # interpolating the missing value. This approach is very effective
6 # when it is appropriate, usually with time - series data.
7 rm(list = ls()) # clear work space
8 setwd("d:/temp") # set current work directory
9 # import data file
10 dat <- read.csv("missing_values.csv", na.strings = "")
11 dat$senior <- ifelse(is.na(dat$years_seniority), 11.5,
12 dat$years_seniority)
```



## Replace missing values with a constant

```
1 # Replace missing values with a constant
2 # column = income
3 # Missing values are Missing Not at Random(MNAR)
4 # Those with very high incomes preferred not to state them
5 # Make a reasonable guess for what "high" means and fill
6 # in the blanks. It will still be inaccurate, but better than blank
7
8 rm(list = ls()) # clear work space
9 setwd("d:/temp") # set current work directory
10 # import data file
11 dat <- read.csv("missing_values.csv", na.strings = "")
12 dat$income4 <- ifelse(is.na(dat$income), 250, dat$income)
```

## Replace missing values using imputation MICE

```
1 # Column = years_seniority, age, income
2 # Method = multivariate imputation by chained equations(MICE)
3 # MICE is the method of choice for complex incomplete data
4 # Good for missing data in more than one variable
5 # powerful when features are somewhat related
6
7 rm(list = ls())
8 dat <- read.csv("missing_values.csv", na.strings = "")
9 # A fast, consistent tool for working with data frame
10 install.packages("dplyr")
11 library(dplyr) # attach packages
12 # Make new dataframe with column years_seniority,
13 # age, income with column type = numeric
14 dat <- dat %>% # make new 3 columns with type numeric
15 mutate(
16 senior1 = as.numeric(years_seniority),
17 age1 = as.numeric(age),
18 income1 = as.numeric(income)
19)
```

## Replace missing values using imputation MICE

```
21 # Replace missing values using imputation MICE
22 keep <- c("senior1", "age1", "income1")
23 #drop all columns but keep 3 col
24 dat <- dat[, keep, drop = FALSE]
25 install.packages('mice') # standard command
26 library(mice) # standard command
27 init = mice(dat, maxit = 0) # standard command
28 meth = init$method # standard command
29 predM = init$predictorMatrix
30 # Bayesian linear regression (การวิเคราะห์คด拐อยเชิงเส้นแบบเบย์ส)
31 meth[c("senior1")] = "norm"
32 Predictive mean matching
33 meth[c("age1")] = "pmm" #
34 meth[c("income1")] = "pmm"
35 # Replace missing values using imputation MICE
36 set.seed(103) # seed for pseudo random number generator
37 imputed = mice(dat, method = meth, predictorMatrix = predM, m = 5)
38 imputed <- complete(imputed)
```

## Replace missing values with a missing rank

```
1 # Replace missing values with a missing rank
2 # Column = parking_space
3 # Our knowledge of how parking spaces are numbered,
4 # let us make a guess here
5 # All the space numbers from 1 - 11
6 # Missing one might be 12
7 rm(list = ls())
8 dat <- read.csv("missing_values.csv", na.strings = "")
9 dat$park <- ifelse(is.na(dat$parking_space), 12,
10 dat$parking_space)
```

## Replace missing values with a dummy

```
1 # eplace missing values with a dummy
2 # Column = parking_space
3 # Filling in a dummy value
4 # Clearly different from actual values
5 # Such as a negative rank
6 # Used to indicate that the feature is not applicable
7
8 rm(list = ls())
9 dat <- read.csv("missing_values.csv", na.strings = "")
10 dat$park <- ifelse(is.na(dat$parking_space), -99,
11 dat$parking_space)
```

## Replace missing values with 0

```
1 # eplace missing values with 0
2 # Column = attending_party
3 # A missing numerical value can mean zero.
4 # In the case of an RSVP, invitees who are not planning
5 # to attend sometimes neglect to respond, but guests
6 # planning to attend are more likely to.
7 # In this case, filling in missing blanks with a zero is reasonable
8 rm(list = ls())
9 dat <- read.csv("missing_values.csv", na.strings = "")
10 dat$party <- ifelse(is.na(dat$attending_party), 0, dat$attending_party)
```

## Create an indicator variable for "missing"

```
1 # Create an indicator variable for "missing"
2 # Column = pets
3 # Replacing missing values requires making assumptions.
4 # Whenever your confidence in those assumptions is low,
5 # it is safer to also create a true / false feature
6 # indicating that the value was missing.
7 # This allows many algorithms to learn to weight those differently.
8 rm(list = ls())
9 dat <- read.csv("missing_values.csv", na.strings = "")
10 dat$pet1 <- ifelse(is.na(dat$pet), 0, dat$pets)
11 dat$pet2 <- complete.cases(dat$pets)
```

## Replace missing values with a string

```
1 # Replace missing values with a string
2 # Column = emergency_contact
3 # Replace NA with "no"
4
5 rm(list = ls())
6 dat <- read.csv("missing_values.csv", na.strings = "")
7 dat$emer <- ifelse(is.na(dat$emergency_contact), 'no',
8 dat$emergency_contact)
```

Add an indicator variable showing which strings are considered "missing."

```
1 # Add an indicator variable showing which strings
2 # are considered "missing."
3 # Column = emergency_contact
4 # There are lots of ways to communicate the concept
5 # of "missing" in a string
6 # Replace no, NA, n / a, None, _, "", empty, null with 0
7 # Otherwise = 1
8 rm(list = ls())
9 dat <- read.csv("missing_values.csv", na.strings = "")
10 dat$emer <- ifelse(dat[9] == 'NA' |
11 is.na(dat[9]) | dat[9] == 'n/a' |
12 dat[9] == 'None' | dat[9] == 'empty' |
13 dat[9] == '_' | dat[9] == '""' | dat[9] == 'null', 0, 1)
```

Delete columns that are missing too many values to be useful

```
1 # Column = pets
2 # Is a feature that missing too many values
3 # Not enough information available to make reasonable
4 # assumptions about how to replace the missing values
5 # Best policy = delete the column entirely.
6
7 rm(list = ls())
8 dat <- read.csv("missing_values.csv", na.strings = "")
9 dat <- dat[, !(names(dat) %in% 'pets')]
```

## Delete rows that are missing critical values

```
1 # Delete rows that are missing critical values
2 # Rows that are missing important features can be deleted.
3 # This is particularly useful when you have the luxury of
4 # hand - picking high - quality data
5 # such as when training a model
6 rm(list = ls())
7 dat <- read.csv("missing_values.csv", na.strings = "")
8 dat <- dat[complete.cases(dat),]
```

## More information on Missing value handling in R

Bayesian linear regression analysis without tears (R)

<https://www.r-bloggers.com/bayesian-linear-regression-analysis-without-tears-r/>

Source code

<https://github.com/laploy/ML/blob/master/Missing%20R%20Script.zip>

# **SETUP MICROSOFT POWER BI**



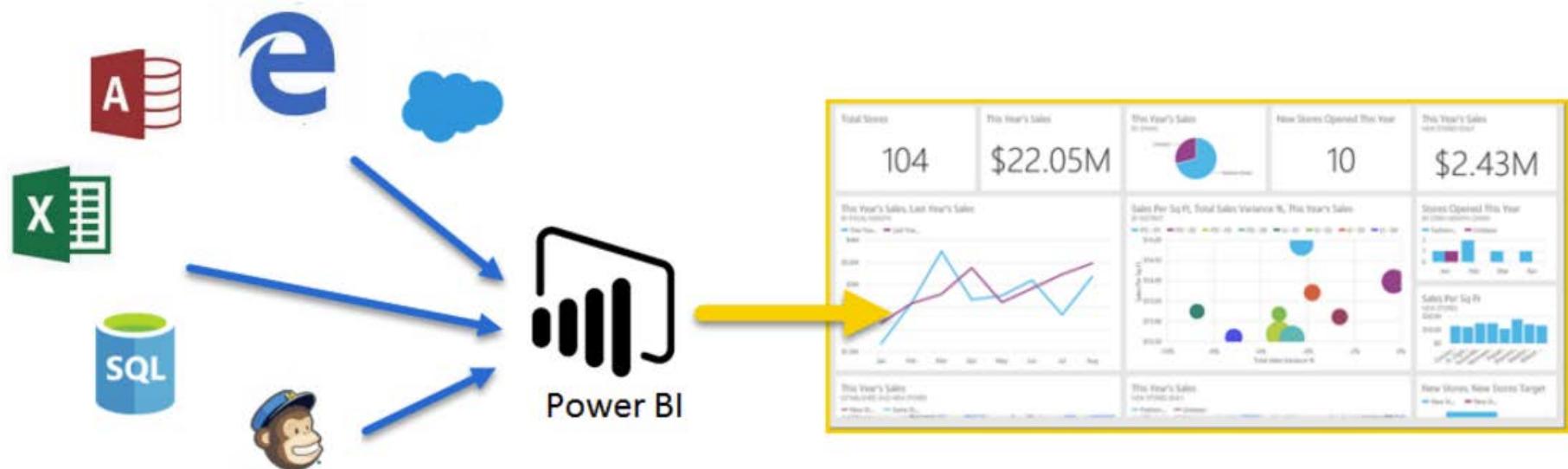
**Power BI**

## In this session

- What is Microsoft Power BI?
- Go to website Microsoft Power BI
- Choose how to get started
- Get started
- Email sent notification
- Check your email
- Create Microsoft Power BI account
- Power BI account initialization
- Power BI welcome page

## What is Microsoft Power BI?

- A collection of software services, apps, and connectors
- Turn unrelated sources of data into interactive insights.
- Easily connect to data sources
- Visualize
- Share with anyone
- Simple and fast
- Robust and enterprise-grade
- Real-time analytics



Go to website Microsoft Power BI

<https://powerbi.microsoft.com/>

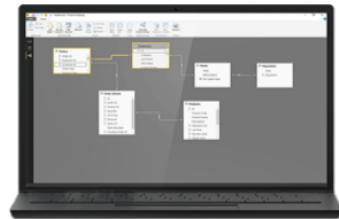
Click Get Started free

The screenshot shows the Microsoft Power BI homepage. At the top, there is a navigation bar with the Microsoft logo, 'Power BI' text, and links for 'Products', 'Solutions', 'Partners', 'Learning', and 'Sign in'. Below the navigation bar, there is a large yellow background area with a speaker icon and the text 'Bring your data to life with Microsoft Power BI'. A red-bordered button labeled 'Get started free' is located at the bottom left of this yellow area. To the right, a laptop screen displays a complex dashboard titled 'Executive Metrics Dashboard' containing various charts and graphs.

Choose how to get started

Click Sign up

## Choose how to get started



### Power BI Desktop for Windows

**Analytics tools at your fingertips**

Connect and transform data, create advanced calculations, and build stunning reports in minutes.

[Download](#)



### Power BI

**The easy way to see your important data in one place**

With a few clicks, connect to data from applications you use and get started with pre-built dashboards from experts.

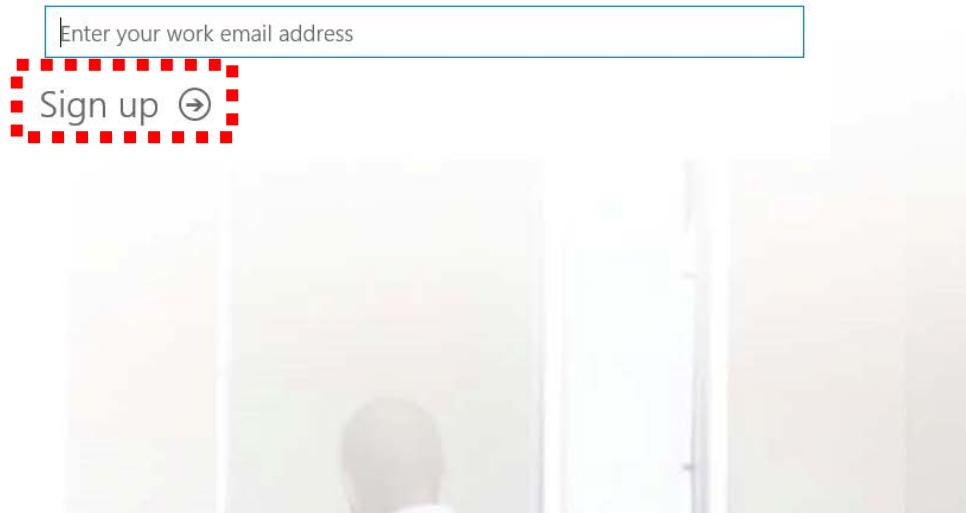
[Sign up](#)

## Get started

Enter work email address / click Sign up

Microsoft Power BI

Get started



Email sent notification

An email is sent to you inbox

Microsoft Power BI

Great! Go check your email.

To finish signing up, click the link in the mail from Power BI.

Didn't get the mail? Check your spam folder or [resend the mail](#)

## Check your email

Check email and Click Yes, that's me

**Finish signing up for Microsoft Power BI**

From Power BI    
To test@generalcomtech.com    
Reply-To Microsoft Online Services Team    
Date Today 16:15

[View this email in your browser.](#)

Microsoft Power BI

Your data awaits.  
We just need to verify your address.

Does this look right?  
[test@generalcomtech.com](mailto:test@generalcomtech.com)

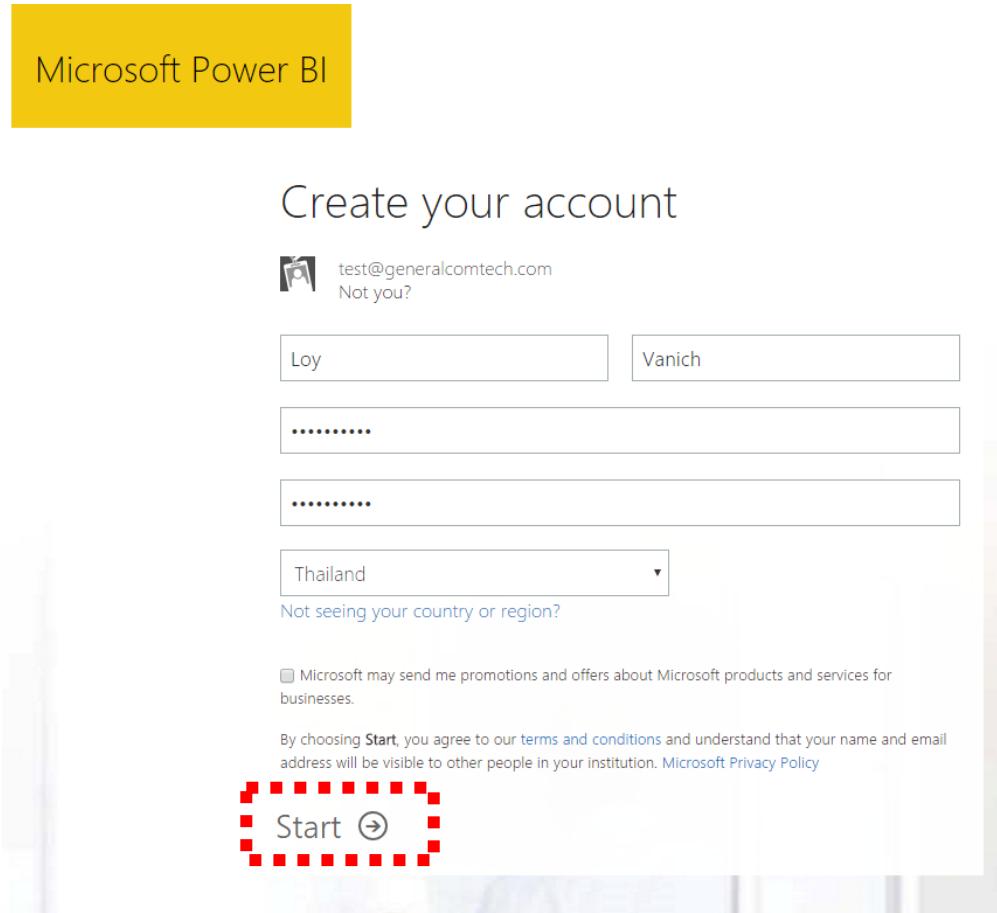
 Yes, that's me

And FYI: Because you're signing up with a work email address, your employer may control your communications and data. Their policies apply to your use of the service.

Don't want to sign up? Just ignore this email. Thanks.

## Create Microsoft Power BI account

Enter information and click Start



The image shows a screenshot of the Microsoft Power BI account creation page. At the top, there's a yellow header bar with the text "Microsoft Power BI". Below it, the main title "Create your account" is centered. To the left of the input fields, there's a small user icon and the email address "test@generalcomtech.com". Below the email, there's a link "Not you?". The form consists of several input fields: first name ("Loy"), last name ("Vanich"), a password field with masked input ("....."), a confirmation password field with masked input ("....."), and a dropdown for country/region ("Thailand"). Below the dropdown, there's a link "Not seeing your country or region?". Further down, there's a checkbox for receiving promotions ("Microsoft may send me promotions and offers about Microsoft products and services for businesses.") and a note about terms and conditions ("By choosing **Start**, you agree to our [terms and conditions](#) and understand that your name and email address will be visible to other people in your institution. [Microsoft Privacy Policy](#)"). At the bottom, there's a large red button with the text "Start" and a circular arrow icon.

Microsoft Power BI

### Create your account

 test@generalcomtech.com  
Not you?

Loy Vanich

.....

.....

Thailand ▾

Not seeing your country or region?

Microsoft may send me promotions and offers about Microsoft products and services for businesses.

By choosing **Start**, you agree to our [terms and conditions](#) and understand that your name and email address will be visible to other people in your institution. [Microsoft Privacy Policy](#)

**Start** 

Invite more people

Send invitations or Skip

Microsoft Power BI

Invite more people

Power BI makes it easy to create and share data stories. Tell your friends. It's free.

User name

@generalcomtech.com

Send invitations →

Skip

Power BI account initialization

Wait for Setup initialize

# กำลังจัดเตรียม Power BI

การตั้งค่ากำลังจะเสร็จสิ้น...

---

เหลือเวลาอีกน้อยกว่าหนึ่งนาที

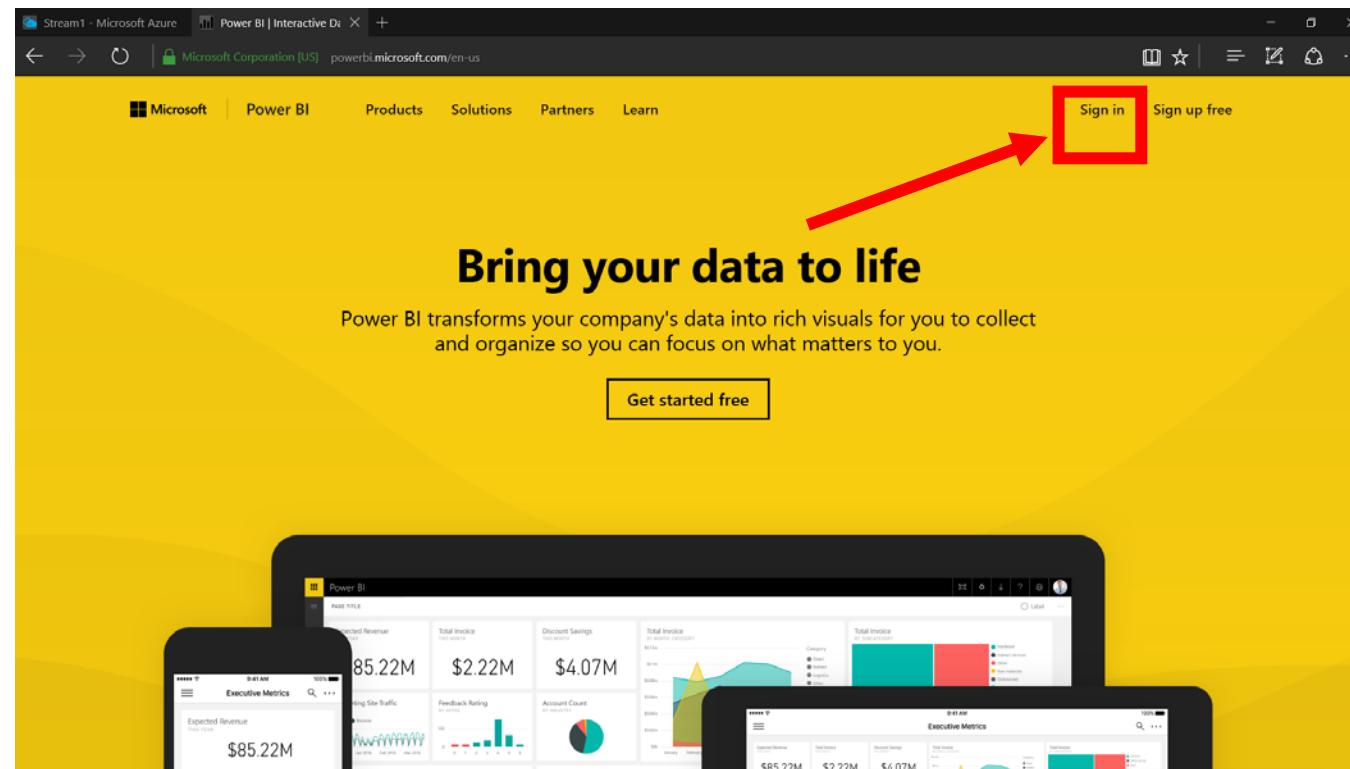
## Power BI welcome page

Your account is ready

The screenshot shows the Power BI welcome page. At the top, there's a navigation bar with icons for Home, Recent, Settings, Help, and Profile. Below the bar, a search icon is visible. The main heading is "ยินดีต้อนรับสู่ Power BI" (Welcome to Power BI). A sub-headline says "คุณกำลังทำการสำรวจข้อมูลของคุณและตรวจสอบรายการที่มีความสำคัญ มาเริ่มต้นโดยการรับข้อมูลบางรายการกัน" (You are performing a data exploration and checking important items. Start by receiving some data). Below this, there's a note: "ต้องการค่าแนะนำเพิ่มเติมหรือไม่ [ลองทำตามโปรแกรมแนะนำนี้](#) หรือ [ชมวิดีโอ](#)" (Do you want more recommendations? [Try this recommended program](#) or [Watch video](#)). The page is divided into two main sections: "ไลบรารีชุดเนื้อหา" (Topic Library) and "นำเข้าหรือเขื่อมต่อกับข้อมูล" (Import or connect to data). The Topic Library section contains four cards: "องค์กรของฉัน" (My organization), "บริการ" (Services), "ไฟล์" (Files), and "ฐานข้อมูล" (Data sources). Each card has a "รับ" (Receive) button with a right-pointing arrow. The Data Sources section also has a "รับ" (Receive) button with a right-pointing arrow. At the bottom left, there's a "ตัวอย่าง" (Example) link with a document icon.

## Microsoft Power BI home page

Go to Microsoft Power BI page <https://powerbi.microsoft.com>  
and click Sign in



More information

## More on Microsoft Power BI Sign Up

Power BI Sign Up Walkthrough

<https://powerbi.microsoft.com/en-us/blog/power-bi-sign-up-walkthrough/>

# OneDrive



OneDrive

In this session

## In this session

- What is OneDrive
- To find OneDrive Folder
- To run desktop OneDrive
- To sign in desktop OneDrive
- Create and sync folder

## What is OneDrive

### What is OneDrive

- Previously SkyDrive, Windows Live
- Is a file-hosting service
- Operated by Microsoft
- Similar to DropBox, Google Drive
- Allows users to store files in the cloud
- Files can be synced to a PC
- Accessed from a web browser
- Free 5 GB space
- Build-in with Windows 10

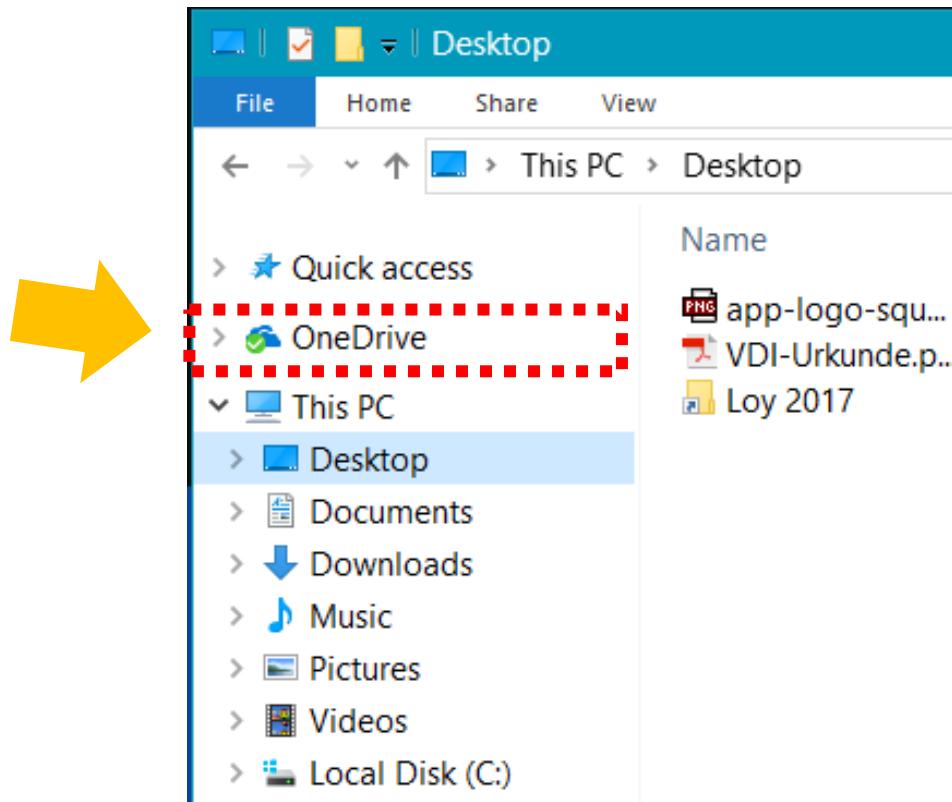
In this course we will use OneDrive to Sync data between Azure Machine Learning and Power BI



## To find OneDrive Folder

### To find OneDrive Folder

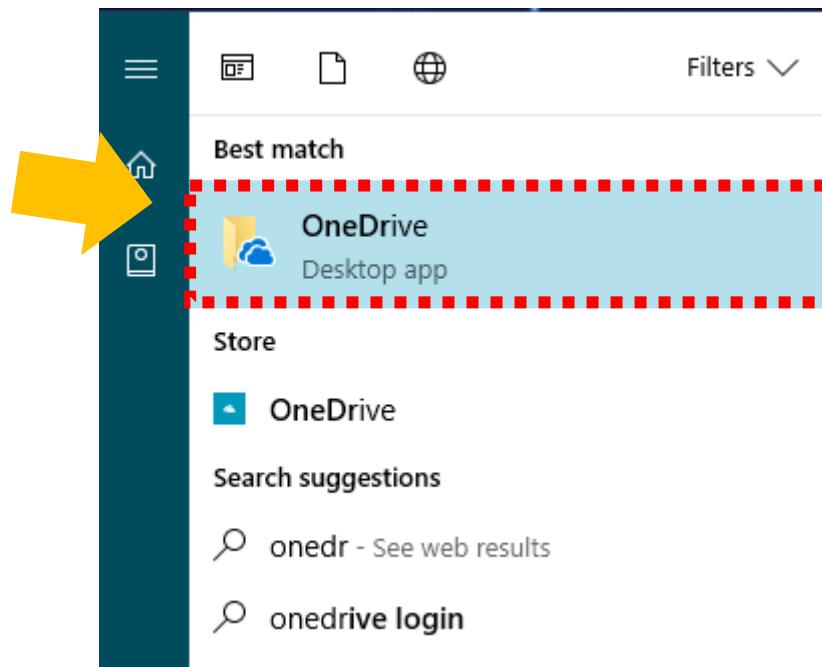
1. Open Windows Files Exploror
2. Locate for OneDrive Icon in the right plane



To run desktop OneDrive

To run desktop OneDrive

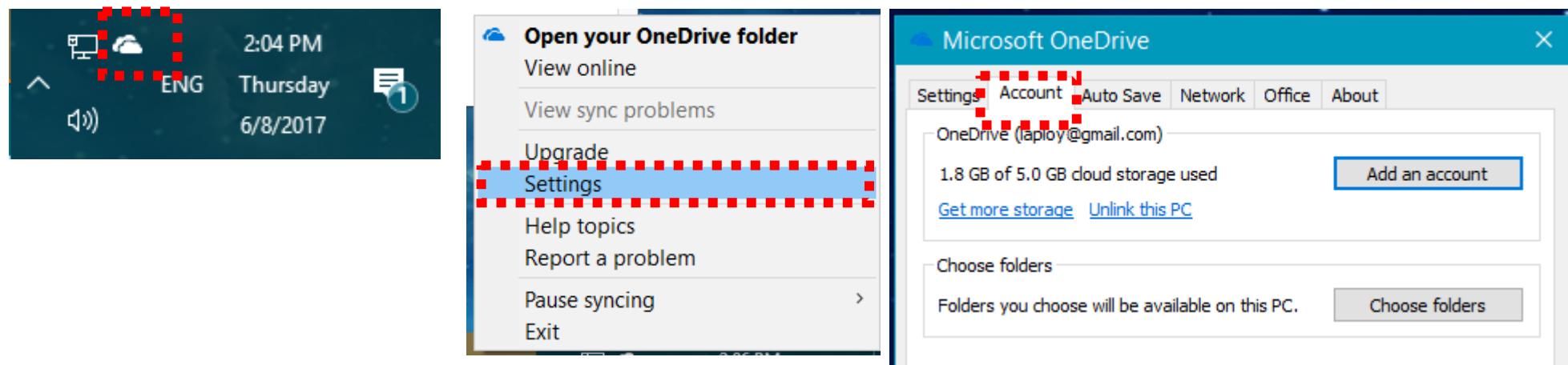
1. Click Start
2. Type OneDrive
3. Click OneDrive Icon



## To sign in desktop OneDrive

### To sign in desktop OneDrive

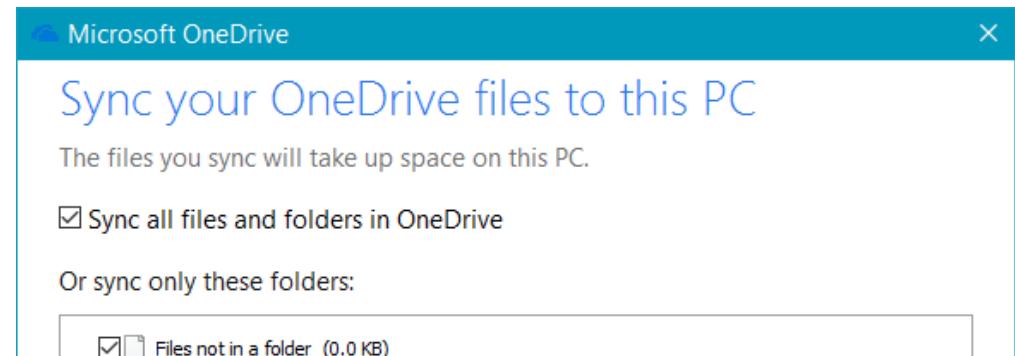
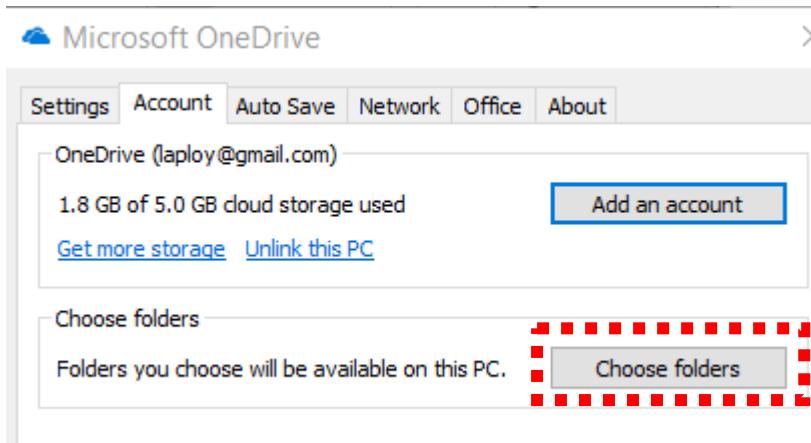
1. Locate OneDrive icon in system tray
2. Right click at the icon
3. Click Settings
4. Click Tab Account



## Create and sync folder

### Create and sync folder

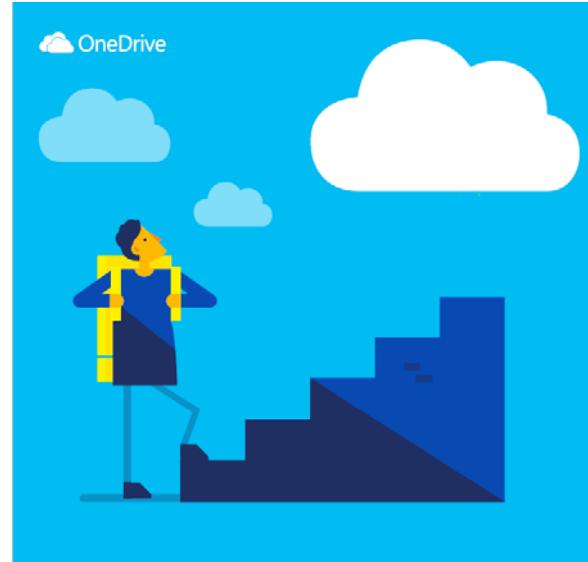
1. Open OneDrive Folder in Windows Files Explorer
2. Create sub folder ML
3. Right click OnDDrive icon
4. Select settings
5. Click tab Account
6. Click Choose folders
7. Make sure folder ML is in the sysn list



## More information

eBook: Get started with OneDrive

<https://support.office.com/en-us/article/eBook-Get-started-with-OneDrive-498739ec-8574-4439-9945-660a273966fa>



Get started with  
OneDrive

# **POWER BI AND ML**

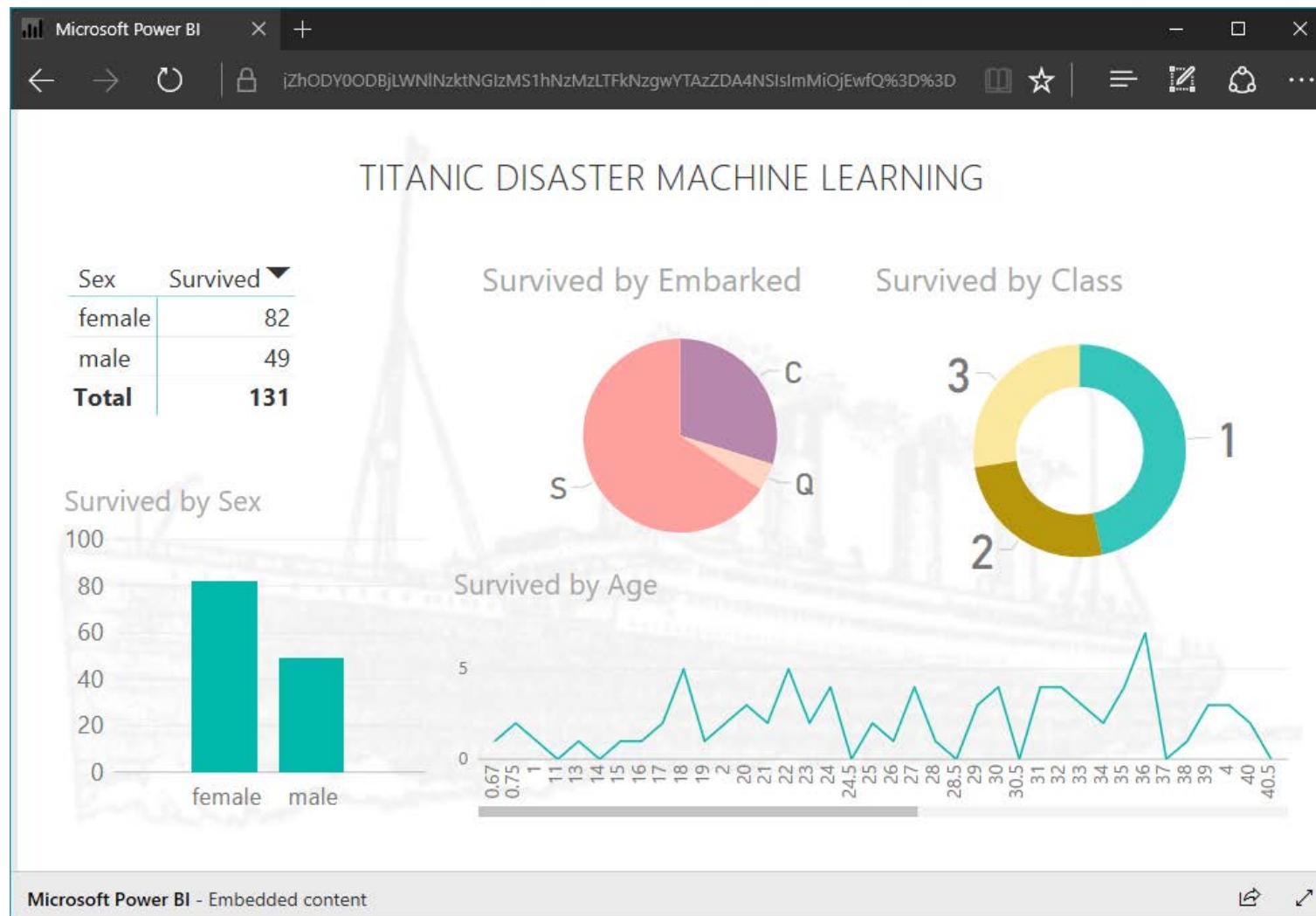


**Power BI**

## In this session

- Create and publish ML BI Report
- Get Dataset from GitHub
- Import or Connect to Data
- Get dataset from OneDrive
- Connect to dataset
- Create Report Title
- Add summarize matrix
- Add Column chart
- Add Pie chart
- Add Line chart
- Add Donut chart
- Add report background
- Publish Report

## Create and publish ML BI Report



## Get Dataset from GitHub

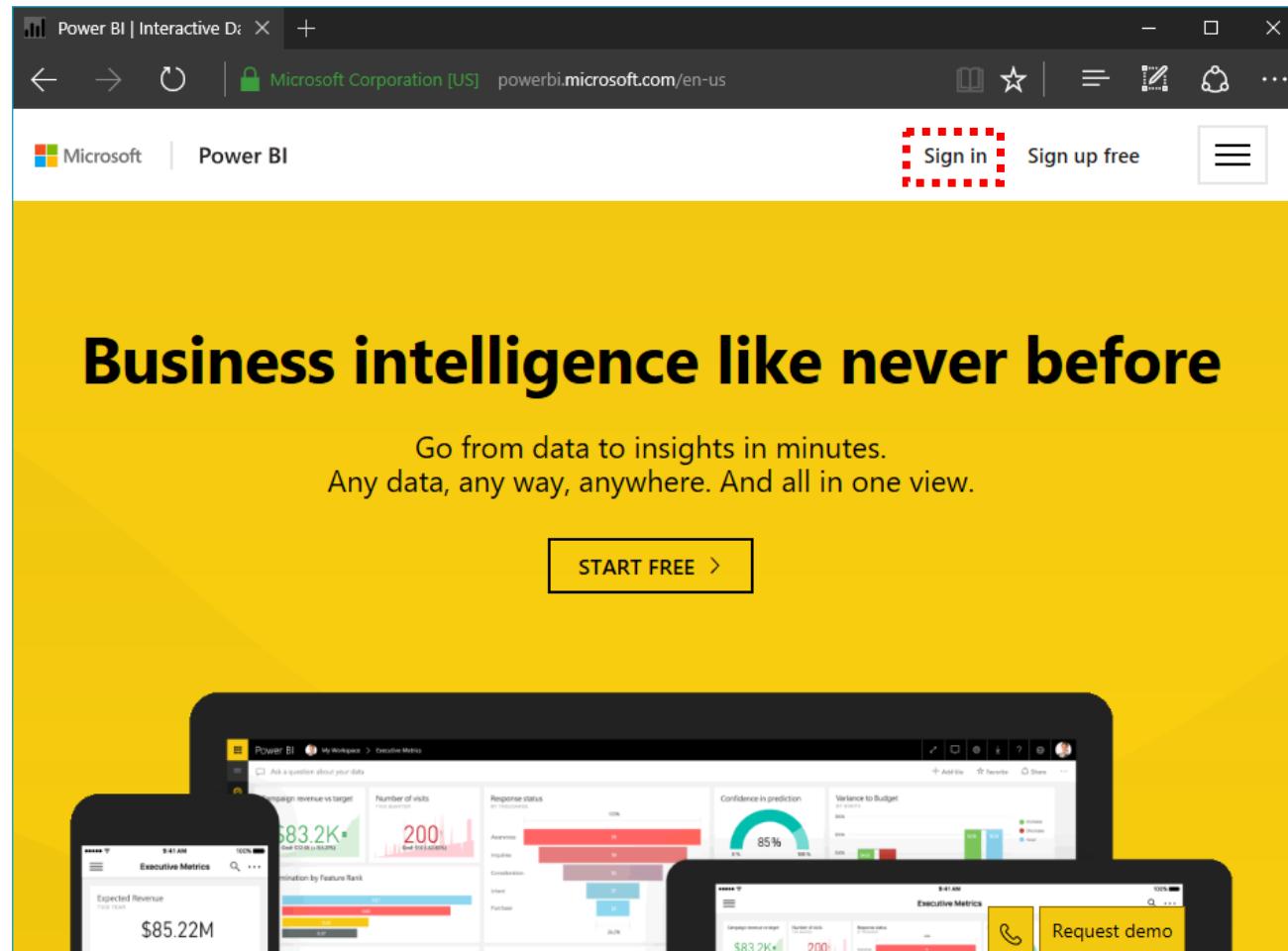
1. Go to <https://github.com/laploy/ML>
2. right click at `TitanicBI.csv`/save link as...
3. save file to ML folder created from previous session

The screenshot shows the GitHub repository page for 'laploy / ML'. The repository has 9 commits, 1 branch, 0 releases, and 1 contributor. The latest commit was 8 minutes ago. A red dotted line highlights the file `TitanicBI.csv`, which was added 8 minutes ago.

| File                                | Action               | Time Ago      |
|-------------------------------------|----------------------|---------------|
| <code>README.md</code>              | Update README.md     | 3 days ago    |
| <code>TitanicBI.csv</code>          | Add files via upload | 8 minutes ago |
| <code>TitanicData.csv</code>        | Add files via upload | 3 days ago    |
| <code>adult col R Script.txt</code> | Add files via upload | 3 days ago    |

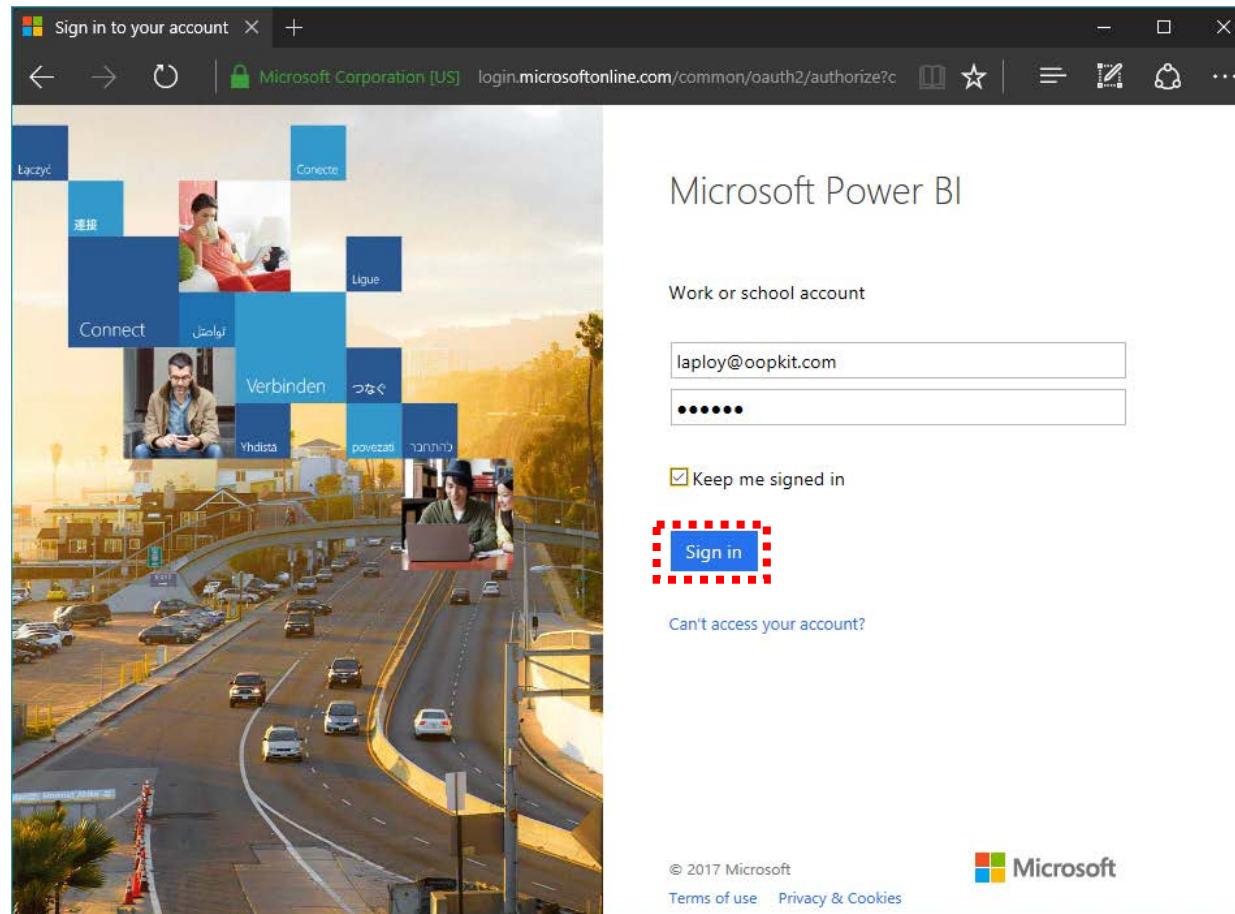
Go to Power BI home

<https://powerbi.microsoft.com/en-us>



## Login to Power BI

Login with credential created in the last session



## Import or Connect to Data

Click Get Data / Import or Connect to Data / Files

The screenshot shows the Microsoft AppSource 'Get Data' page. At the top center is a large 'Get Data' button. Below it is a sub-section titled 'Import or Connect to Data' containing four cards: 'Files' (with a red dashed box around its 'Get' button), 'Databases', 'My organization', and 'Services'. At the bottom are links for 'Samples', 'Solution Templates', and 'Partner Showcase'.

**Get Data**

Need more guidance? [Try this tutorial](#) or [watch a video](#)

**Microsoft AppSource**

**Import or Connect to Data**

**My organization**  
Browse content packs that other people in your organization have published.

**Services**  
Choose content packs from online services that you use.

**Files**  
Bring in your reports, workbooks, or data from Excel, Power BI Desktop or CSV files.

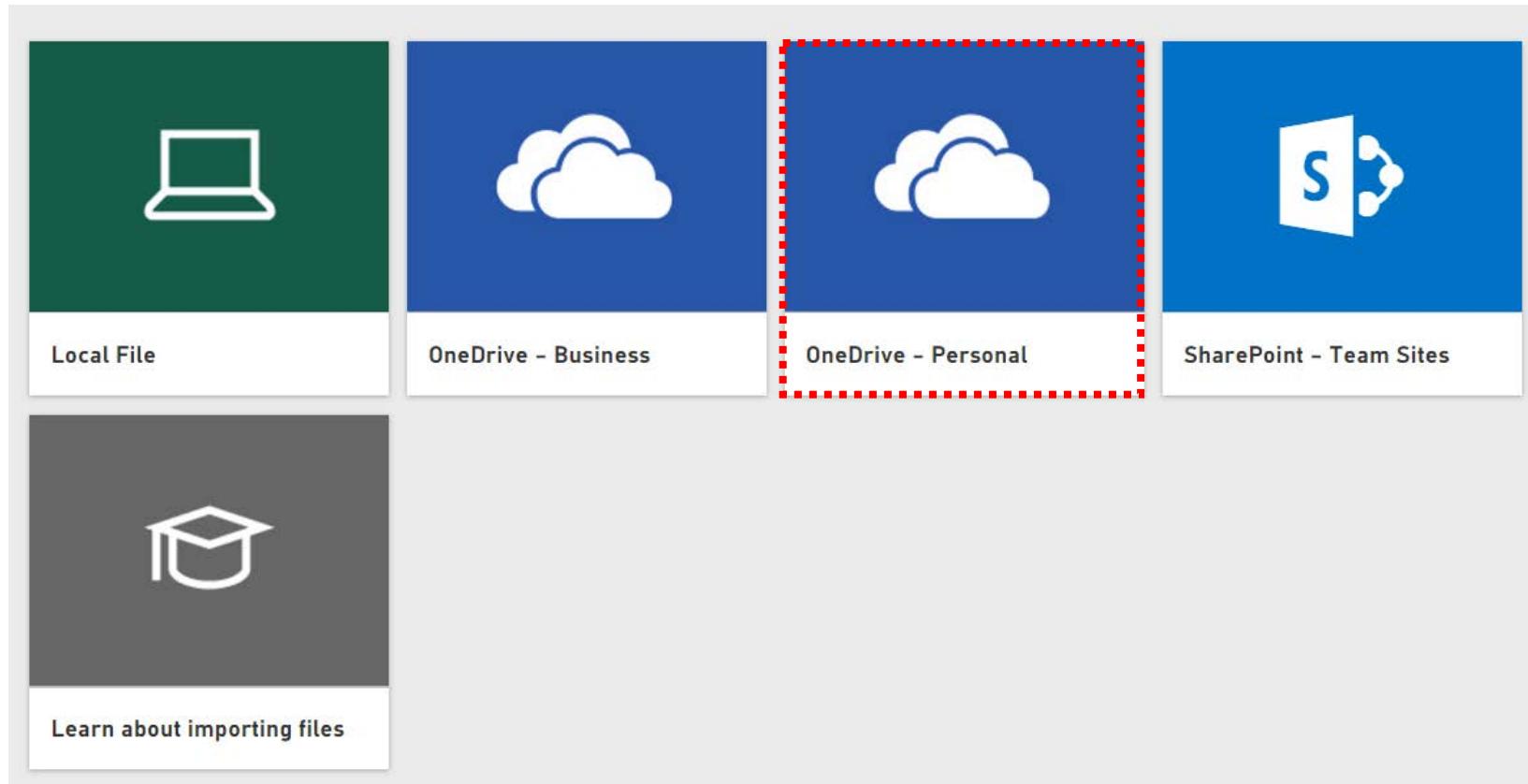
**Databases**  
Connect to live data in Azure SQL Database and more.

[Get →](#) Get → [Get →](#) [Get →](#)

[Samples](#) [Solution Templates](#) [Partner Showcase](#)

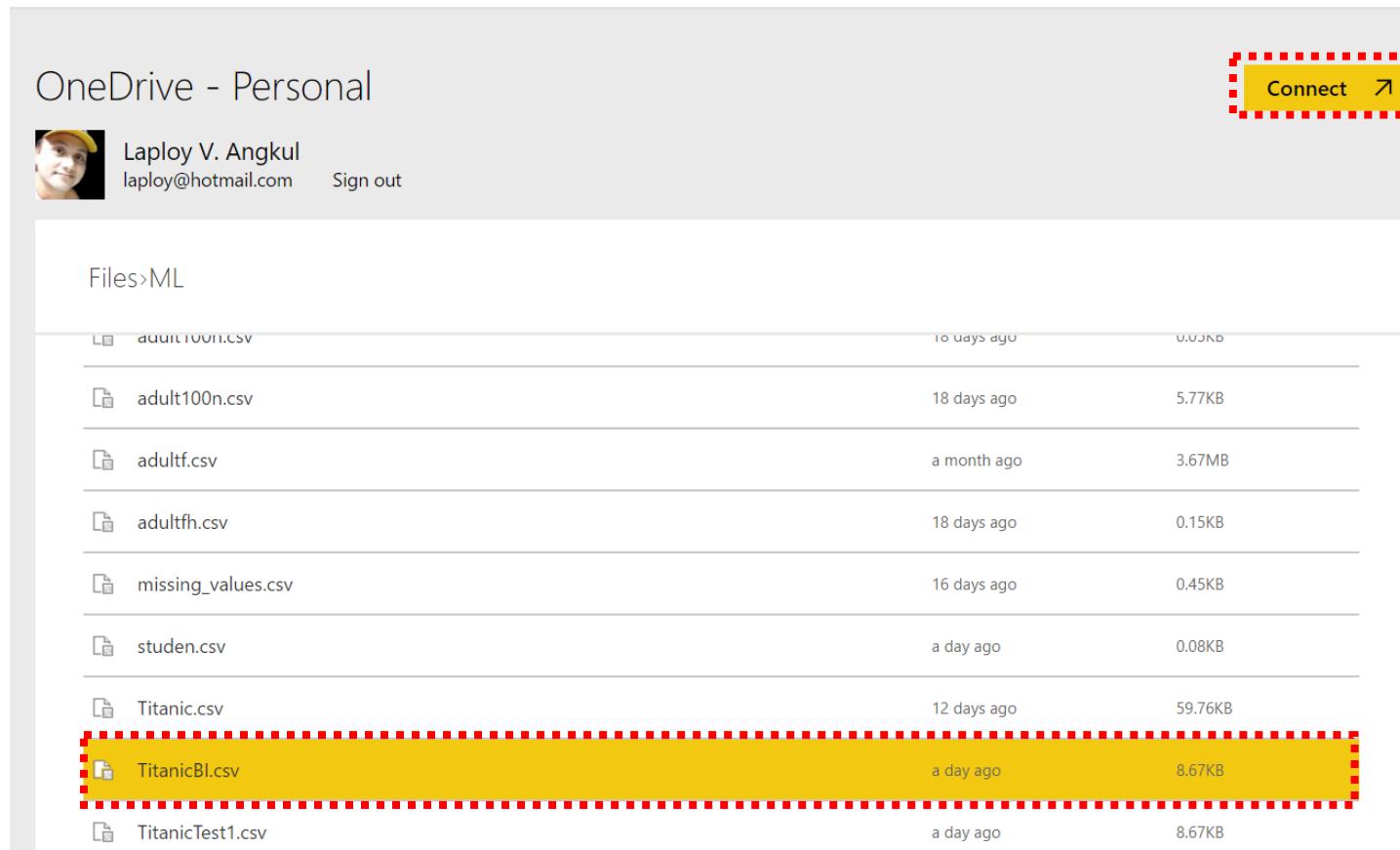
## Get dataset from OneDrive

Click OneDrive - Personal



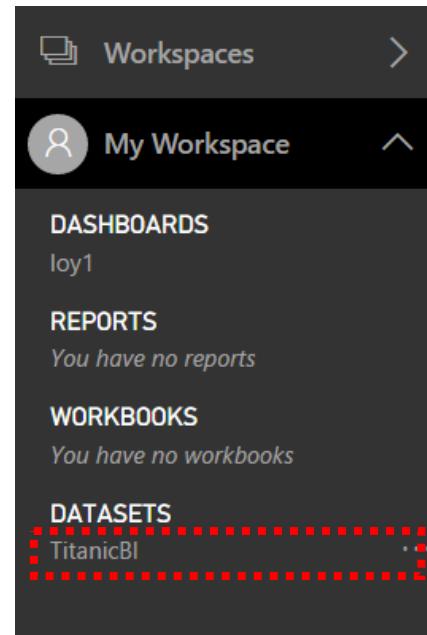
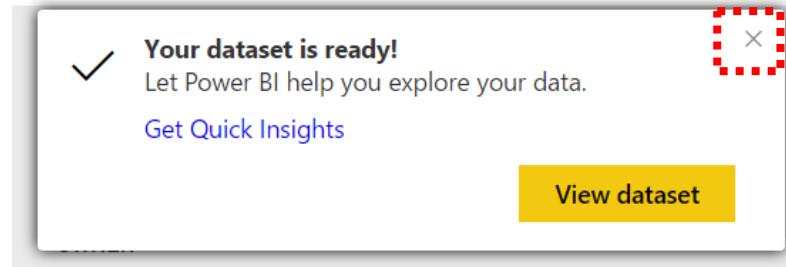
## Connect to dataset

Select file TitanicBI.csv in ML folder in OneDrive  
Click Connect



## Get data connection confirmation

Click x to close the confirmation dialog box

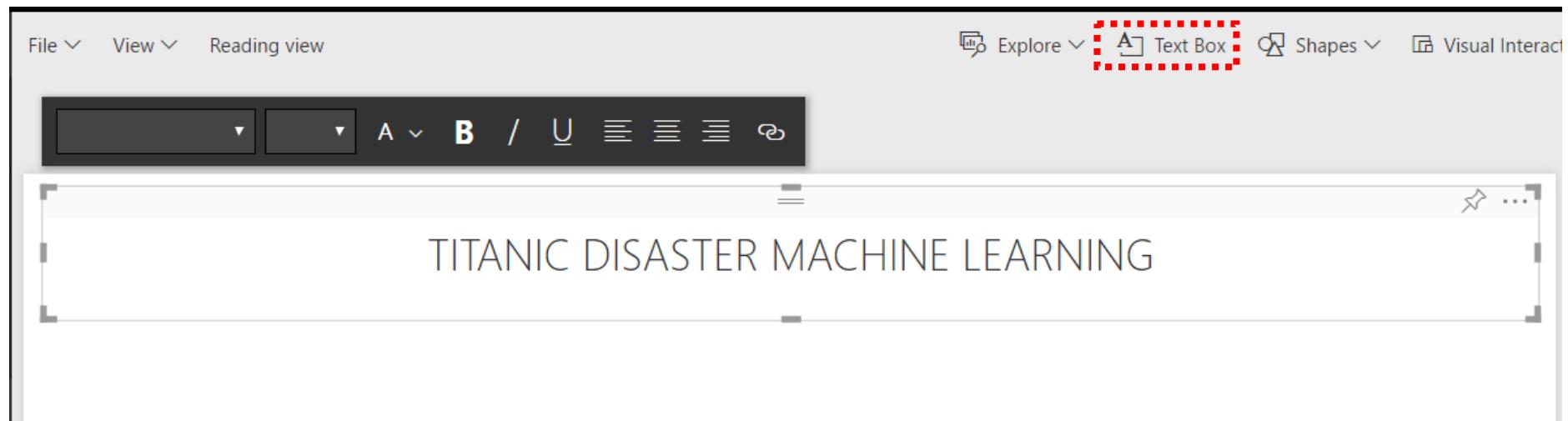


Click TitanicBI under DATASETS

## Create Report Title

### Create Report Title

- Click Text box
- Move text box to top
- Make text box width == report width
- Enter text TITANIC DISASTER MACHINE LEARNING
- Change text size to 36

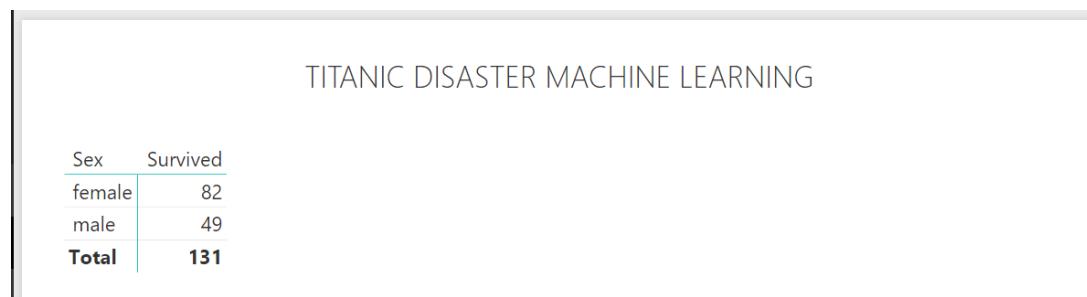


## Add summarize matrix

### Add summarize matrix

1. Click Matrix in Visualizations
2. Drag field Sex to Rows
3. Drag field Survived to Values
4. Change Matrix font size
5. Move to shown position

| Sex          | Survived   |
|--------------|------------|
| female       | 82         |
| male         | 49         |
| <b>Total</b> | <b>131</b> |

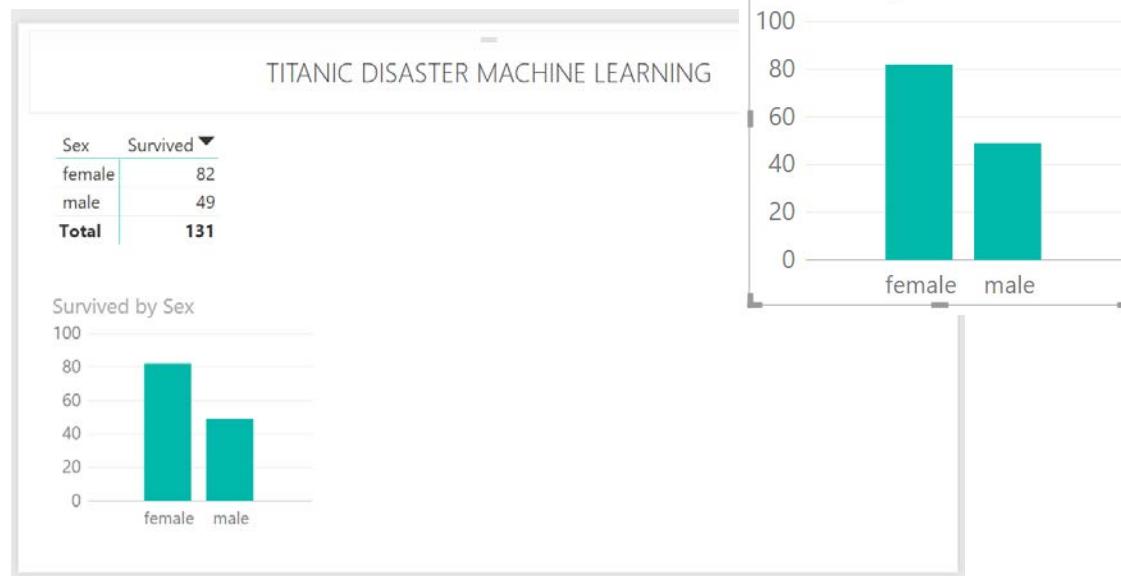


This screenshot shows the 'Visualizations' pane of the Power BI desktop application. It displays the configuration for the summarize matrix. Under 'Rows', 'Sex' is selected. Under 'Values', 'Survived' is selected. Under 'Filters', 'Sex(All)' and 'Survived(All)' are listed. Two yellow arrows point from the 'Values' and 'Filters' sections towards the corresponding settings in the matrix visualization on the left.

## Add Column chart

### Add summarize matrix

1. Click Column chart in Visualizations
2. Drag field sex to Rows
3. Drag field Survived to Values
4. Change X, Y, Title font size
5. Move to shown position

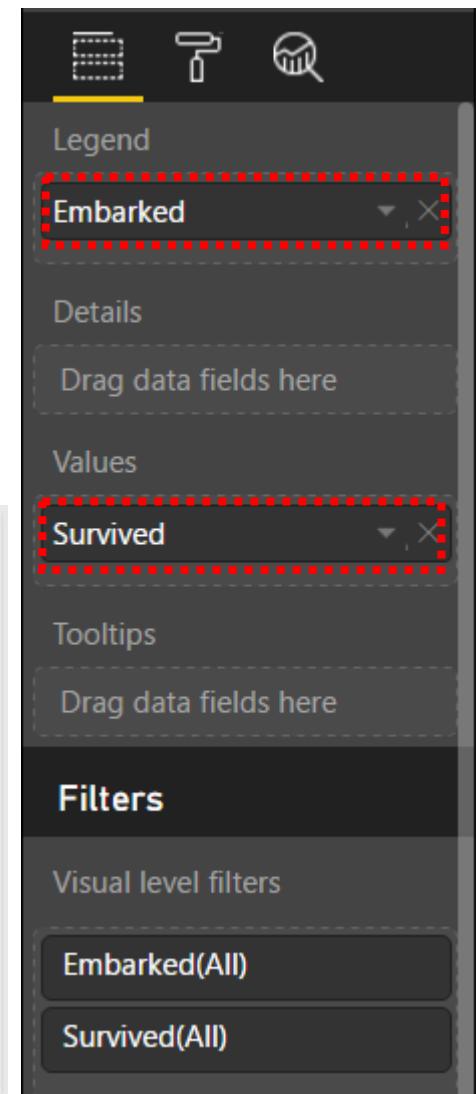
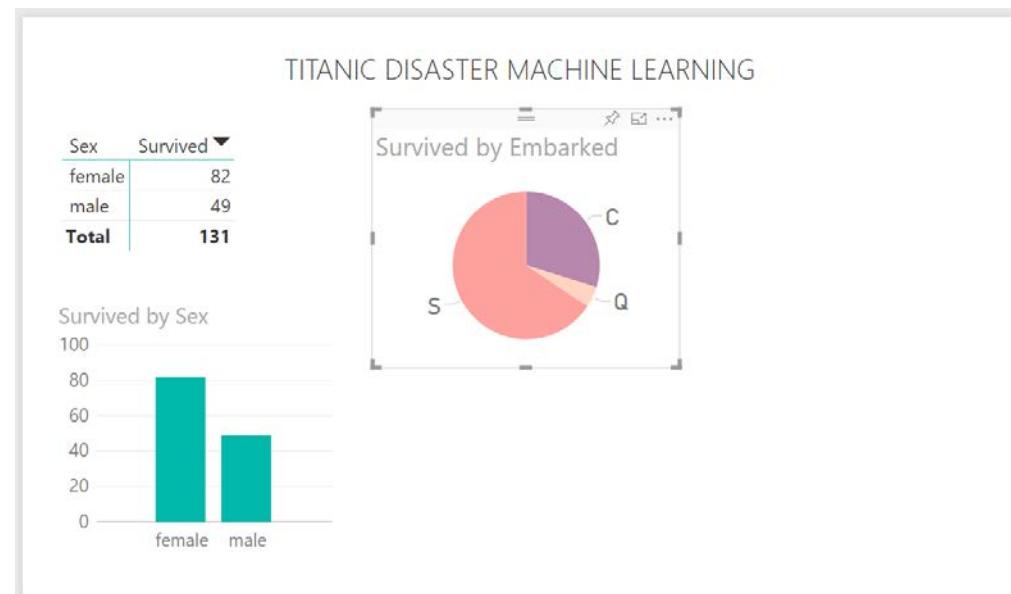


The screenshot shows the Power BI visualizations and fields panes. The visualizations pane on the left contains various chart icons, with the "Column chart" icon highlighted. The fields pane on the right lists fields from the "TitanicBI" dataset: Age, Class, Embarked, Fare, Parent, Sex (selected), Sibling, and Survived. The "Rows" section shows "Sex" selected. The "Values" section shows "Survived" selected. The "Filters" section shows "Sex(All)" and "Survived(All)" selected. Two yellow arrows point from the "Values" and "Filters" sections towards the corresponding areas in the screenshot below.

## Add Pie chart

### Add summarize matrix

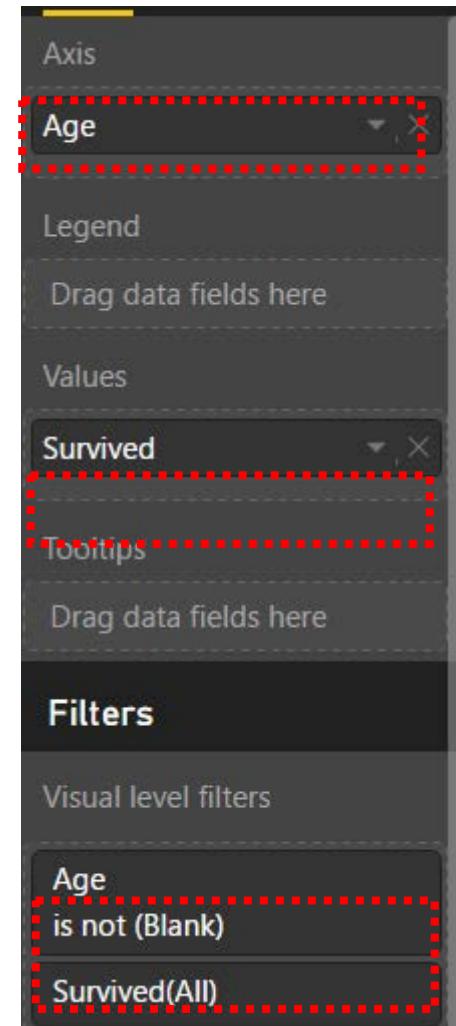
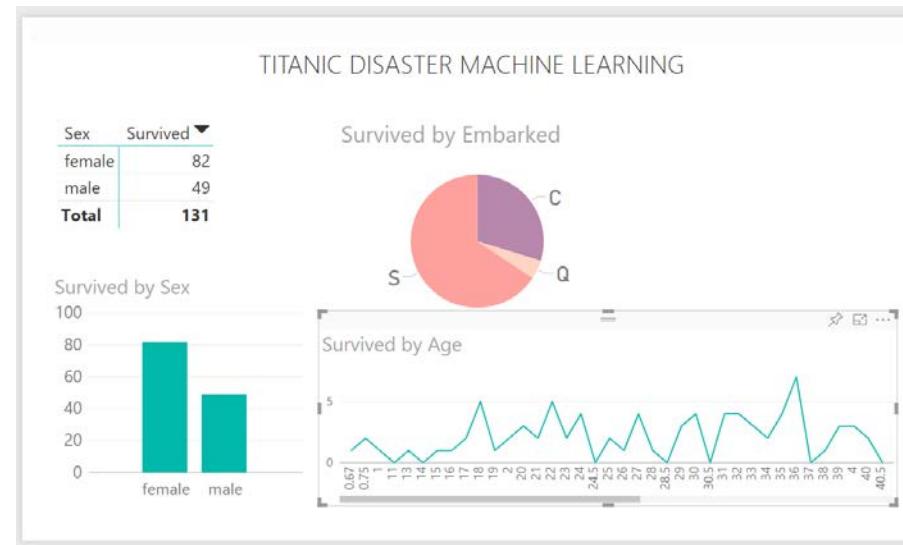
1. Click Pie chart in Visualizations
2. Drag field Embarked to Legend
3. Drag field Survived to Values
4. Change Legend and Title font size
5. Move to shown position



## Add Line chart

### Add summarize matrix

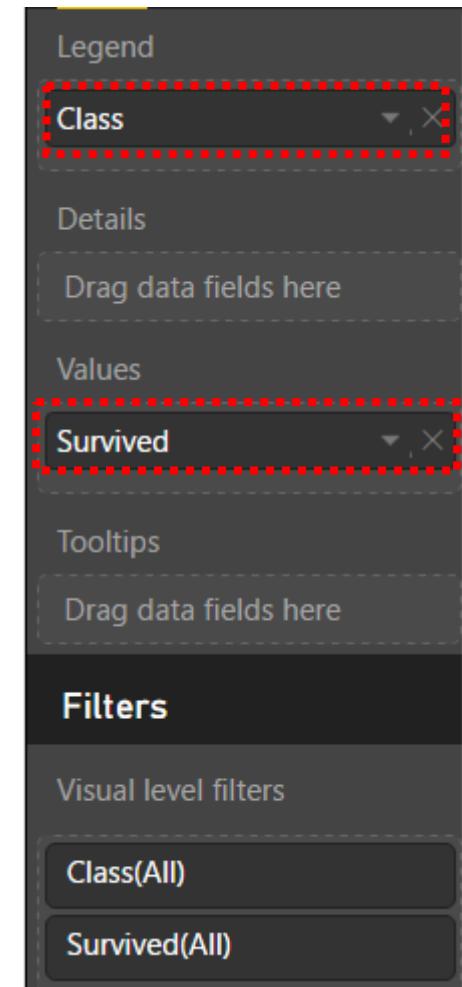
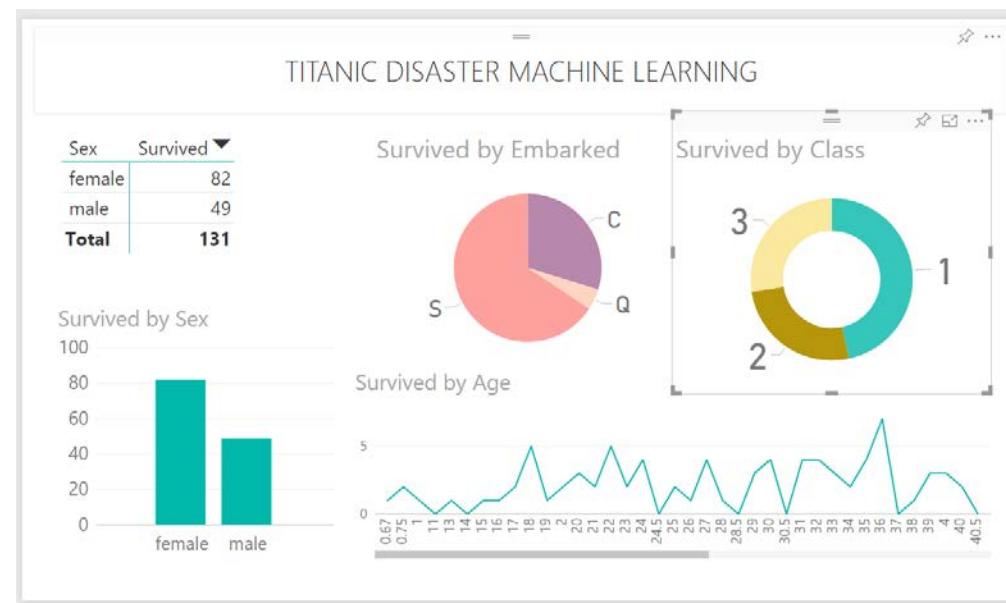
1. Click Line chart in Visualizations
2. Drag field Age to Axis
3. Drag field Survived to Values
4. Filter blank from Age
5. Change X,Y and Title font size
6. Move to shown position



## Add Donut chart

### Add summarize matrix

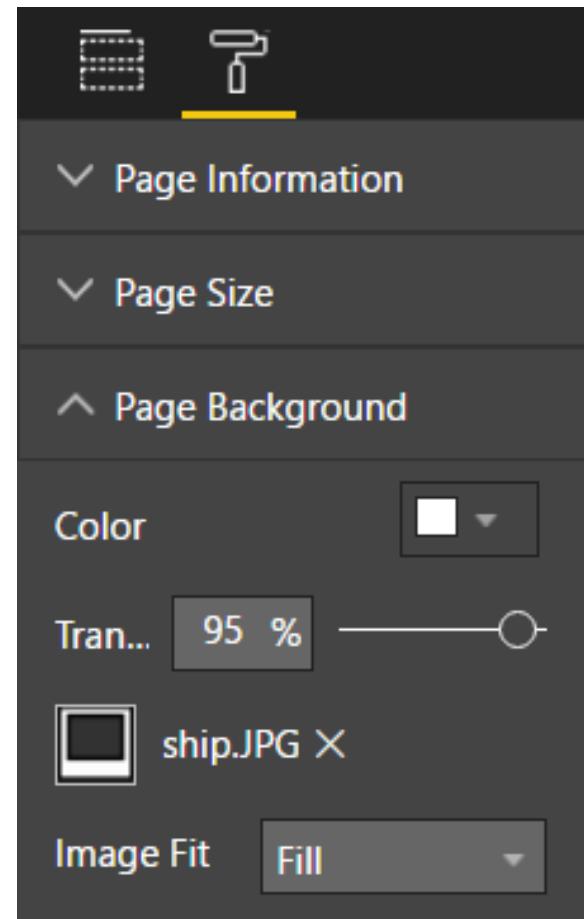
1. Click Line chart in Visualizations
2. Drag field Class to Legend
3. Drag field Survived to Values
4. Change font size
5. Move to shown position



## Add report background

### Add report background

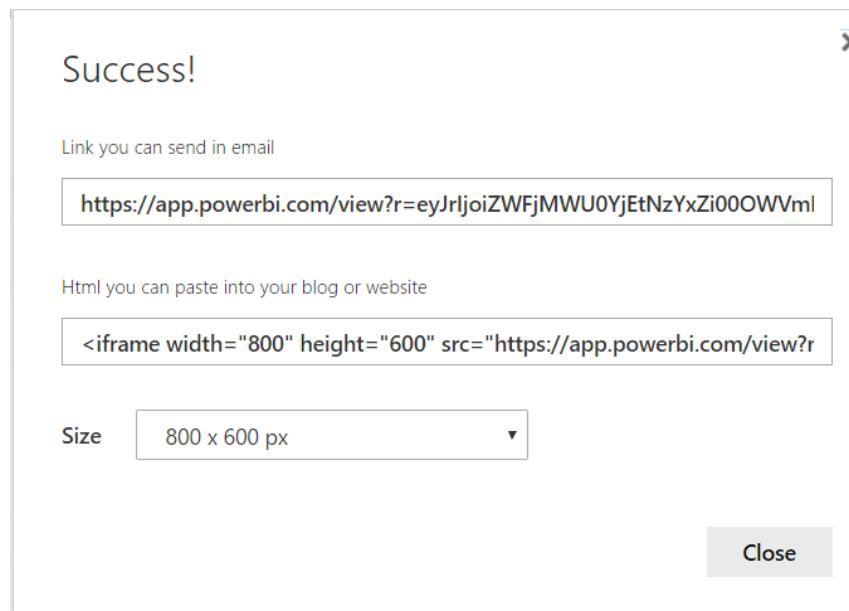
- Down load image from
- <https://github.com/laploy/ML/blob/master/ship.JPG>
- Click report blank area
- Click format button
- Click Page Background
- Add picture
- Set Transparent to 95%
- Image fit = Fit



## Publish Report

### Publish Report

- Click File/Save report: name = mlreport1
- Click File/Publish to web
- Click agree
- Copy link and paste into browser to view report



## More information

Free eBook: Introducing Microsoft Power BI

[https://blogs.msdn.microsoft.com/microsoft\\_press/2016/06/16/free-ebook-introducing-microsoft-power-bi/](https://blogs.msdn.microsoft.com/microsoft_press/2016/06/16/free-ebook-introducing-microsoft-power-bi/)



# ALGORITHM ANOMALY DETECTION



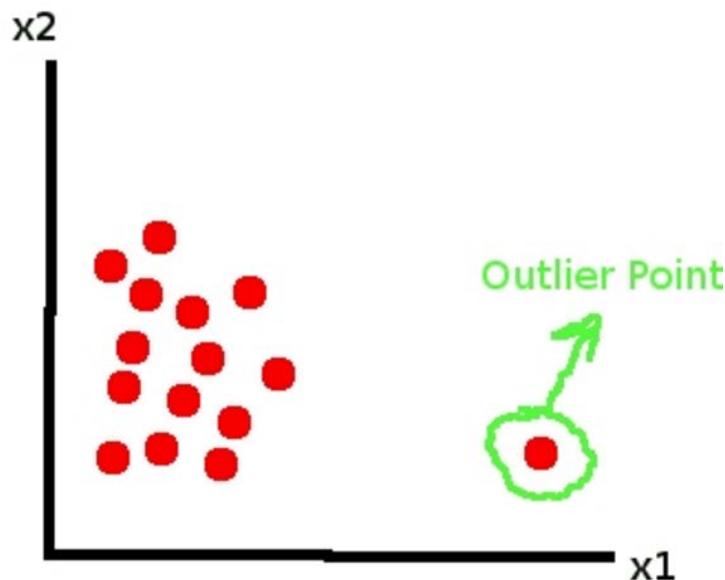
## In this session

- Anomaly Detection
- One-Class SVM Algorithm
- PCA-Based Algorithm
- Data set
- Data attribute
- Experiment Steps

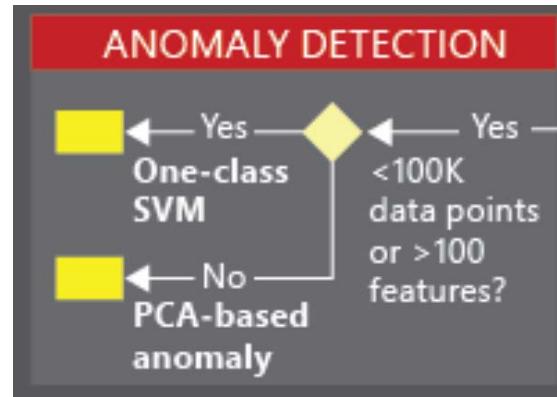
## Anomaly Detection

### Anomaly Detection

- Credit card fraud, transaction, medical, text etc.
- Also referred to as outliers, novelties, noise, deviations and exceptions
- The data consists of 'normal' applications and 'risky' applications
- Risky transactions = anomalous



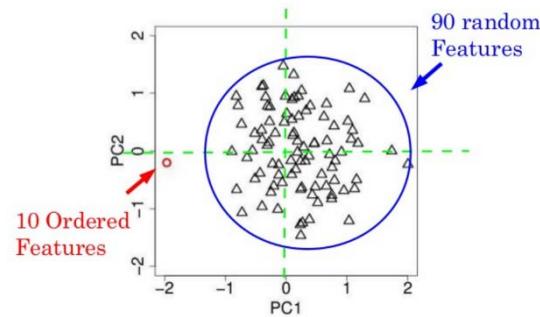
## One-Class SVM



## One-Class SVM

- SVM = Support Vector Model
- Supervised learning models
- Analyze data and recognize patterns
- Have a lot of "normal" data and not many cases of the anomalies
- Use with Train Anomaly Detection Model
- The train data set contain all or mostly normal cases.

## PCA-Based

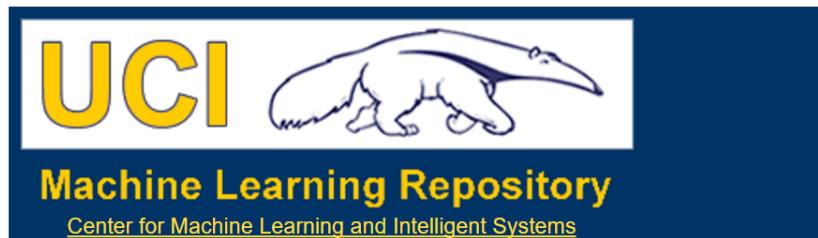


### PCA-Based Anomaly Detection module

- Principal Component Analysis (PCA)
- Use when easy to obtain training data from one class
- One class = acceptable transactions
- Use when difficult to obtain sufficient samples of the targeted anomalies
- Detect fraudulent transaction
- You might not have enough examples of fraud to train the mode
- But have many examples of good transactions

## Data set

[https://archive.ics.uci.edu/ml/datasets/Statlog+\(German+Credit+Data\)](https://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data))



### Statlog (German Credit Data) Data Set

Download: [Data Folder](#), [Data Set Description](#)

Quelle:

Professor Dr. Hans Hofmann  
Institut für Statistik und "Okonometrie  
Universität in Hamburg  
FB Wirtschaftswissenschaften  
Von-Melle-Park 5  
2000 Hamburg 13

### German credit dataset

- Credit card application
- 1000 instances (rows)
- Attributes = 20 (7 numerical, 13 categorical)
- Label 1 = normal, 2 = risky

## Data attribute

A11 6 A34 A43 1169 A65 A75 4 A93 A101 4 A121 67 A143 A152 2 A173 1 A192 A201 1  
A12 48 A32 A43 5951 A61 A73 2 A92 A101 2 A121 22 A143 A152 1 A173 1 A191 A201 2  
A14 12 A34 A46 2096 A61 A74 2 A93 A101 3 A121 49 A143 A152 1 A172 2 A191 A201 1  
A11 42 A32 A42 7882 A61 A74 2 A93 A103 4 A122 45 A143 A153 1 A173 2 A191 A201 1  
A11 24 A33 A40 4870 A61 A73 3 A93 A101 4 A124 53 A143 A153 2 A173 2 A191 A201 2  
A14 36 A32 A46 9055 A65 A73 2 A93 A101 4 A124 35 A143 A153 1 A172 2 A192 A201 1  
A14 24 A32 A42 2835 A63 A75 3 A93 A101 4 A122 53 A143 A152 1 A173 1 A191 A201 1

Attribute: Account status, month, credit history, propose, amount, saving, employ since, installment rate, sex ...

Attribute 9: (qualitative)

Personal status and sex

A91 : male : divorced/separated

A92 : female : divorced/separated/married

A93 : male : single

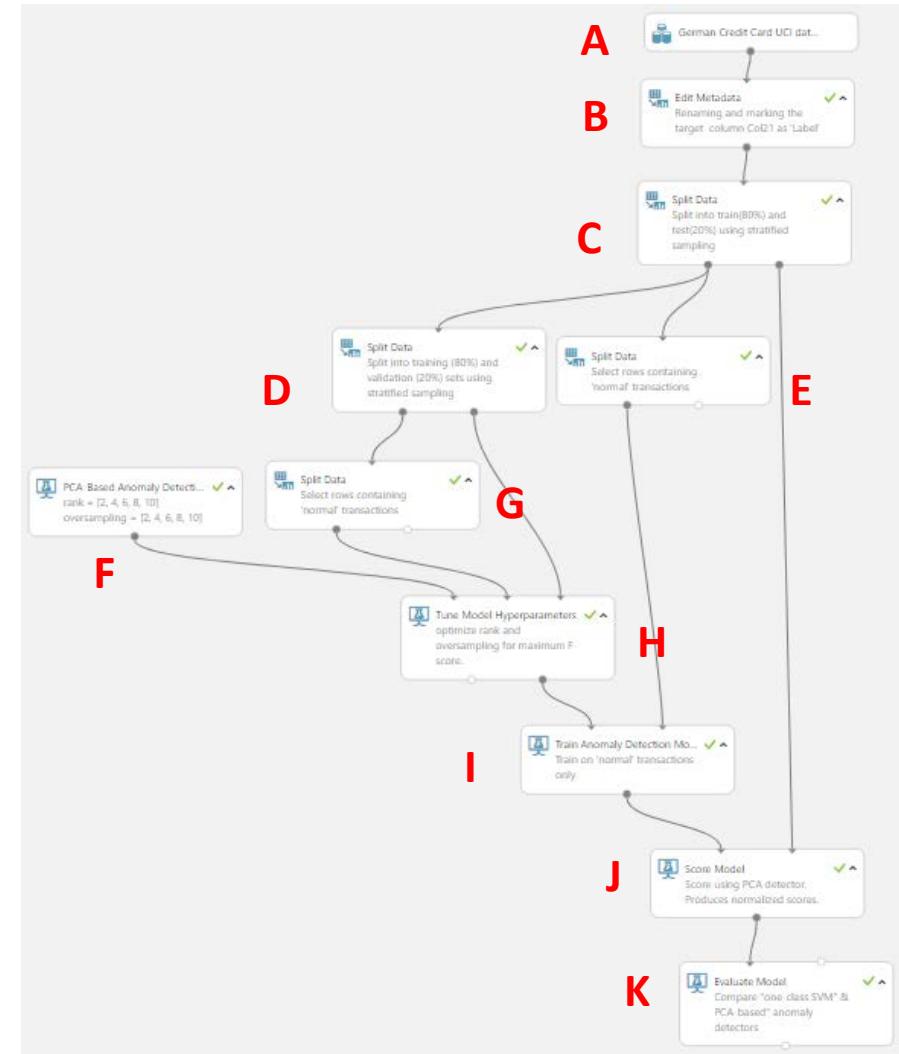
A94 : male : married/widowed

A95 : female : single

## Experiment Steps

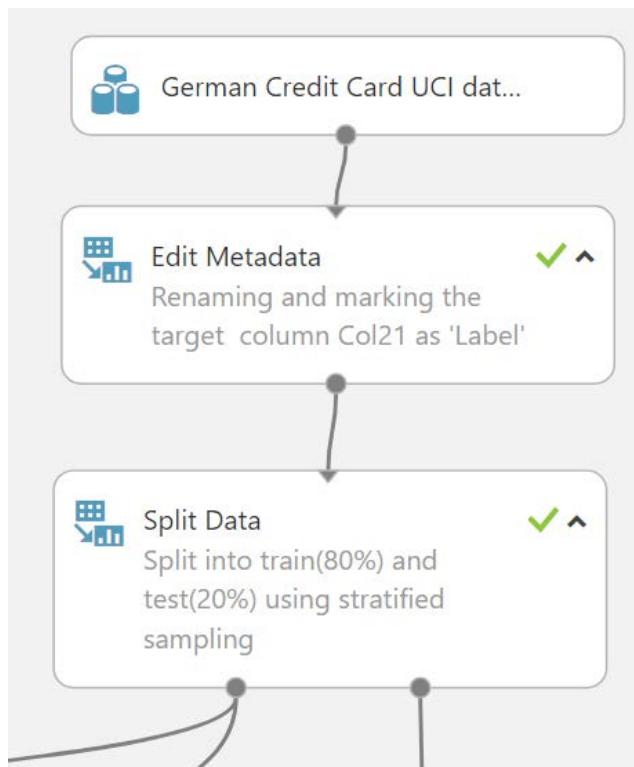
### Experiment steps

1. Import data set
2. Edit metadata
3. Split data for training
4. Split data for Score
5. Add PCA Base method
6. Add Tune Model Hyper parameters
7. Add Train Anomaly Detection Model
8. Add Score model
9. Add Evaluate Model



## Experiment Steps

- A. Import data set
- B. Edit metadata
- C. Split data for training



### Edit Metadata

Column

**Selected columns:**  
**Column names:** Col21

Launch column selector

Data type

Unchanged

Categorical

Unchanged

Fields

Label

New column names

Label

### Split Data

Splitting mode

Split Rows

Fraction of rows in the fir...

0.75

Randomized split

Random seed

0

Stratified split

True

Stratification key column

**Selected columns:**  
**Column names:** Label

Launch column selector

## Experiment Steps

### Add 4 Split data models and PCA Based

▲ Split Data

Splitting mode  
Split Rows

Fraction of rows in the first split  
0.75

Randomized split

Random seed  
0

Stratified split  
True

Stratification key column  
**Selected columns:**  
**Column names:** Label

Launch column selector

▲ Split Data

Splitting mode  
Regular Expression

Regular expression  
\"Label" ^1

▲ Split Data

Splitting mode  
Regular Expression

Regular expression  
\"Label" ^1

## Experiment Steps

### Training mode

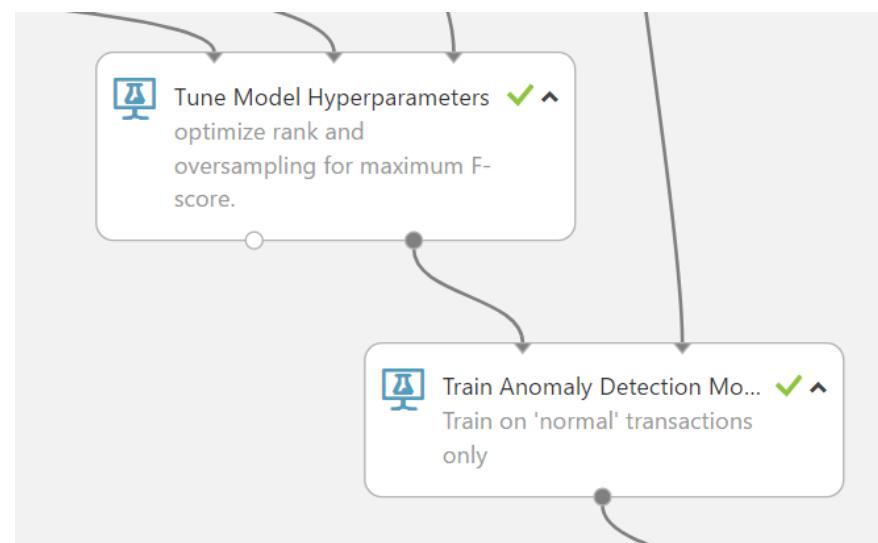
- Single Parameter: If you know how you want to configure the model, you can provide a specific set of values as arguments. You might have learned these values by experimentation or received them as guidance.
- Parameter Range: If you are not sure of the best parameters, you can find the optimal parameters by specifying multiple values and using a parameter sweep to find the optimal configuration.

The screenshot shows the Azure Machine Learning studio interface. On the left, there is a preview pane for the 'PCA-Based Anomaly Detecti...' component. It displays two parameters: 'rank = [2, 4, 6, 8, 10]' and 'oversampling = [2, 4, 6, 8, 10]'. On the right, the main workspace shows the configuration for this component. Under 'Training mode', it is set to 'Parameter Range'. Below this, there are two sections for 'Range for number of PCA c...' and 'Range for the oversampling...'. Both sections show a checkbox for 'Use Range Builder' followed by a list box containing the values '2, 4, 6, 8, 10'. At the bottom, there is another checkbox for 'Enable input feature me...'.

## Experiment Steps

H. Add Tune Model Hyperparameters

I. Add Train Anomaly Detection Model



### Tune Model Hyperparameters

Specify parameter sweeping mo...

Entire grid ▾

Label column

**Selected columns:**  
All labels

Launch column selector

Metric for measuring perfor...

F-score ▾

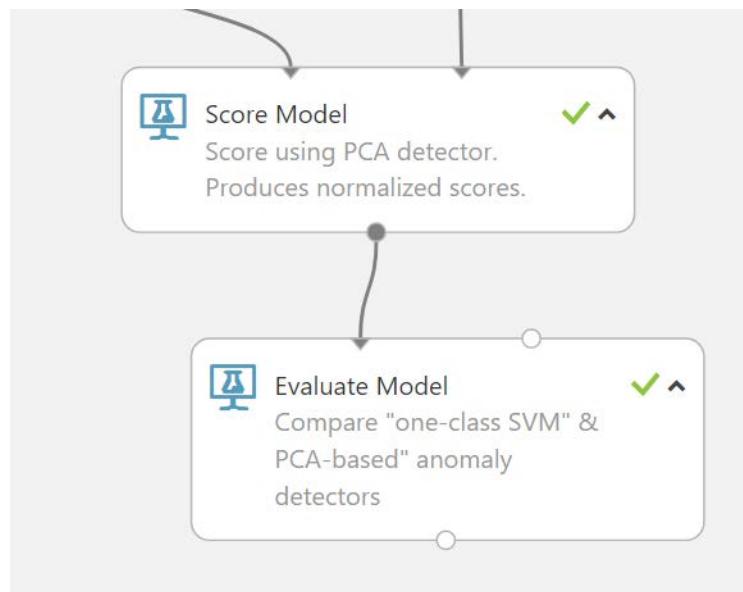
Metric for measuring perfor...

Mean absolute error ▾

## Experiment Steps

J. Add Score Model

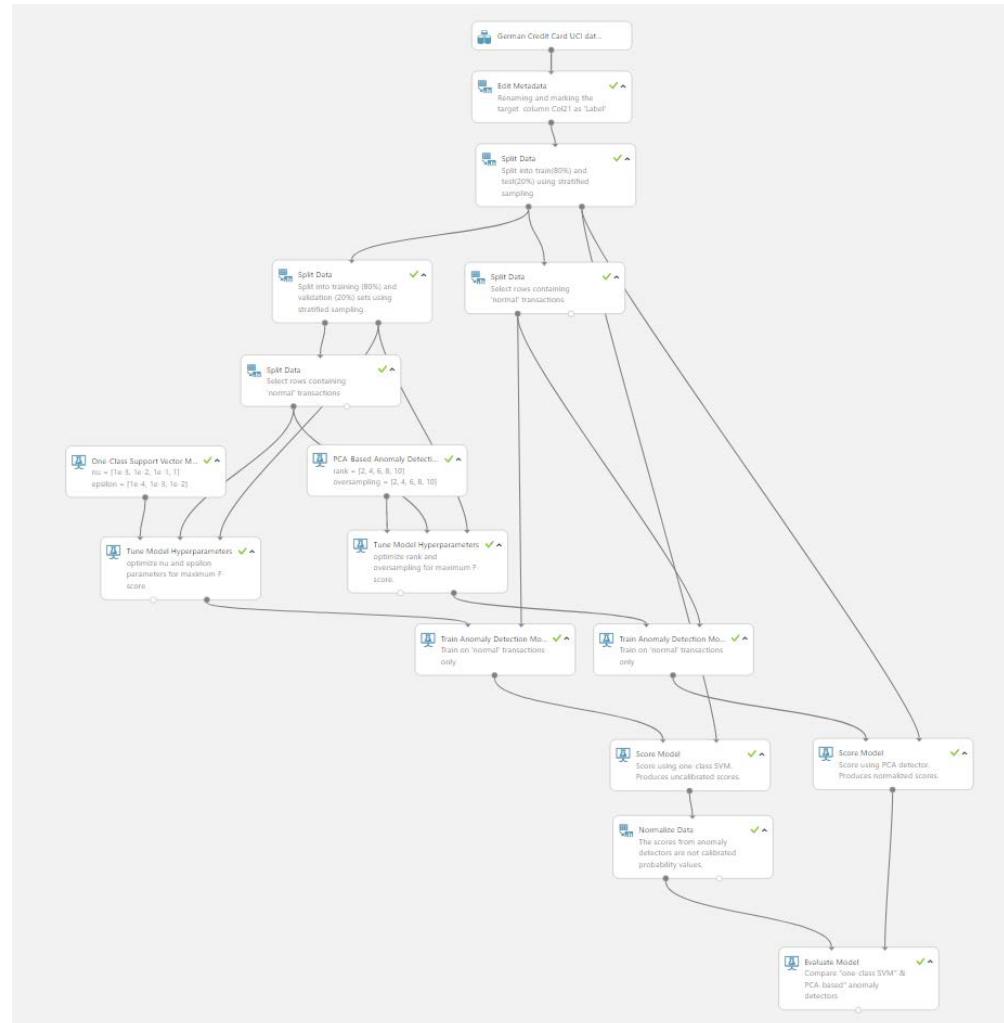
K. Add Evaluate Model



### Score Model

|                                                              |                                               |
|--------------------------------------------------------------|-----------------------------------------------|
| <input checked="" type="checkbox"/> Append score columns ... |                                               |
| START TIME                                                   | 6/25/2017 12...                               |
| END TIME                                                     | 6/25/2017 12...                               |
| ELAPSED TIME                                                 | 0:00:00.000                                   |
| STATUS CODE                                                  | Finished                                      |
| STATUS DETAILS                                               | Task output<br>was present in<br>output cache |

## Compare two anomaly algorithm



## More Information

PCA-Based Anomaly Detection

<https://msdn.microsoft.com/en-us/library/azure/dn913102.aspx>

This Experiment

<https://gallery.cortanaintelligence.com/Experiment/Anomaly-Detection-9>

Anomaly compare

<https://gallery.cortanaintelligence.com/Experiment/Anomaly-compare>

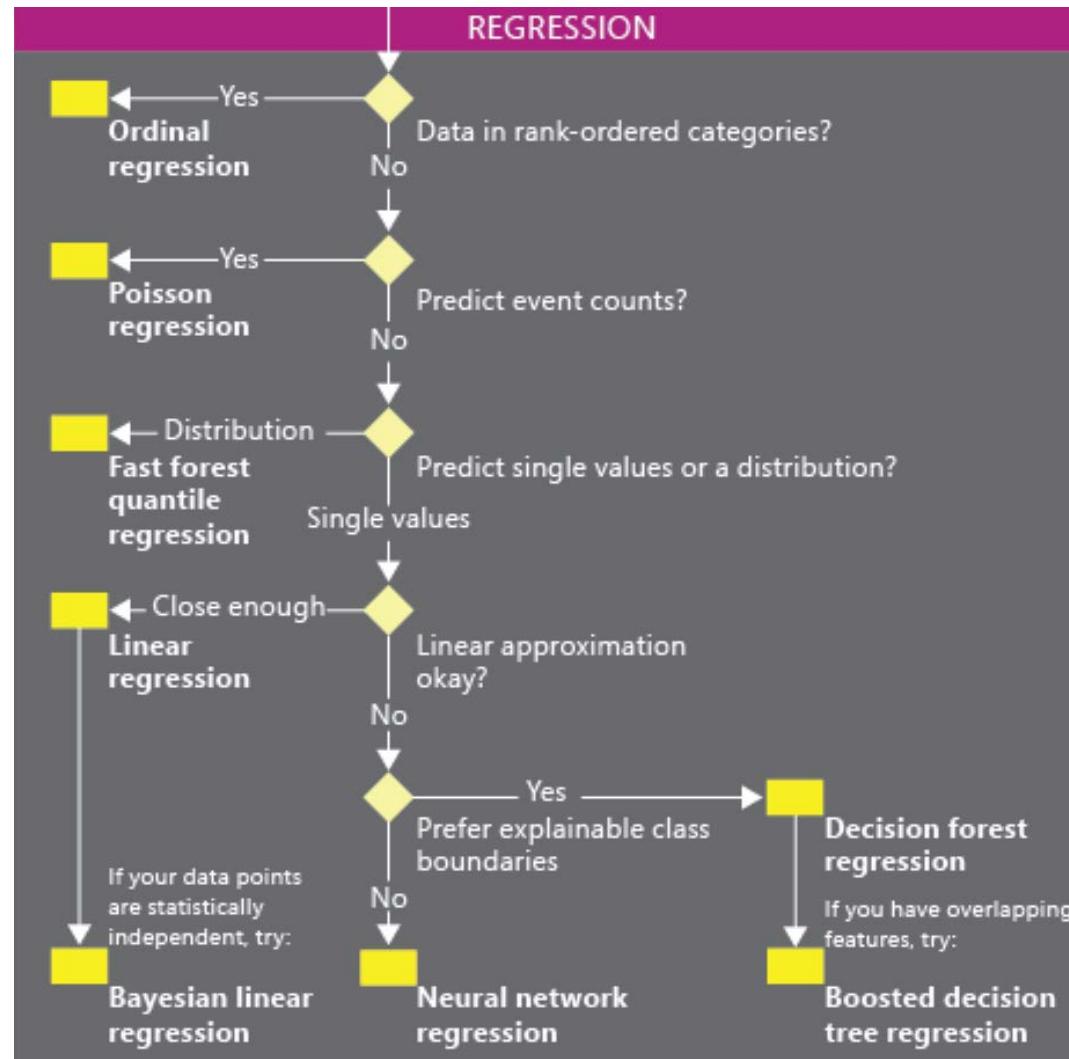
# ALGORITHM REGRESSION



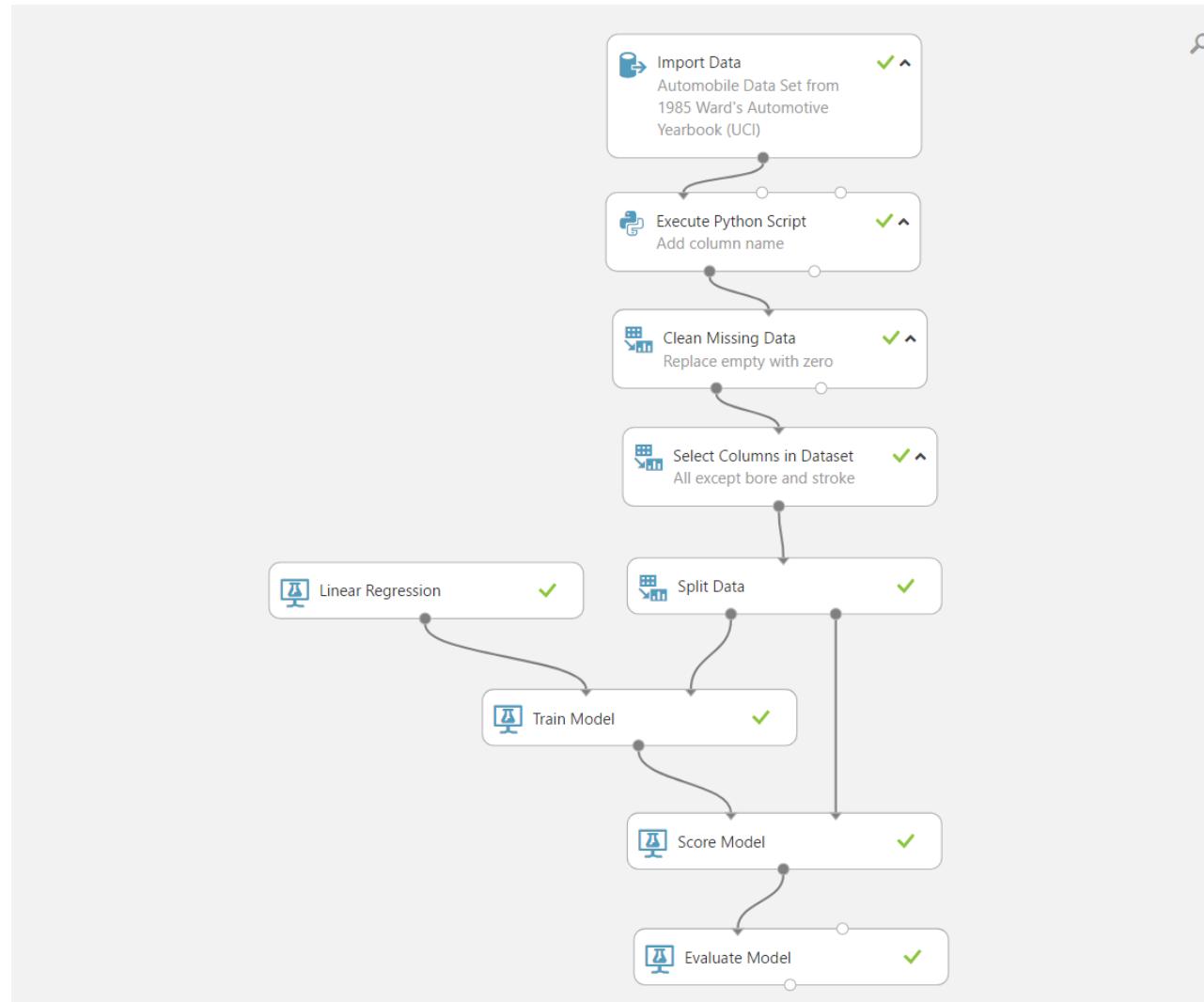
## In this session

- Regression Algorithms in Azure ML
- Create New ML Experiment
- Import auto data from UCI
- Add column name using python
- Clean missing data
- Select column (exclude column)
- Split data
- Add Linear Regression module
- Add Train Model
- Add Score Model
- Add Evaluate Model

## Regression Algorithms in Azure ML



## Over view



## Working steps

### Working Steps

1. Create New ML Experiment
2. Import auto data from UCI
3. Add column name using python
4. Clean missing data
5. Select column (exclude column)
6. Split data
7. Add Linear Regression module
8. Add Train Model
9. Add Score Model
10. Add Evaluate Model

## Data set



### Automobile Data Set

*Download:* [Data Folder](#), [Data Set Description](#)

*Abstract:* From 1985 Ward's Automotive Yearbook



Home

<https://archive.ics.uci.edu/ml/datasets/Automobile>

Data download

<https://archive.ics.uci.edu/ml/machine-learning-databases/autos/imports-85.data>

Element column names/values

<https://archive.ics.uci.edu/ml/machine-learning-databases/autos/imports-85.names>

## Data attribute

1. symboling: -3, -2, -1, 0, 1, 2, 3
2. normalized-losses: continuous from 65 to 256
3. make: alfa-romero, audi, bmw, chevrolet, dodge, honda, isuzu, jaguar, mazda, mercedes-benz, mercury, mitsubishi, nissan, peugot, plymouth, porsche, renault, saab, subaru, toyota, volkswagen, volvo
4. fuel-type: diesel, gas
5. aspiration: std, turbo
6. num-of-doors: four, two
7. body-style: hardtop, wagon, sedan, hatchback, convertible
8. drive-wheels: 4wd, fwd, rwd
9. engine-location: front, rear
10. wheel-base: continuous from 86.6 120.9
11. length: continuous from 141.1 to 208.1
12. width: continuous from 60.3 to 72.3
13. height: continuous from 47.8 to 59.8

## Data attribute

14. curb-weight: continuous from 1488 to 4066
15. engine-type: dohc, dohcv, l, ohc, ohcf, ohcv, rotor
16. num-of-cylinders: eight, five, four, six, three, twelve, two
17. engine-size: continuous from 61 to 326
18. fuel-system: 1bbl, 2bbl, 4bbl, idi, mfi, mpfi, spdi, spfi
19. bore: continuous from 2.54 to 3.94
20. stroke: continuous from 2.07 to 4.17
21. compression-ratio: continuous from 7 to 23
22. horsepower: continuous from 48 to 288
23. peak-rpm: continuous from 4150 to 6600
24. city-mpg: continuous from 13 to 49
25. highway-mpg: continuous from 16 to 54
26. price: continuous from 5118 to 45400

## Add column name Python Script

```
1 #import pandas as pd
2 def azureml_main(dataframe1 = None, dataframe2 = None):
3 dataframe1.columns = [
4 'symboling',
5 'normalized-losses',
6 'make',
7 'fuel-type',
8 'aspiration',
9 'num-of-doors',
10 'body-style',
11 'drive-wheels',
12 'engine-location',
13 'wheel-base',
14 'length',
15 'width',
16 'height',
17 'curb-weight',
18 'engine-type',
19 'num-of-cylinders',
20 'engine-size',
21 'fuel-system',
22 'bore',
23 'stroke',
24 'compression-ratio',
25 'horsepower',
26 'peak-rpm',
27 'city-mpg',
28 'highway-mpg',
29 'price'
30]
31 return dataframe1,
```

## Clean missing data

### ◀ Clean Missing Data

Columns to be cleaned

**Selected columns:**

All columns

Launch column selector

Minimum missing value ra...

0

Maximum missing value ra...

1

Cleaning mode

Custom substitution value ▾

Replacement value

0

Generate missing valu...

## Select (exclude) column

Select columns x

BY NAME  Allow duplicates and preserve column order in selection

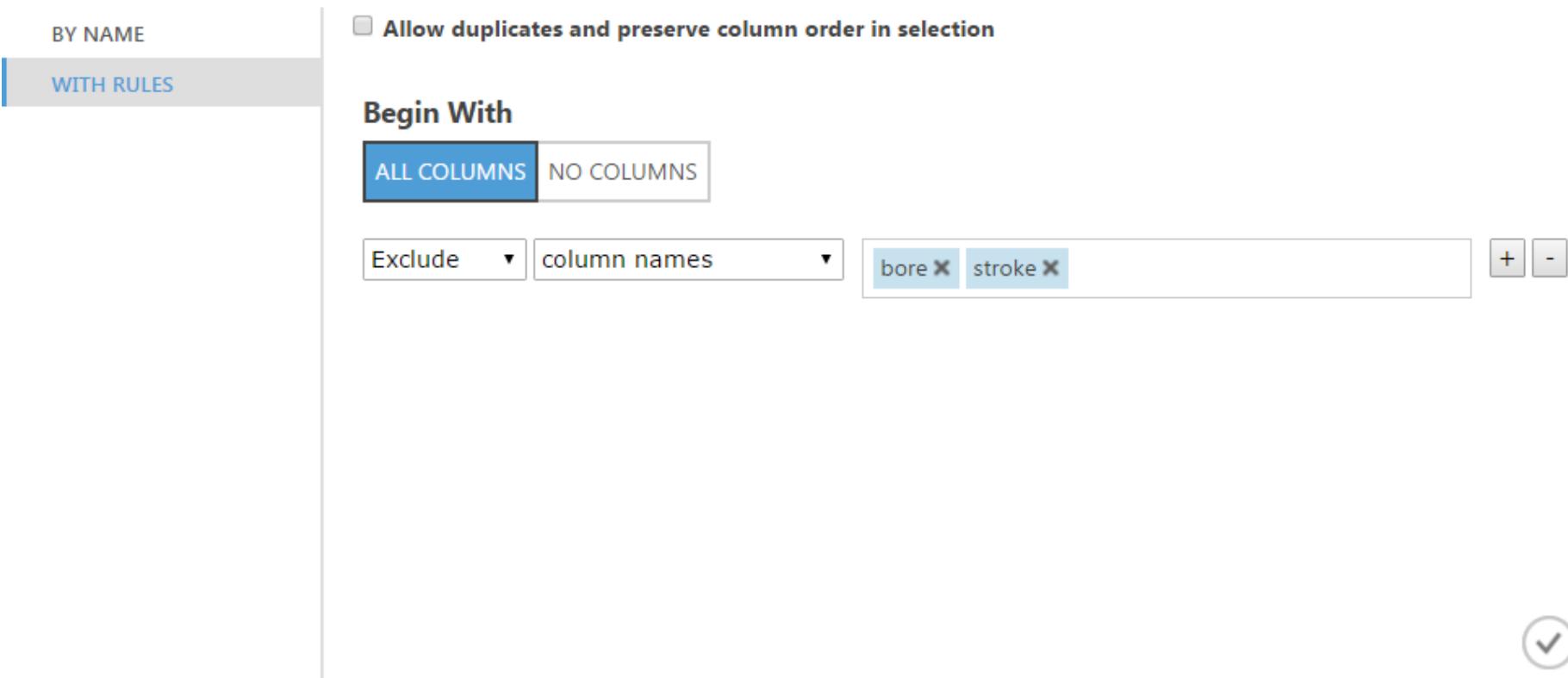
**WITH RULES**

**Begin With**

**ALL COLUMNS** **NO COLUMNS**

Exclude ▾ column names ▾ bore × stroke × + -

✓



## Split data

### ▲ Split Data

Splitting mode

Split Rows

Fraction of rows in the first...

0.8

Randomized split

Random seed

0

Stratified split

False

## Evaluation Metrics

- Add Linear Regression module
- Add Train Model
- Add Score Model
- Add Evaluate Model
- Run
- Inspect Score
- Inspect Evaluate metrics

▲ Linear Regression

Solution method  
Ordinary Least Squares ▾

L2 regularization weight  
0.001

Include intercept term

Random number seed

Allow unknown catego...

▲ Train Model

Label column

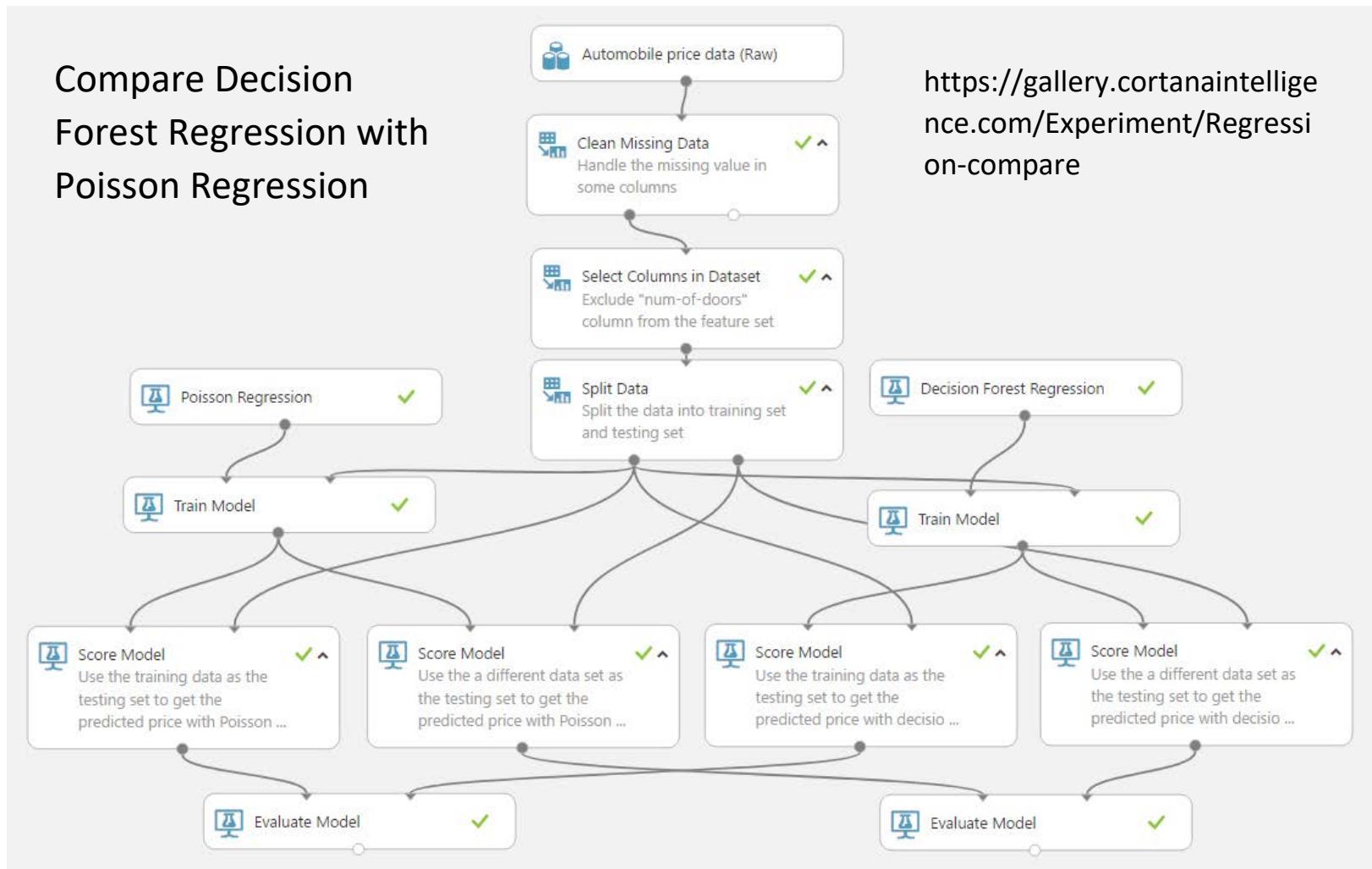
Selected columns:  
Column names: price

Launch column selector

▲ Metrics

|                              |             |
|------------------------------|-------------|
| Mean Absolute Error          | 2728.117473 |
| Root Mean Squared Error      | 4130.273478 |
| Relative Absolute Error      | 0.388729    |
| Relative Squared Error       | 0.200384    |
| Coefficient of Determination | 0.799616    |

## Regression compare



## More information

Using linear regression in Azure Machine Learning

<https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-linear-regression-in-azure>

This Experiment

<https://gallery.cortanaintelligence.com/Experiment/Regression-3>

Regression compare

<https://gallery.cortanaintelligence.com/Experiment/Regression-compare>

# ALGORITHM CLUSTER

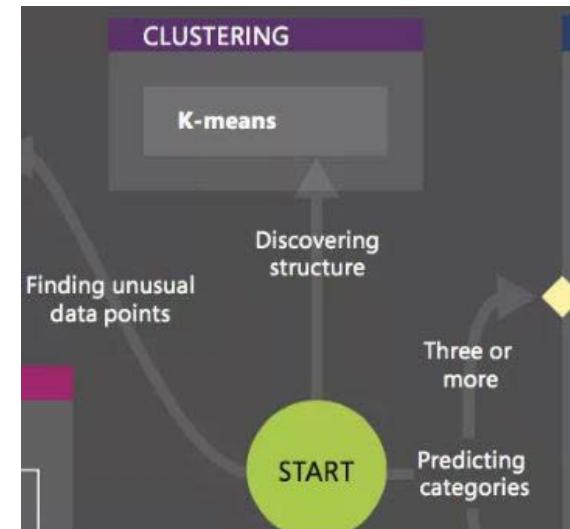


## In this session

- Cluster Algorithms in Azure ML
- Model overview
- Dataset
- Feature Hashing module
- Train
- Edit Metadata

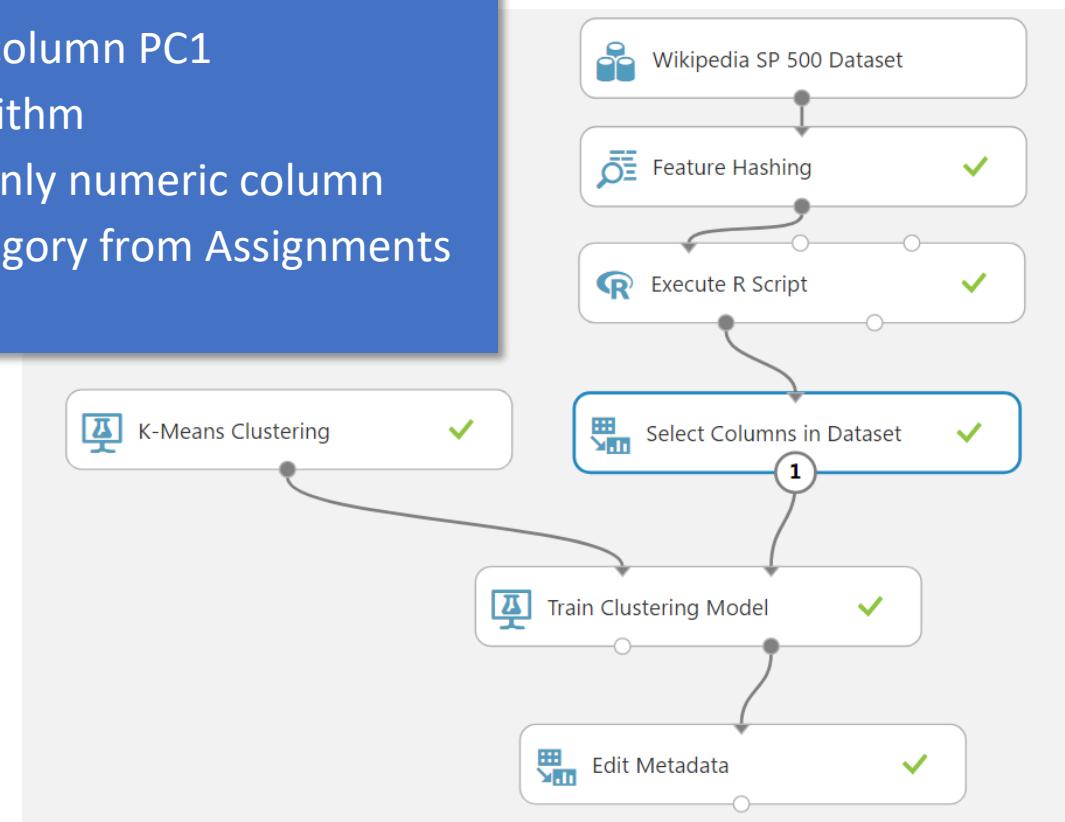
## Cluster Algorithms in Azure ML

- Uses iterative techniques
- Group cases in a dataset into clusters
- Contain similar characteristics
- Useful for exploring data
- Identifying anomalies in the data
- Making predictions
- Identify relationships in a dataset
- Not logically derive by browsing or simple observation
- Used in the early phases of machine learning tasks
- Explore the data and discover unexpected correlations
- Only algorithm in AML that is Unsupervised



## Model overview

- Dataset: Wikipedia SP 500 Dataset
- Feature Hashing: create feature from column Text
- R Script: reduce feature to 10 columns
- Select Columns: exclude column PC1
- K-Means: clustering algorithm
- Train: train model using only numeric column
- Edit Metadata: make category from Assignments column



## Dataset

rows

466

columns

3

|         | Title                                                                                                                                                               | Category                                                                            | Text                                                                                                                                                                                                             |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| view as |   |  |                                                                                                                               |
|         | Apple Inc.                                                                                                                                                          | Information Technology                                                              | nasdaq 100 component s<br>p 500 component<br>foundation founder<br>location city apple campus<br>1 infinite loop street infinite<br>loop cupertino california<br>cupertino california<br>location country united |

### Pre-processed outside Azure ML Studio

- Removing wiki formatting
- Removing non-alphanumeric characters
- Converting all text to lowercase
- Adding company categories, where known

## Feature Hashing module

### Feature Hashing module

- Tokenizes the text string
- Transforms the data into a series of numbers
- Based on the hash value of each token

#### ▲ Feature Hashing

Target column(s)

**Selected columns:**

**Column names:** Text

[Launch column selector](#)

Hashing bitsize

12

N-grams

1

rows  
466      columns  
4099

|  | Title | Category | Text | Text_HashingFeature_1 | Text_HashingFeature_2 |
|--|-------|----------|------|-----------------------|-----------------------|
|--|-------|----------|------|-----------------------|-----------------------|

view as



nasdaq 100  
component s p  
500  
component

## R Script

### R Script

```
1 dataset1 <- maml.mapInputPort(1)
2 titles_categories = dataset1[,1:2]
3 pca = prcomp(dataset1[,4:4099])
4 top_pca_scores = data.frame(pca$x[,1:10])
5 data.set = cbind(titles_categories,top_pca_scores)
6 maml.mapOutputPort("data.set");
```

- Dimensionality of the data from hashing is too high (4K)
- Cannot be used by the K-Means clustering algorithm directly
- Principal Component Analysis (PCA) was applied using a custom R script
- Reduce the dimensionality to 10 variables
- View the result = double-clicking the right-hand output of the Execute R Script

### Execute R Script

#### R Script

```
1 dataset1 <- maml.mapInputPort(1)
2 titles_categories = dataset1[,1:2]
3 pca = prcomp(dataset1[,4:4099])
4 top_pca_scores = data.frame(pca$x[,1:10])
5 data.set = cbind(titles_categories,top_pca_scores)
6 maml.mapOutputPort("data.set");
```

#### Random Seed

#### R Version

- Select Columns in Dataset
- K-Means Clustering

<https://msdn.microsoft.com/en-us/library/azure/dn905944.aspx>

#### ▲ Select Columns in Dataset

Select columns

**Selected columns:**

All columns

**Exclude column names:** PC1

Launch column selector

#### ▲ K-Means Clustering

Create trainer mode

Single Parameter ▾

Number of Centroids

3

Initialization

K-Means++ Fast ▾

Random number seed

7654

Metric

Cosine ▾

Iterations

100

Assign Label Mode

Ignore label column ▾

- Train Clustering Model
- Edit Metadata

#### ▲ Train Clustering Model

Column Set

**Selected columns:**

**Column type:** Numeric, All

Launch column selector

Check for Append or Uncheck f... 

#### ▲ Edit Metadata

Column

**Selected columns:**

**Column names:** Assignments

Launch column selector

Data type

Unchanged 

Categorical 

Make categorical 

Fields 

Unchanged 

New column names 

## More information

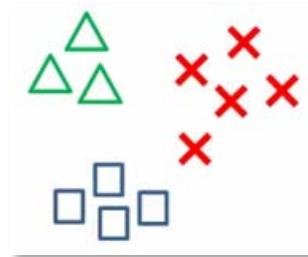
### K-Means Clustering

<https://msdn.microsoft.com/en-us/library/azure/dn905944.aspx>

### This Experiment

<https://gallery.cortanaintelligence.com/Experiment/Clustering-K-Means-basic>

# ALGORITHM MULTI CLASS

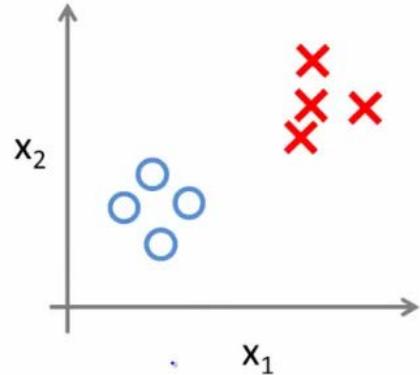


## In this session

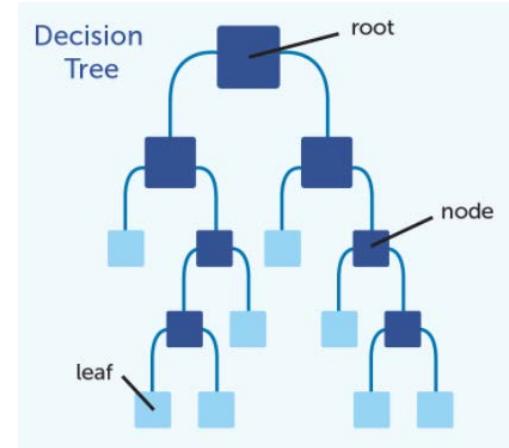
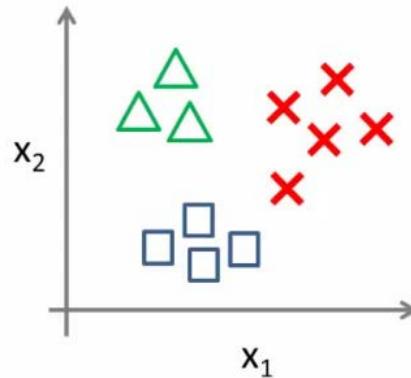
- Multi Class Algorithms in Azure ML
- Data importing and engineering
- Feature engineering
- Modeling and evaluation
- Reuter Data set
- Edit Metadata
- Confusion Matrix

## Multi Class Algorithms in Azure ML

Binary classification:



Multi-class classification:

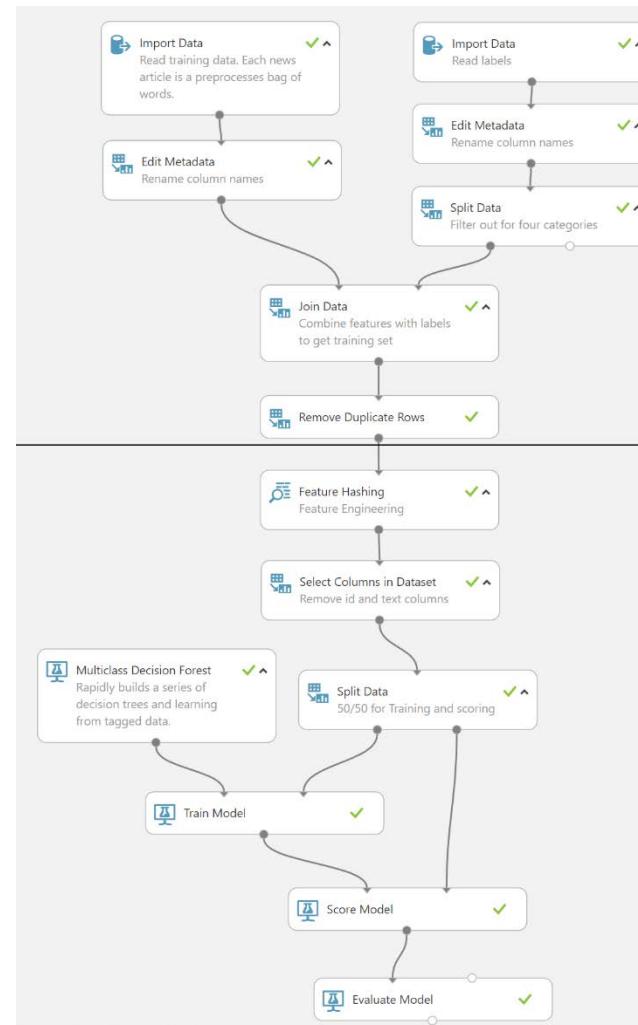


## Multiclass Decision Forest

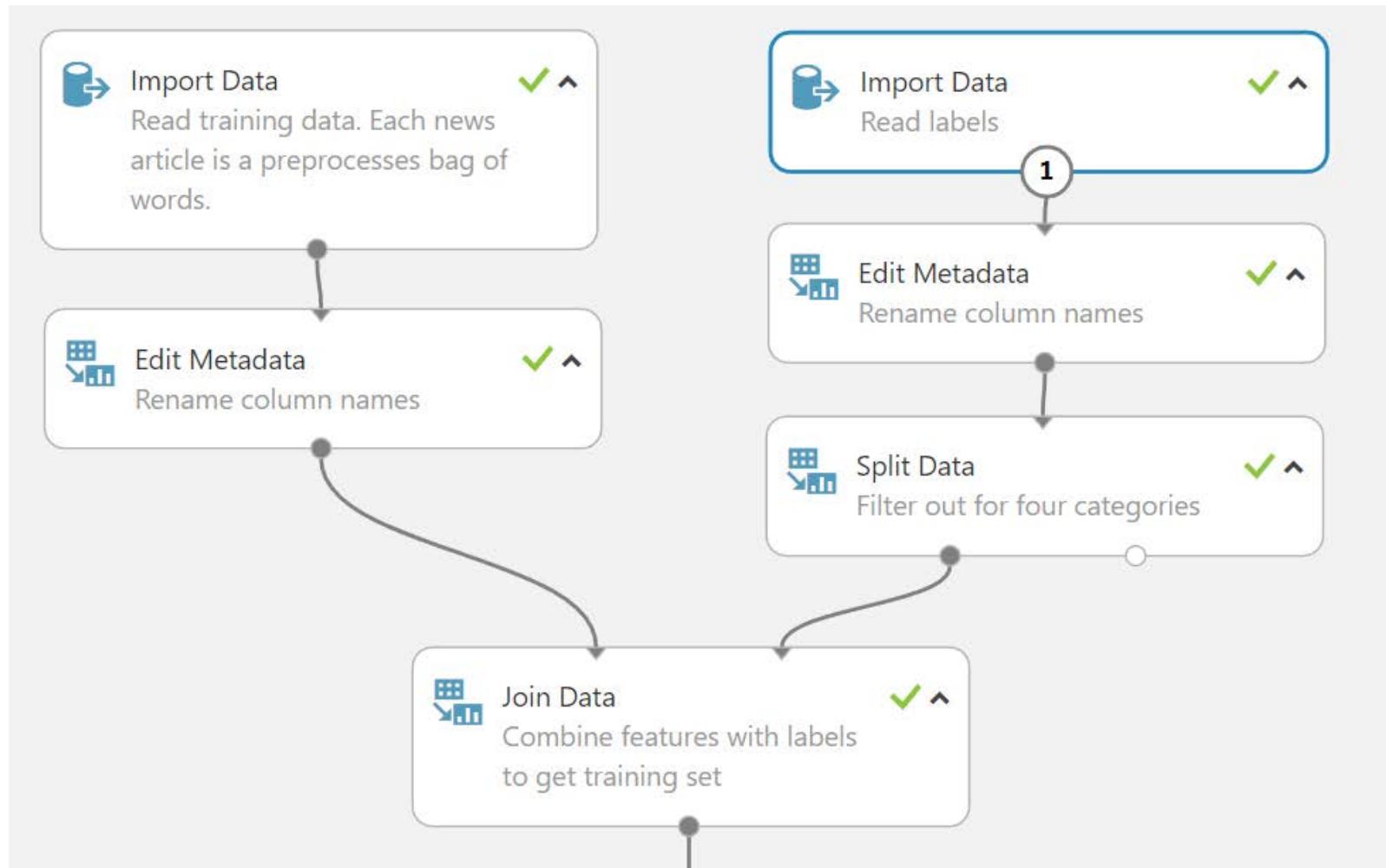
- Based on the decision forest algorithm
- Rapidly builds a series of decision trees
- learning from tagged data.
- Voting on the most popular output class
- Voting is a form of aggregation

## Over all Experiment

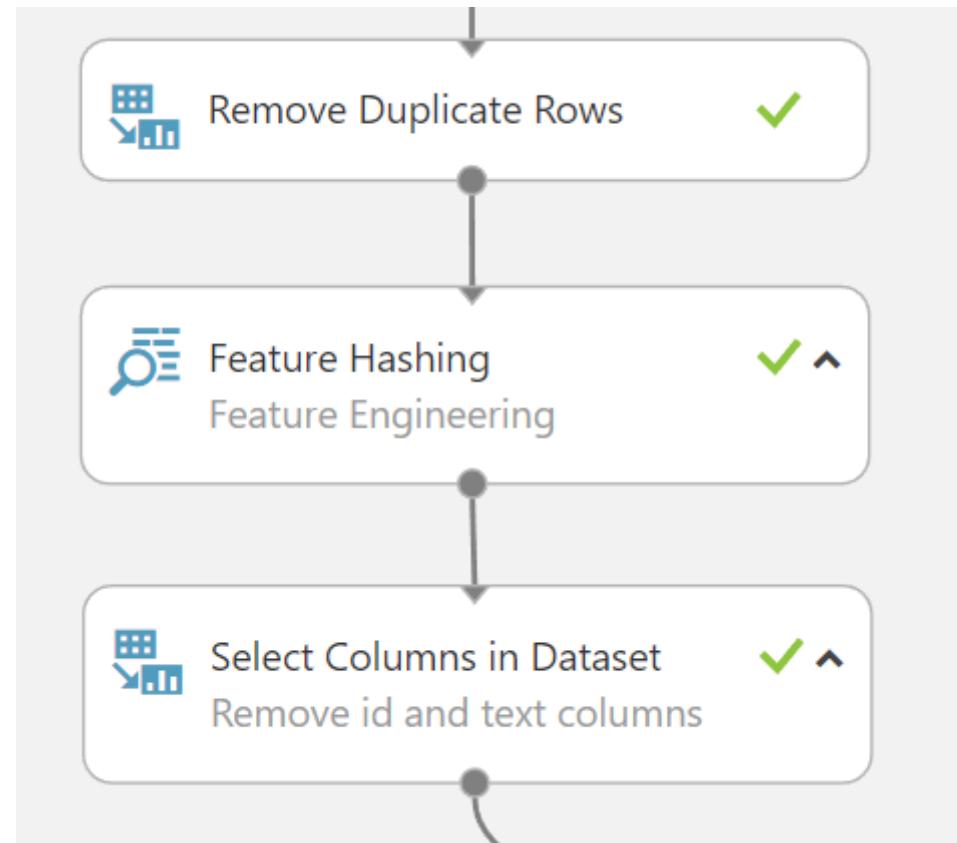
- multiclass classifiers
- Feature engineering using hashing
- Classify news into four categories



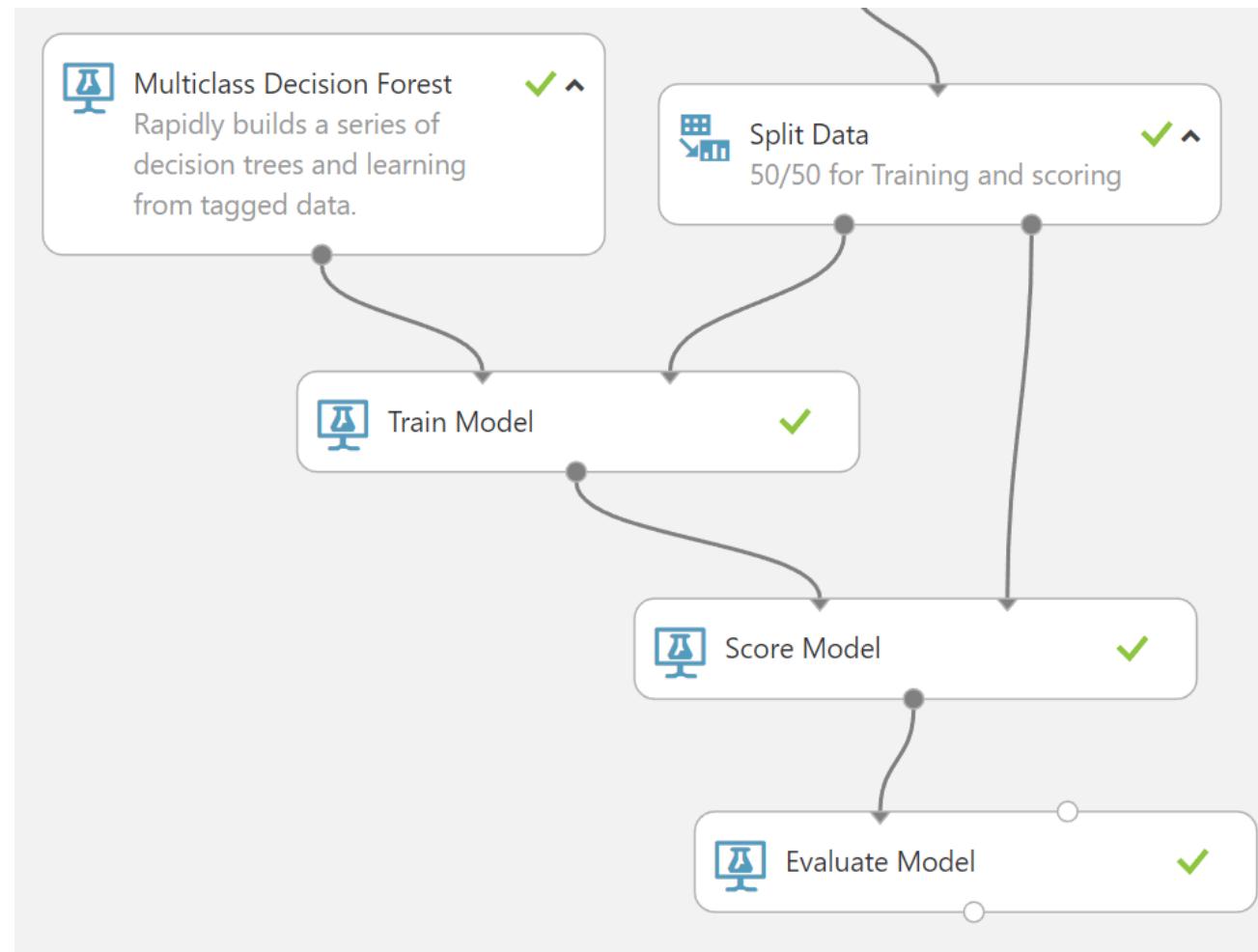
## Data importing and engineering



## Feature engineering



## Modeling and evaluation



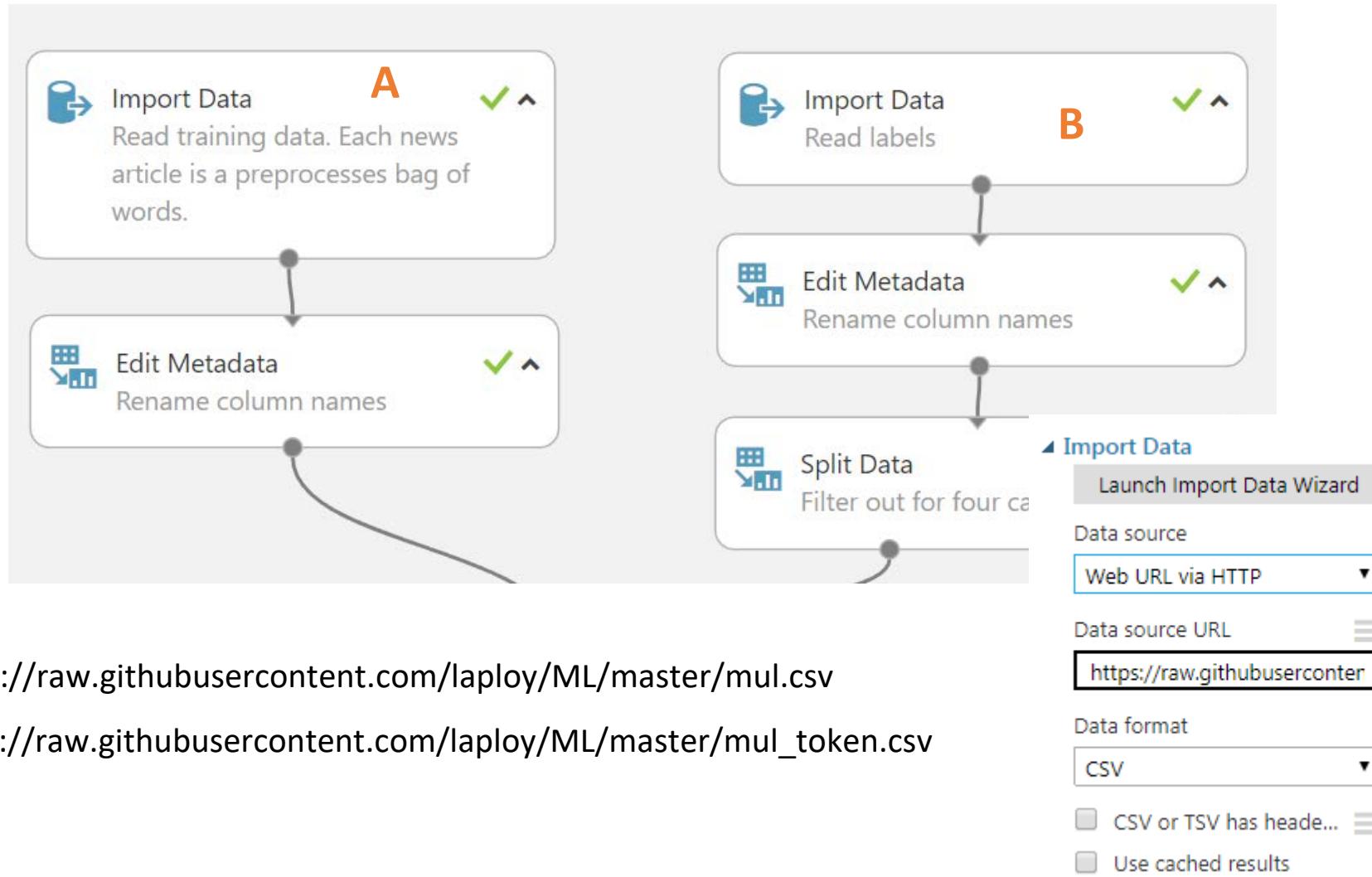
## Reuter Data set

|    | A     | B                                                                                                               | C | D | E | F | G | H | I | J | K |
|----|-------|-----------------------------------------------------------------------------------------------------------------|---|---|---|---|---|---|---|---|---|
| 1  | 26151 | socc colomb colomb colomb beat beat chil chil chil chil world world world cup cup cup qualif qualif             |   |   |   |   |   |   |   |   |   |
| 2  | 26152 | world world world world qualif qualif sunday minut minut won hold athlet time time time time lucky              |   |   |   |   |   |   |   |   |   |
| 3  | 26153 | hernandez socc jess colomb valient beat beat gabriel gabriel chil urdanet world world vera cup cup cu           |   |   |   |   |   |   |   |   |   |
| 4  | 26154 | beat beat beat world world world world cup americ match cycl cycl sunday rousseau rousseau r                    |   |   |   |   |   |   |   |   |   |
| 5  | 26155 | world world sunday day stand stand point point race race final final franc franc champ champ result res         |   |   |   |   |   |   |   |   |   |
| 6  | 26156 | open nick knight knight hit hit hundr provid backbon inn maid centur centur recent match match mat              |   |   |   |   |   |   |   |   |   |
| 7  | 26157 | knight beat beat sunday scor day day won final final ijaz ahm sery intern win cricket cricket pakist pakist pak |   |   |   |   |   |   |   |   |   |
| 8  | 26158 | world sunday minut put athlet komen record daniel keny keny keny keny keny keny riet men men me                 |   |   |   |   |   |   |   |   |   |
| 9  | 26159 | world world sunday minut week final athlet time time komen komen komen komen komen break break br               |   |   |   |   |   |   |   |   |   |
| 10 | 26160 | world sunday minut athlet komen komen break record record daniel keny men noureddin alger morcel met            |   |   |   |   |   |   |   |   |   |
| 11 | 26161 | knight match sunday bat fall fall day won won won asif asif mujtab mujtab saeed anwar anwar shahid shahi        |   |   |   |   |   |   |   |   |   |
| 12 | 26162 | pilsudsk pilsudsk outsid quart quart length length gross prei von baden baden baden half equal delight brav     |   |   |   |   |   |   |   |   |   |
| 13 | 26163 | pilsudsk quart quart quart length length gross prei von baden baden baden sunday frankie dettor sui             |   |   |   |   |   |   |   |   |   |
| 14 | 26164 | colomb beat beat beat beat beat world world juan match cycl cycl rousseau rousseau sunday flo                   |   |   |   |   |   |   |   |   |   |

|    | A    | B    |
|----|------|------|
| 1  | E11  | 2286 |
| 2  | ECAT | 2286 |
| 3  | M11  | 2286 |
| 4  | M12  | 2286 |
| 5  | MCAT | 2286 |
| 6  | C24  | 2287 |
| 7  | CCAT | 2287 |
| 8  | C151 | 2288 |
| 9  | C15  | 2288 |
| 10 | CCAT | 2288 |
| 11 | E41  | 2288 |
| 12 | ECAT | 2288 |
| 13 | GCAT | 2288 |

- 2004 Reuters news dataset
- 10,000 News examples
- 5K Training / 5K Scoring

## Import data set



A = <https://raw.githubusercontent.com/laploy/ML/master/mul.csv>

B = [https://raw.githubusercontent.com/laploy/ML/master/mul\\_token.csv](https://raw.githubusercontent.com/laploy/ML/master/mul_token.csv)

## Edit Metadata

## ▲ Edit Metadata

Column

**Selected columns:****Column names:**

Col1,Col2

A

[Launch column selector](#)

Data type

Unchanged ▾

Categorical

Unchanged ▾

Fields

Unchanged ▾

New column names

id,article

## ▲ Edit Metadata

Column

**Selected columns:****All columns**

B

[Launch column selector](#)

Data type

Unchanged ▾

Categorical

Unchanged ▾

Fields

Unchanged ▾

New column names

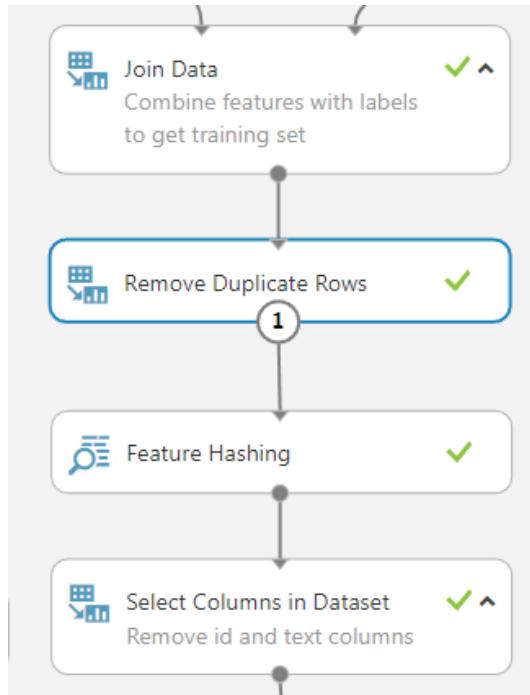
category,id

## Splitting Data

Used only the rows already tagged with hierarchy names (CCAT,ECAT,GCAT,MCAT)

|                        |  | Before splitting |         | After splitting |         |
|------------------------|--|------------------|---------|-----------------|---------|
|                        |  | rows             | columns | rows            | columns |
| <b>Split Data</b>      |  | 36457            | 2       | 13417           | 2       |
| Splitting mode         |  |                  |         |                 |         |
| Regular Expression     |  |                  |         |                 |         |
| Regular expression     |  |                  |         |                 |         |
| \\"category" [GCME]CAT |  |                  |         |                 |         |
|                        |  | Col1             | Col2    | category        | id      |
| view as                |  | view as          | view as | view as         | view as |
|                        |  |                  |         |                 |         |
|                        |  | E11              | 2286    | ECAT            | 2286    |
|                        |  | ECAT             | 2286    | MCAT            | 2286    |
|                        |  | M11              | 2286    | CCAT            | 2287    |
|                        |  | M12              | 2286    | CCAT            | 2288    |
|                        |  | MCAT             | 2286    | ECAT            | 2288    |
|                        |  | C24              | 2287    | GCAT            | 2288    |

## Feature & Clean



### Join Data

Join key columns for L

**Selected columns:**  
Column names: id

[Launch column selector](#)

Join key columns for R

**Selected columns:**  
Column names: id

[Launch column selector](#)

Match case



Join type

Inner Join



Keep right key colu...



### Remove Duplicate Rows

Key column selection filter exp...

**Selected columns:**  
Column names: id

[Launch column selector](#)

Retain first duplicate r...

## Feature Engineering

### ▲ Feature Hashing

Target column(s)

**Selected columns:**

**Column names:** article

Launch column selector

Hashing bitsize

8

N-grams

1

### ▲ Select Columns in Dataset

Select columns

**Selected columns:**

**All columns**

**Exclude column names:**

id,article

Launch column selector

# Algorithm

## ▲ Multiclass Decision Forest

Resampling method 

Bagging 

Create trainer mode 

Single Parameter 

Number of decision trees 

8 

Maximum depth of the ... 

32 

Number of random split... 

128 

Minimum number of sa... 

1 

Allow unknown val... 

## ▲ Train Model

Label column

**Selected columns:**

**Column indices:** 1

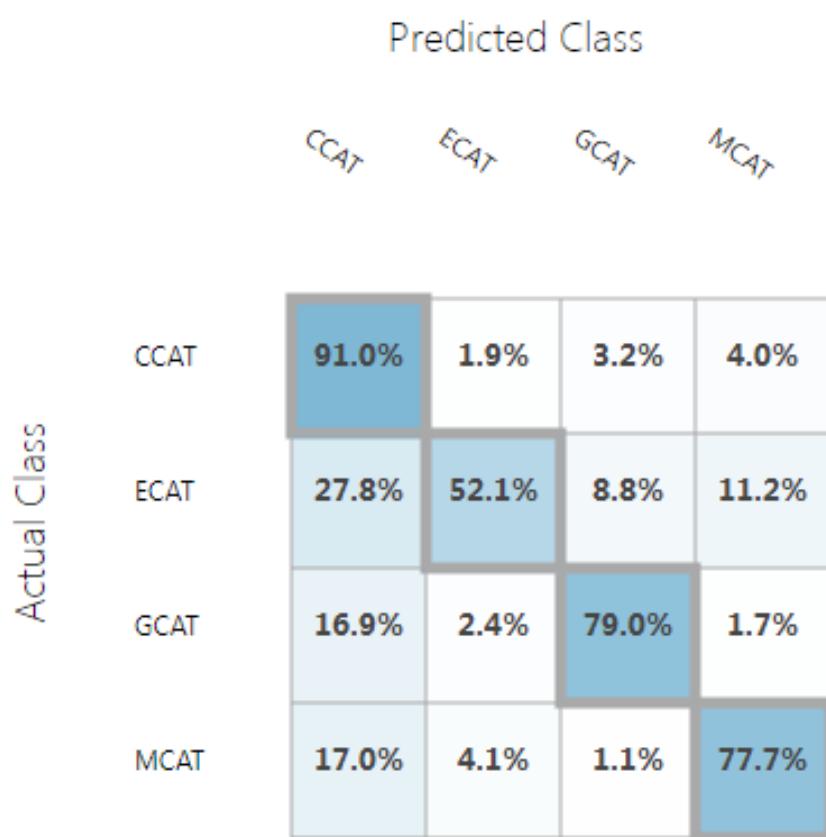
[Launch column selector](#)

## Confusion Matrix

Test data = <https://raw.githubusercontent.com/laploy/ML/master/mul-test.txt>

### Metrics

|                          | Predicted Class |
|--------------------------|-----------------|
| Overall accuracy         | 0.813474        |
| Average accuracy         | 0.906737        |
| Micro-averaged precision | 0.813474        |
| Macro-averaged precision | 0.802249        |
| Micro-averaged recall    | 0.813474        |
| Macro-averaged recall    | 0.749342        |



More information

Multiclass Decision Forest

<https://msdn.microsoft.com/en-us/library/azure/dn906015.aspx>

This Experiment

<https://gallery.cortanaintelligence.com/Experiment/Multi-Class>

# RETRAIN ML



## In this session

- Retrain workflow
- Create new training experiment
- Create/Publish predictive experiment
- Create/publish a retrain experiment (add IO)
- Create C# console Application BES
- Get keys from Azure Storage Account
- Update C# code input/output
- Get iLearner information
- Review retrain evaluation
- Add a new Endpoint
- Update endpoint

## Retrain workflow

Create the initial Predictive Web service:

- Create a training experiment
- Create a predictive web experiment
- Deploy a predictive web service

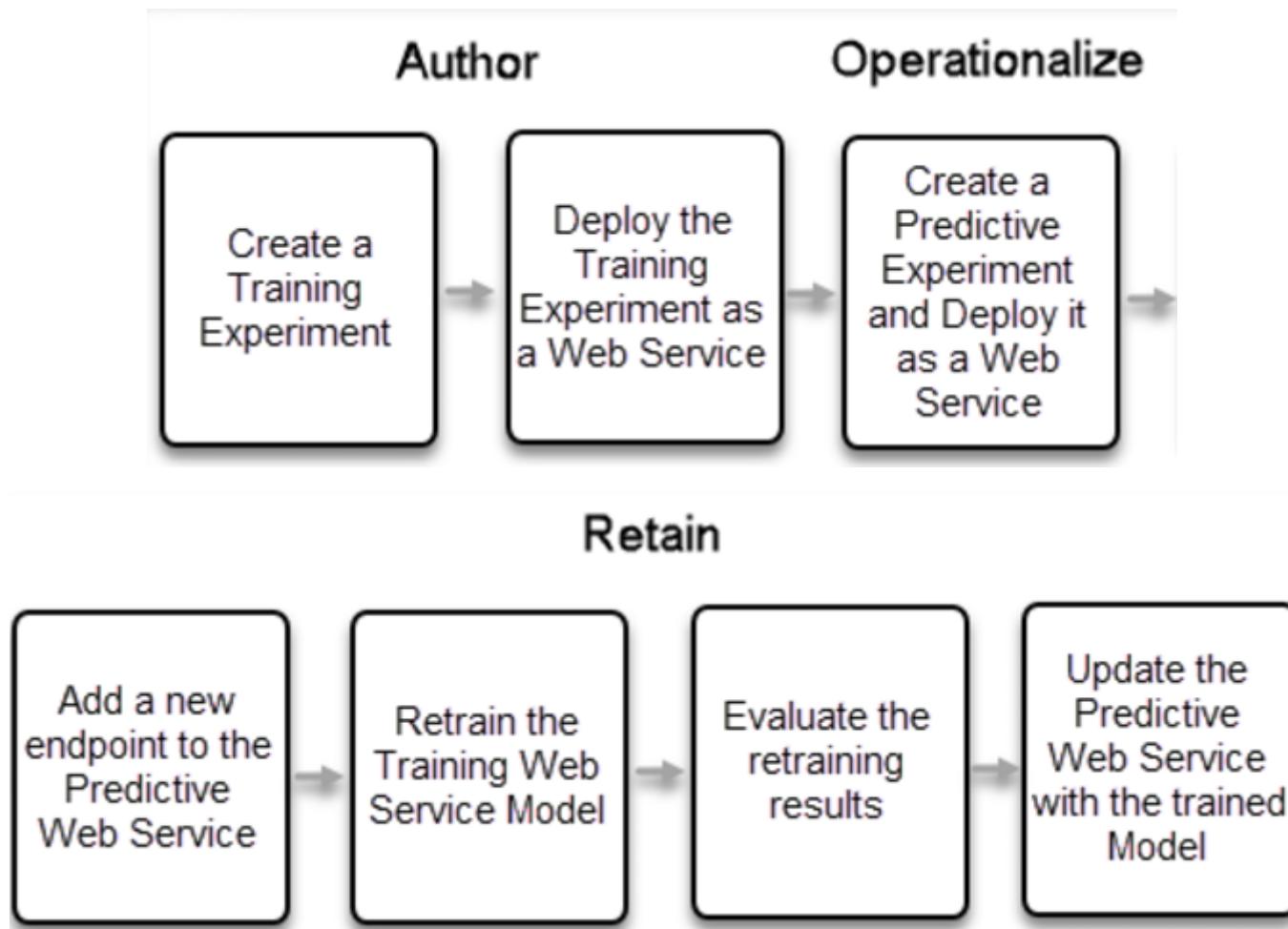
Retrain the Web service:

- Update training experiment to allow for retraining
- Deploy the retraining web service
- Use the Batch Execution Service code to retrain the model

Update endpoint

- Create a new Endpoint on the Predictive Web service
- Get the PATCH URL and code
- Use the PATCH URL to point the new Endpoint at the retrained model

## Retrain workflow diagram



## Create new training experiment

Create new ML training experiment by downing an example from Cortana intelligence gallery

1. Go to webpage Cortana intelligence gallery <https://gallery.cortanaintelligence.com>
2. Enter loy in search box
3. Click Census Model 001
4. Click Open in Studio
5. RUN
6. Click SET UP WEB SERVICE and Predictive web service
7. RUN
8. Change name of Predictive experiment to Census Model 001 Predic
9. RUN
10. Click DEPLOY WEB SERVICE

## Create a training experiment

Cortana Intelligence Gallery

Browse all Industries Solutions Experiments More

EXPERIMENT

# Census Model 001

LA Laploy V. Angkul • July 9, 2017

[edit](#)

### Summary

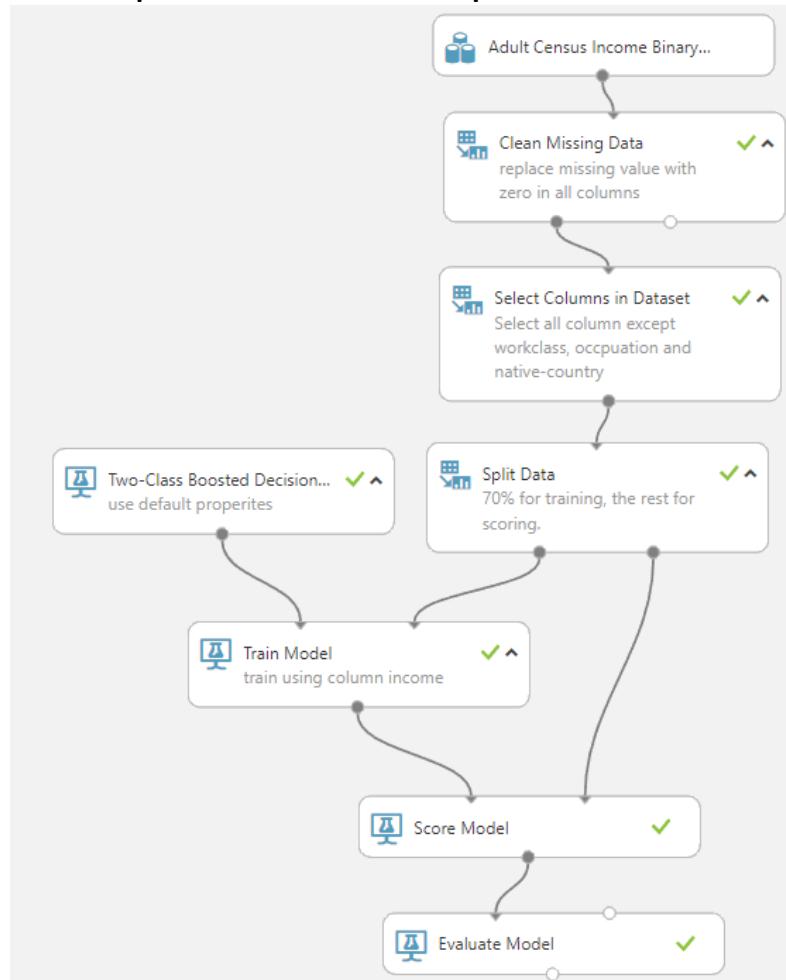
This experiment demonstrates how we can build a binary classification model to predict income levels of adult individuals. The process includes training, testing and evaluating the model on the Adult dataset.

### Description

[Open in Studio](#)[+ Add to Collection](#)

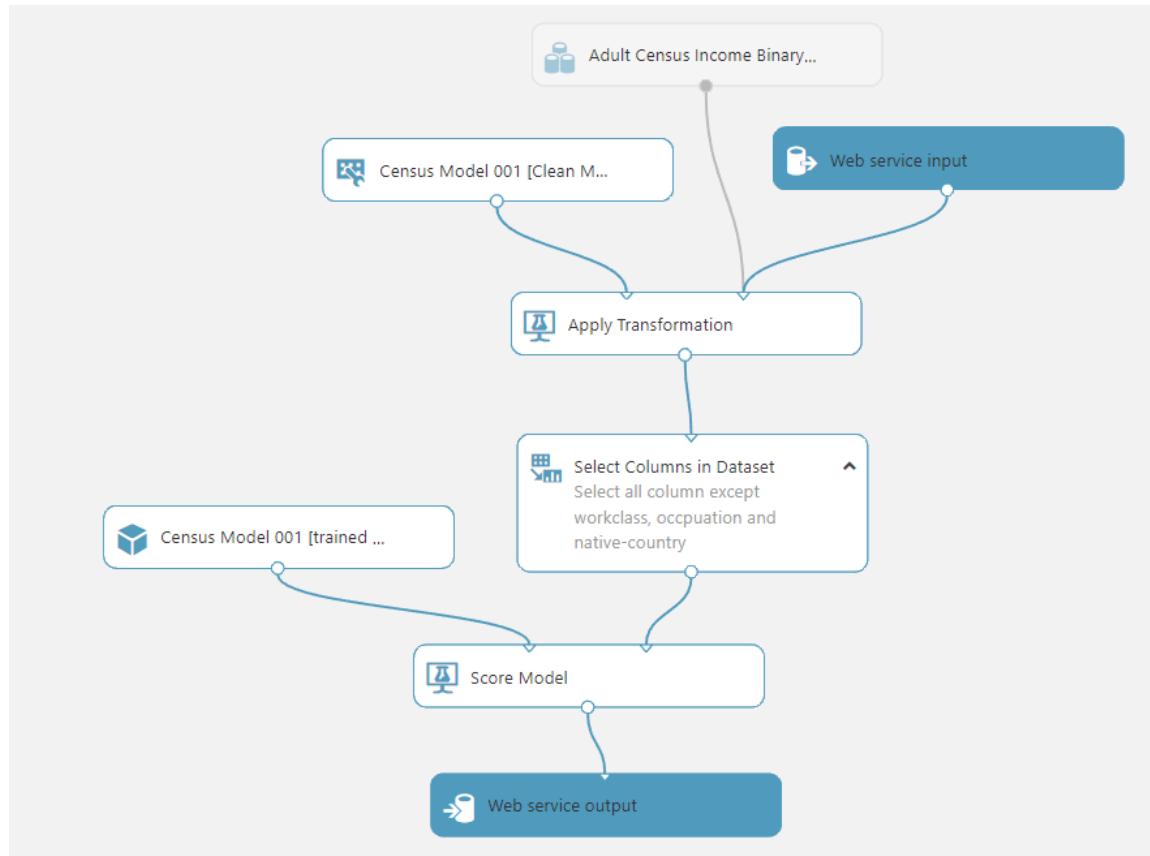
## Create new training experiment

### Experiment when open in Studio



## Create/Publish predictive experiment

RUN, SET UP WEB SERVICE / Predictive Web Service [Recommended]



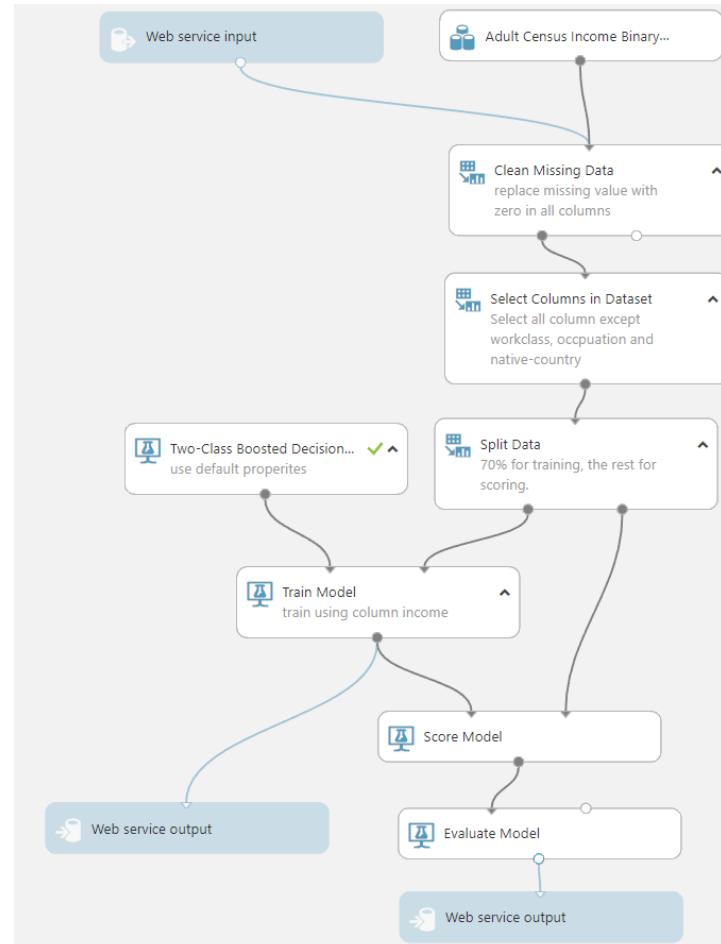
Experiment after SET UP WEB SERVICE and Predictive web service

Create/publish a retrain experiment (add IO)

1. Go back to Census Model 001 Experiment
2. Click Training experiment tab
3. Add a web service input module
4. Add two web service output modules
5. Run
6. Click SET UP WEB SERVICE / DEPLOY WEB SERVICE

## Create/publish a retrain experiment (add IO)

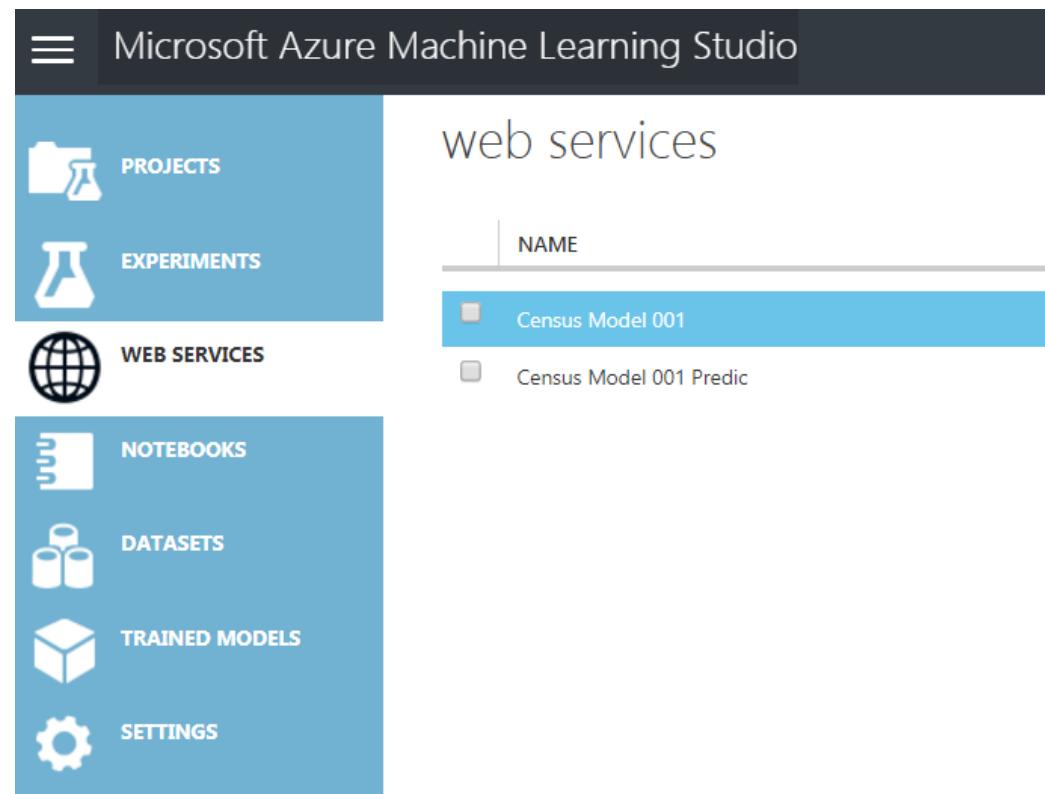
### Experiment after add web service input / outputs



## Create/publish a retrain experiment (add IO)

Click WEB SERVICES

1. Census Model 001 = retrain
2. Census Model 001 Predic = production



## Create C# console Application BES

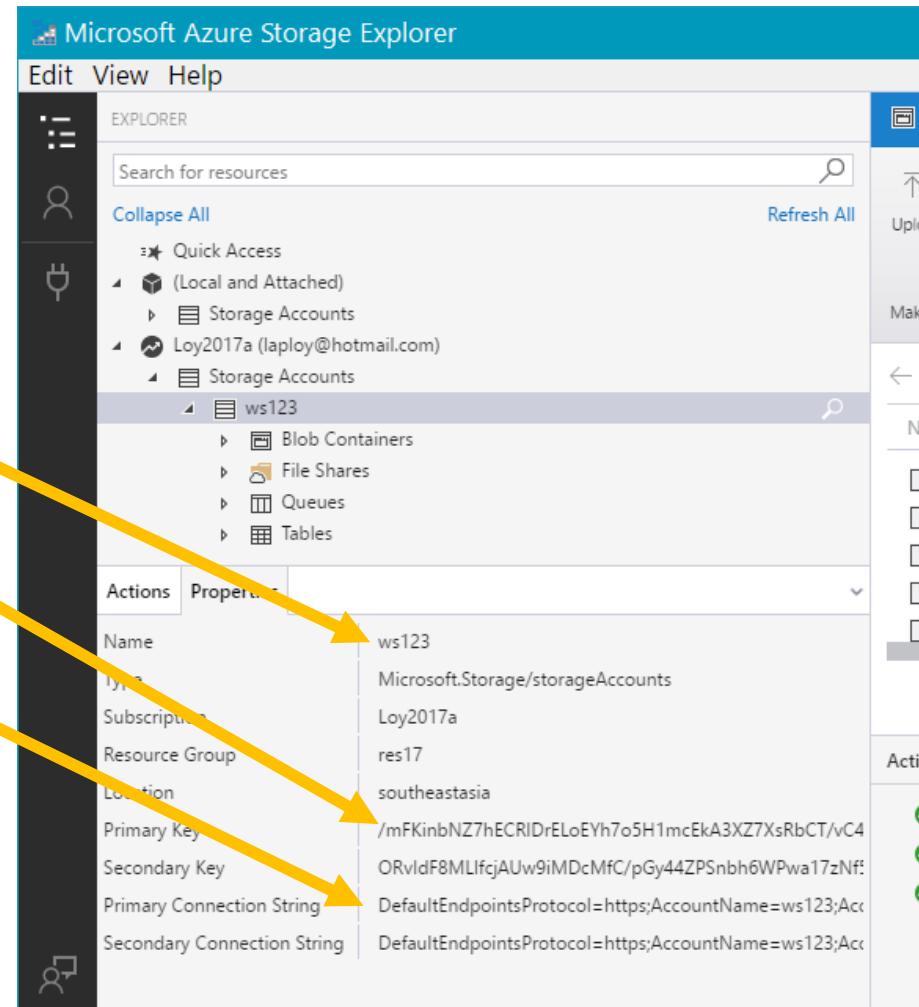
1. Create a C# Console Application in Visual Studio (New->Project->Windows Desktop->Console Application)
2. Solution/project name = census1
3. Nuget add Microsoft.WindowsAzure.Storage.dll
4. Nuget add Microsoft.AspNet.WebApi.Client
5. Open Microsoft Azure Machine Learning Studio page
6. Click Web Service
7. Click census model 001
8. Click BATCH EXECUTION
9. Copy C# sample code
10. Past code \*\* Note on name space

## Get keys from Azure Storage Account

Run program Microsoft Azure Storage Explorer

Copy and save to Notepad

1. Storage Container Name
2. Storage Account Key
3. Storage Connection String



## Update C# code input/output

### 1. Get Web service API Key from AML BES page (copy and save to Notepad)

The screenshot shows the Microsoft Azure Machine Learning Studio interface. The title bar says "Microsoft Azure Machine Learning Studio". The main content area is titled "census model 001". On the left, there's a vertical sidebar with icons for Dashboard, General, Published experiment, Description, Predictive web service, and Census Model 001 Prediction. The "Census Model 001 Prediction" section is expanded, showing an "API Key" field containing the value "Yqls/huXn6l7M5vose4uVkosD15hDRhxPYhZY2sH6qwDZG5T4EaftrwujtOMW81pn997rZTp26/SC". This field is highlighted with a red dashed border. Below it is a "Default Endpoint" field, also highlighted with a red dashed border. At the bottom, there are tabs for "API HELP PAGE" (selected), "REQUEST/RESPONSE", and "BATCH EXECUTION". Under "REQUEST/RESPONSE", there are "Test" and "Test preview" buttons, with "Test" being blue and "Test preview" being grey.

## Update C# code input/output

### Update C# code for keys

```
// *****
// loy
const string StorageAccountName = "ws123"; // Replace this with your storage account name
const string StorageAccountKey = "/mFKinbNZ7hECR1DrELoEYh7o5H1mcEk";
const string StorageContainerName = "test1"; // Replace this with the name of the container you want to use
string storageConnectionString = string.Format("DefaultEndpointsProtocol=https;AccountName={0};AccountKey={1};BlobEndpoint=https://{0}.blob.core.windows.net;QueueEndpoint=https://{0}.queue.core.windows.net;TableEndpoint=https://{0}.table.core.windows.net;FileEndpoint=https://{0}.file.core.windows.net");
const string apiKey = "mbrj11ijM8MB3IyQ5h08tZJbnn+101Ru00RWCS6xQ50I";
```

## Update C# code input/output

- Update C# code UploadFileToBlob

```
// **** Loy
UploadFileToBlob(@"d:\temp\cenin1.csv" /*Replace this with the location of your input file*/,
 "cenin1.csv" /*Replace this with the name you would like to use for your Azure blob; this r
StorageContainerName, storageConnectionString);
```

- Update C# code input file name

```
Inputs = new Dictionary<string, AzureBlobDataReference>()
{
 {
 "input1",
 new AzureBlobDataReference()
 {
 ConnectionString = storageConnectionString,
 // **** Loy
 RelativeLocation = string.Format("{0}/cenin1.csv", StorageContainerName)
 }
 },
},
```

## Update C# code input/output

### Update C# code Output1 / Output2 file name

```
Outputs = new Dictionary<string, AzureBlobDataReference>()
{
 {
 "output2",
 new AzureBlobDataReference()
 {
 ConnectionString = storageConnectionString,
 // *****
 RelativeLocation = string.Format("/{0}/cenout.ilearnert", StorageContainerName)
 }
 },
 {
 "output1",
 new AzureBlobDataReference()
 {
 ConnectionString = storageConnectionString,
 // *****
 RelativeLocation = string.Format("/{0}/cenout.csv", StorageContainerName)
 }
 },
}
```

## Update C# code input/output

Update C# code Main method to show End program status

```
U references
static void Main(string[] args)
{
 InvokeBatchExecutionService().Wait();
 // ***** Loy
 Console.WriteLine("End program");
 Console.Read();
}
```

## Retrain and evaluate

### Run program

1. Download file cenin1.csv from <https://github.com/laploy/ML/blob/master/cenin1.csv>
2. Place file cenin1.csv in to d:\temp
3. Run C# Program
4. Wait for End program message

Run this program whenever you have a good training dataset and want to retrain the model

## Get iLearner information

Get iLearner information from program output

Copy and paste to Notepad

1. RelativeLocation
2. BaseLocation
3. SasBlobToken

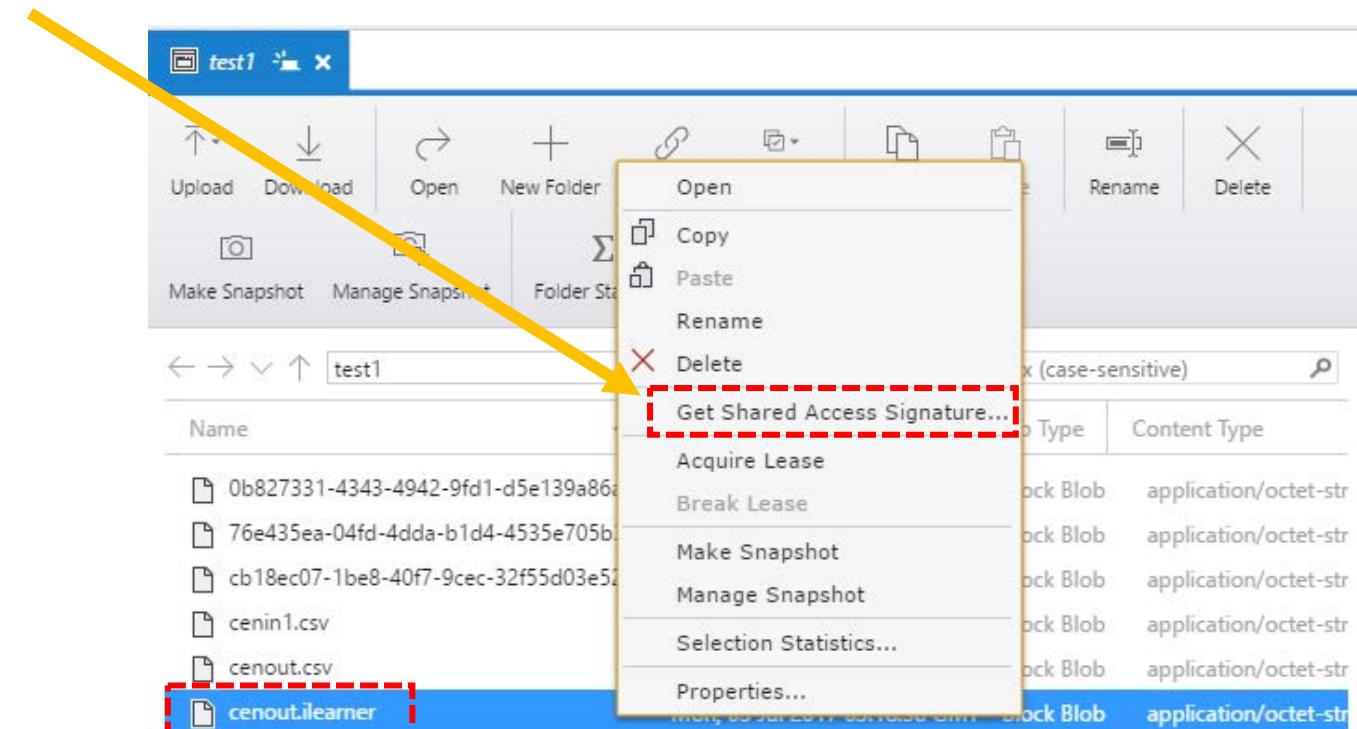
iLearner is the training model we need  
to update the Prediction model

```
The result 'output1' is available at the following A
BaseLocation: https://aihelpwebsitestorage.blob.core
RelativeLocation: experimentoutput/output1results.il
SasBlobToken: ?sv=2015-02-21&sr=b&sig=0EJsI719TZ39sh
nyw%3D&st=2017-03-2FnVMgsavxnK0urZ&se=2017-03-27T01%
```

## Get iLearner information

Or using ASE

1. Open Microsoft Azure Storage Explorer
2. Right-click at file cenout.ilearner
3. Click Get Shared Access Signature



## Get iLearner information

Click Create

Shared Access Signature

Access policy:

Start time:

Expiry time:

Time zone:

Local  
 UTC

Permissions:

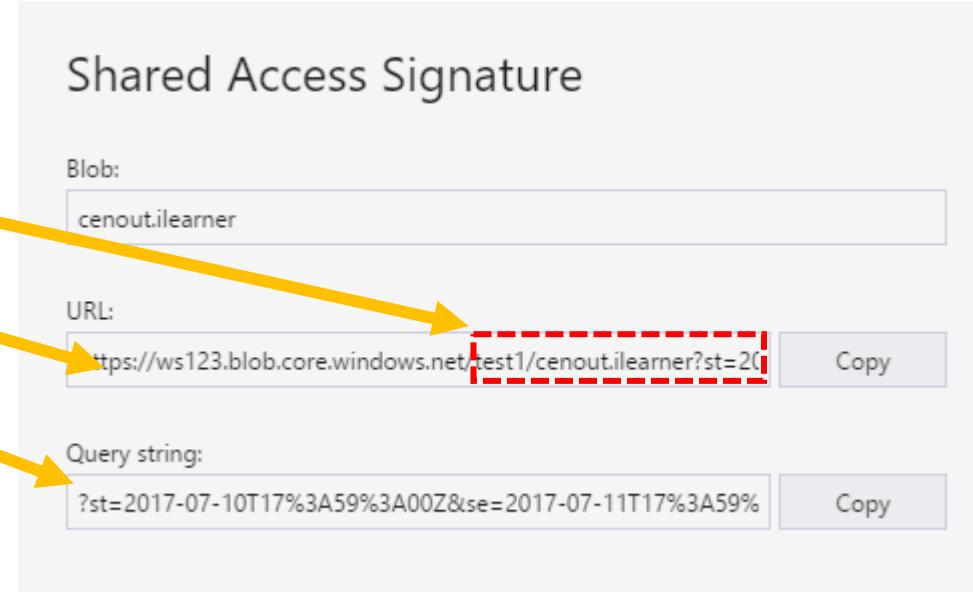
Read  
 Write  
 Delete  
 List

Generate container-level shared access signature URI

## Get iLearner information

Copy and paste to Notepad

- 4. RelativeLocation
- 5. BaseLocation
- 6. SasBlobToken

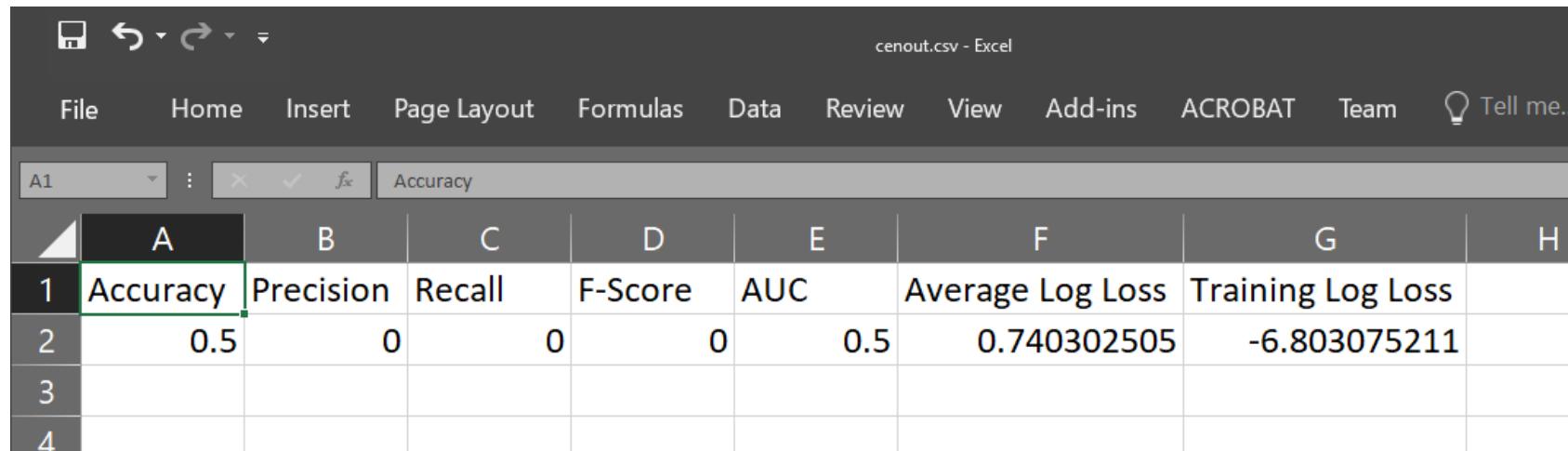


Base location = from start to .net

## Review retrain evaluation

### Review retrain evaluation

1. Open Microsoft Azure Storage Explorer
2. Download file cenout.csv
3. Open in Microsoft Excel or Windows Notepad
4. Examine the results



The screenshot shows a Microsoft Excel spreadsheet titled "cenout.csv - Excel". The table has the following data:

|   | A        | B         | C      | D       | E   | F                | G                 | H |
|---|----------|-----------|--------|---------|-----|------------------|-------------------|---|
| 1 | Accuracy | Precision | Recall | F-Score | AUC | Average Log Loss | Training Log Loss |   |
| 2 | 0.5      | 0         | 0      | 0       | 0.5 | 0.740302505      | -6.803075211      |   |
| 3 |          |           |        |         |     |                  |                   |   |
| 4 |          |           |        |         |     |                  |                   |   |

## Add a new Endpoint

Click New Web Services Experience

Microsoft Azure Machine Learning Studio

census model 001 **predic**

DASHBOARD CONFIGURATION

General **New Web Services Experience** preview

Published experiment

[View snapshot](#) [View latest](#)

Description

No description provided for this web service.

Training web service

Prediction WS not Retrain WS

## Add a new Endpoint

Click Census Model 001 Predic

Microsoft Azure Machine Learning Web Services

Quickstart   Dashboard   Batch Request Log   Configure   Consum

Census Model 001 Predict

default

BASICS

MANAGE & MONITOR

Test endpoint   Configure endpoint   Use endpoint

Tutorial: Retrain web service

## Add a new Endpoint

Click + NEW

Microsoft Azure Machine Learning Web Services

← Classic Web Services

# Census Model 001 Predict

No description provided for this web service.

Search

| NAME    | BATCH CALLS | FAILURES |
|---------|-------------|----------|
| default | 0           | 0        |

+ NEW    DELETE

## Add a new Endpoint

Enter name, description and click Save

Microsoft Azure Machine Learning Web Services

### Census Model 001 Predict

No description provided for this web service.

Search

+ NEW DELETE

Create new endpoint

Name: retrain

Description: retrain test 1

Logging: None Error All Logging Help

Sample Data Enabled?: Yes No

Cancel Save

## Update endpoint

Click retrain end point

The screenshot shows the Microsoft Azure Machine Learning Web Services interface. The top navigation bar includes a menu icon and the text "Microsoft Azure Machine Learning Web Services". Below the navigation bar, there is a back arrow labeled "Classic Web Services" and the title "Census Model 001 Predict". A note below the title states "No description provided for this web service." A search bar is present above the endpoint list. Below the search bar are buttons for "+ NEW" and "DELETE". The endpoint list table has columns: NAME, BATCH CALLS, FAILURES, and SUCCES. Two rows are listed: "default" and "retrain". The "retrain" row is highlighted with a red dashed border. At the bottom right of the table, there is a page number indicator "1 ▾ / 1".

|                          | NAME    | BATCH CALLS | FAILURES | SUCCES |
|--------------------------|---------|-------------|----------|--------|
| <input type="checkbox"/> | default | 0           | 0        | 0      |
| <input type="checkbox"/> | retrain | 0           | 0        | 0      |

## Update endpoint

Click Consume

Microsoft Azure Machine Learning Web Services

Quickstart    Dashboard    Batch Request Log    Configure    Consume    Te

← Census Model 001 Predic  
**retrain**  
retrain end point test

BASICS

MANAGE & MONITOR

DEV

## Update endpoint

### Click API Help

Microsoft Azure Machine Learning Web Services

Quickstart Dashboard Batch Request Log

retrain end point test

Web service consumption options

- Excel 2013 or later
- Excel 2010 or earlier
- Request-Response Web App Template

Basic consumption info

Want to see how to consume this information? [Check out this easy tutorial.](#)

Primary Key: vz0MsD4YYo3dCSyhQsqTnTxCeehQtqxxaBh9vNONG611

Secondary Key: 6M+Zn4xmCcq1A4Eg/WFSv7Frz9PEohHmVcuhDlse5LkV

Request-Response: <https://ussouthcentral.services.azureml.net/workspaces/7193/execute?api-version=2.0&format=swagger>  
[API Help](#) [Documentation](#)

Batch Requests: <https://ussouthcentral.services.azureml.net/workspaces/7193/jobs?api-version=2.0>  
[API Help](#) [Documentation](#)

Patch: <https://management.azureml.net/workspaces/ede12cb3/endpoints/retrain>  
[API Help](#) [Documentation](#)

Update endpoint

Click Sample Code

# Update Resource API Documentation

Updated: 07/10/2017 04:08

No description provided for this web service.

- [Request and Response summary](#)
- [Sample Code](#)
- [API Swagger Document](#) 
- [Endpoint Management Swagger Document](#) 

## Updatable Resources

Resource Name

Copy C# code

## Update endpoint

Create a program to update the endpoint

- Open Visual Studio
- Create new C# Windows console app project
- Name = CallUpdateResource
- Paste code to main
- Update
  1. const string apiKey: get key from web service page
  2. BaseLocation
  3. RelativeLocation
  4. SasBlobToken
  5. Add a message to show success end
- Run program

Run this program only once

## More information

Retrain a Machine Learning Model

<https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-retrain-machine-learning-model>

Census Model 001

<https://gallery.cortanaintelligence.com/Experiment/Census-Model-001>



Azure  
Machine Learning

กรุณาใช้ที่อยู่นี้ สำหรับออกใบหักภาษี ณ ที่จ่าย หรือเอกสารธุรกรรมอื่นๆ

**บริษัท เกรทเฟรนด์ บีชชินเนช ดีเวลลอปเม้นท์ จำกัด**

5/35 หมู่ 3 หมู่บ้านมายเพลส วัชรพล ถนนเพิ่มสิน

แขวงอโศก เมืองสายไหม

กรุงเทพฯ 10220

โทรศัพท์: 0-2992-4877

โทรสาร: 0-2992-4878

มือถือ: 081-915-7816 (อ.สุเทพ)