Fundação Vanzolini

Dominando Big Data com o uso de Plataformas Gratuitas (nível intermediário)

Aula 1

LexisNexis® RISK SOLUTIONS

RELX Group

PRO

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

USP UNIVERSIDADE DE SÃO PAULO

Agenda da aula 1

- ✓Apresentação do curso
- ✓Revisão de conceitos: HPCC Systems e ECL
- ✓Configuração do ambiente

Treinamento em ECL/HPCC: learn.lexisnexis.com/hpcc

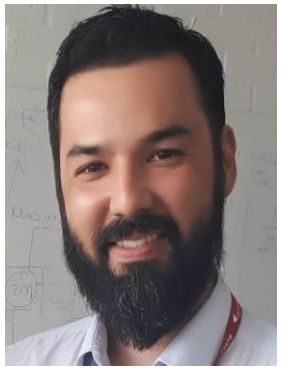
- Introdução ao ECL (parte 1)
  - Conceitos e consultas
- Introdução ao ECL (parte 2)
  - ETL com ECL
- ECL Avançado (parte 1)
  - Dados relacionais
- ECL Avançado (parte 2)
  - Superarquivos, XML/JSON e PLN
- ECL Aplicado
  - Geração e automação de código ECL
- ROXIE ECL (parte 1)
  - Índices e consultas
- ROXIE ECL (parte 2)
  - Otimização de consultas
- Machine Learning com HPCC Systems
  - Tutoriais para uso de plugins
- Administração de Sistemas
  - Conceitos e operação básica
- HPCC para gestores
  - Visão geral e aplicações da plataforma

Suporte e operação

- Computador pessoal (ECL IDE v8.0.2/VSCode/GitPod)
- Cluster: http://54.215.2.79:8010/
- Slides de aula e livro de exercícios
- Moodle: https://ead.vanzolini.org.br/course/view.php?id=666
- Horários das aulas: 19:00hs às 22:00hs
- Dias das aulas: 22, 24, 26, 29 de novembro; 1 e 3 de dezembro
- Certificado USP e badges HPCC Systems

Coordenação

- Prof. Hugo Watanuki (hwatanuki@usp.br)
  - Doutor em engenharia de produção POLI-USP
  - Engenheiro de suporte técnico na LexisNexis Risk Solutions



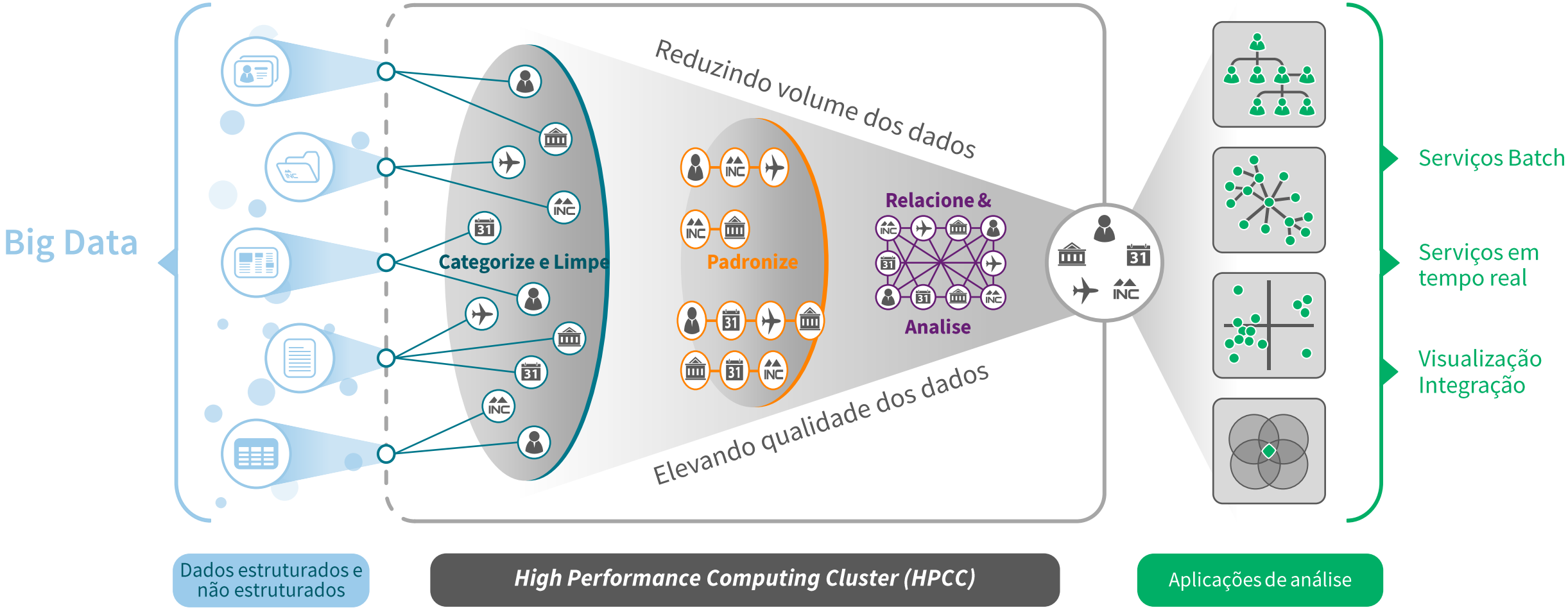
- Prof. Renato Moraes (remo@usp.br)
  - Doutor em administração pela FEA-USP
  - Professor do depto. de engenharia de produção POLI-USP

Introdução

- ✓Nome
- ✓Área de atuação
- ✓Uso e interesse em Big Data

Revisão de conceitos: HPCC Systems e ECL

Extract, Transform, Load

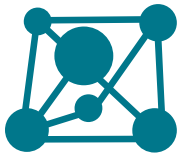


Fundação Vanzolini

“*Stack*” tecnológico

Ferramentas de consulta e visualização

Entrega online de consultas em *Big Data*



Bibliotecas de *Machine Learning*

Supervisionado, não-supervisionado, aprendizagem profunda

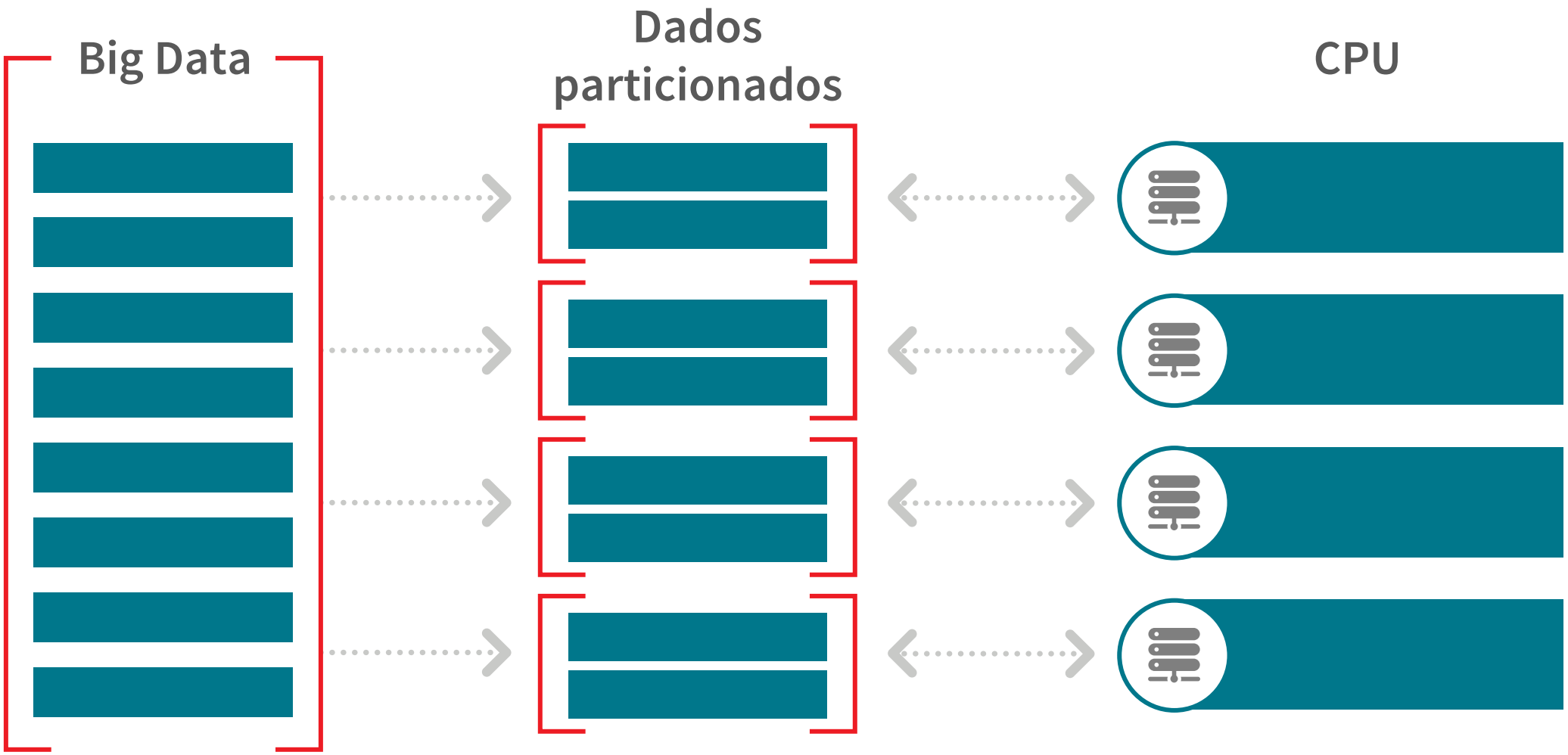
Ferramentas para manipulação de dados

Perfilamento, limpeza, consolidação de dados

ETL

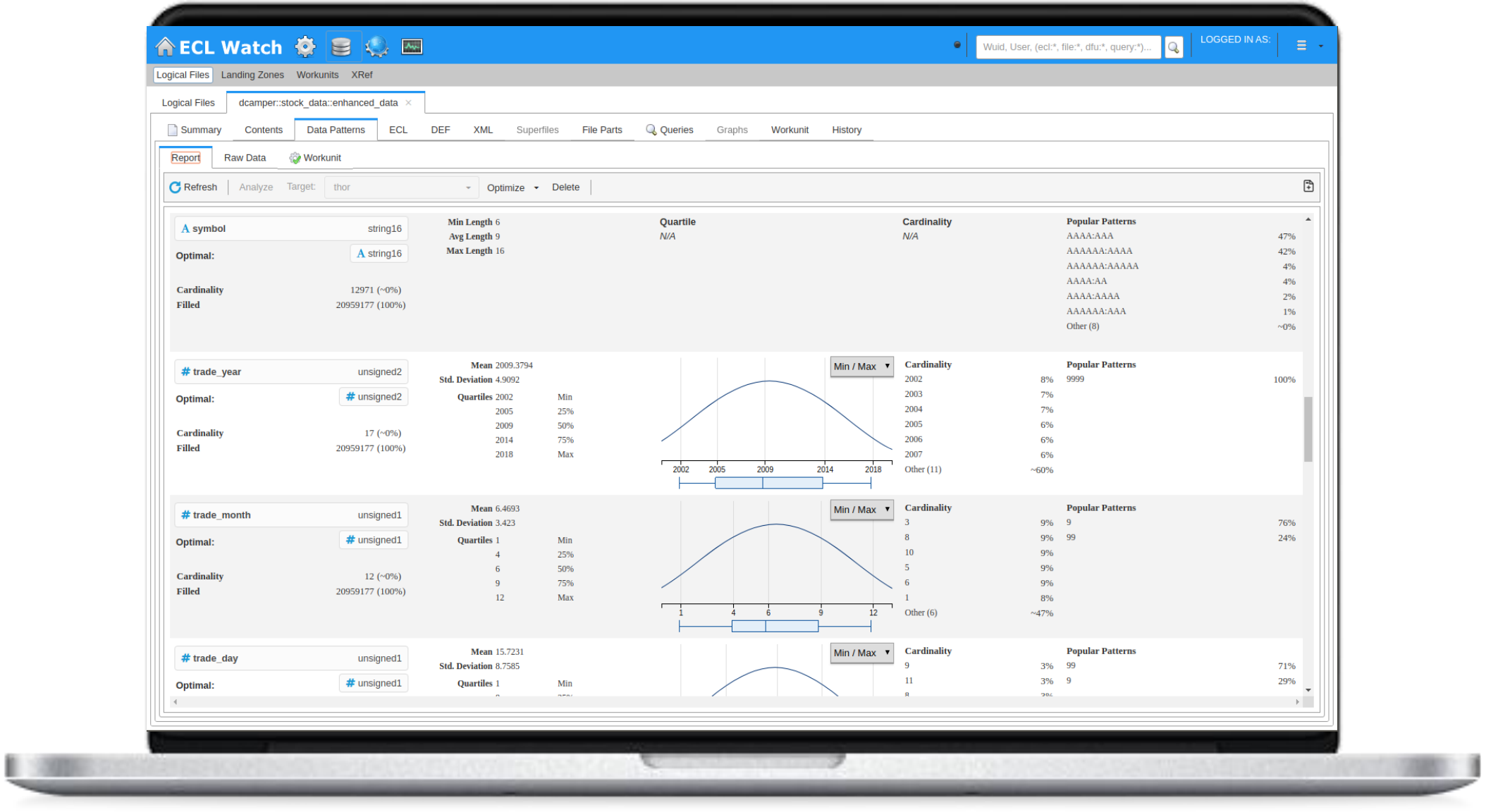
Extração, transformação e carregamento de dados

ETL: Supercomputação



Fundação Vanzolini

Ferramentas de *Profiling*



Fundação Vanzolini

Bibliotecas de *Machine Learning*

Não supervisionado

Clusterização
DBSCAN
K-Means

PLN
Text Vectors

Supervisionado

Classificação
SVM
Árvores de decisão
Regression logística
Classification Forest

Regressão
Regressão linear
GLM
Regression Forest

Redes neurais & Deep Learning

Autoencoders

Redes neurais convolucionais

Redes neurais recorrentes

Perceptrons

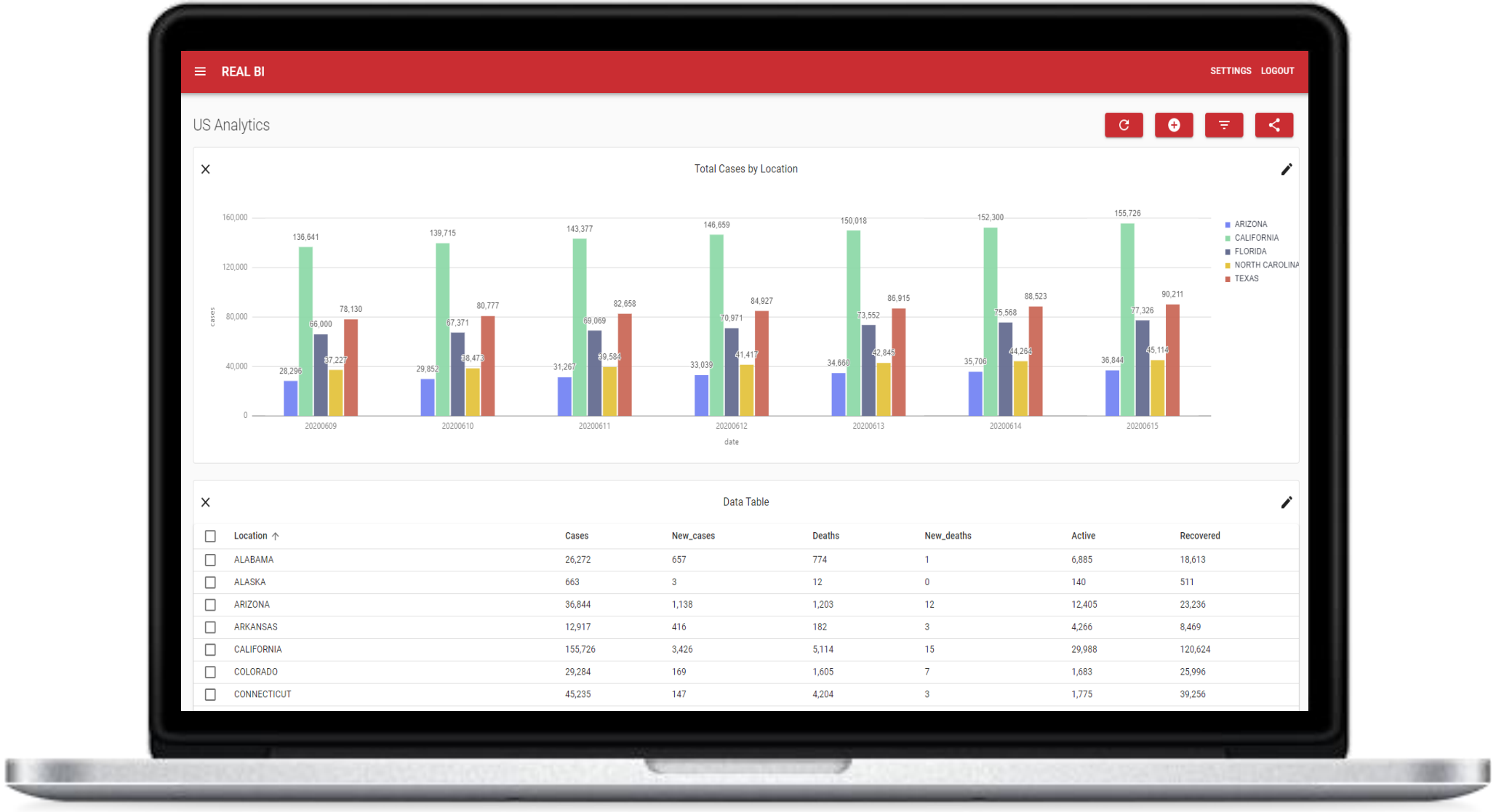
Métodos ensemble

Random Forest

Gradient Boosted Forest

Gradient Boosted Trees

Ferramentas de Consulta e Visualização



Location	Cases	New_cases	Deaths	New_deaths	Active	Recovered
ALABAMA	26,272	657	774	1	6,885	18,613
ALASKA	663	3	12	0	140	511
ARIZONA	36,844	1,138	1,203	12	12,405	23,236
ARKANSAS	12,917	416	182	3	4,266	8,469
CALIFORNIA	155,726	3,426	5,114	15	29,988	120,624
COLORADO	29,284	169	1,605	7	1,683	25,996
CONNECTICUT	45,235	147	4,204	3	1,775	39,256

Fundação Vanzolini

Conceitos básicos de ECL

- Paradigma declarativo (não-procedural)
- Estrutura básica: `Nome := Expressão ;`
- ECL *não é* sensível a caixa alta/baixa
- Espaço em branco é ignorado
- Comentários em linha (//) e em bloco (/* e */)
- ECL utiliza sintaxe objeto.propriedade

```
Dataset.Campo                  // referencia um campo em um dataset
NomedoDiretorio.Definicao      // referencia uma definição em outro diretório
```

Ações vs. Definições

- ✓ O código ECL é constituído de:
  - ✓ *Definições* estabelecem *o que* as coisas são (arquivos de definição ECL)

```
A := 'People';  // não inicia uma WU
```

  - ✓ *Ações* resultam em compilação e execução (arquivos BWR)

```
OUTPUT('People');          // inicia uma WU
```

Tipos de dados primitivos

BOOLEAN

```
BOOLEAN IsFloridian   := TRUE;
```

STRING[n]

```
STRING1 Gender := 'M';
```

INTEGER[n], UNSIGNED[n],

```
INTEGER1 ictr := -100;      // -128 to 127
UNSIGNED1 ctr := 0;      // 0 - 255
```

REAL[n], DECIMALn[_y]

```
REAL4 PI := 3.14159;
DECIMAL7_2 Salary := 75000.00;
```

Tipos básicos de definição ECL

Booleana (*boolean*)

```
IsSeniorCitizen := People.birthdate>19600101;
```

Valor único (*value*)

```
MaleValue := 'M';
```

Conjunto de valores (*set*)

```
GenderValues := ['M','F'];
```

People

##	firstname	lastname	middlename	namesuffix	filedate	bureaucode	maritalstatus	gender	dependentcount	birthdate	streetaddress
1	Cherianne	Khatchatourian	N		19990922	24		M	0		69 BOULDER RIDGE RD # 25
2	Muyesser	Raplee	X		20001111	353		F	0		55 SWAMP RD
3	Roselin	Viceconte			19990325	344		F	0	19800113	107 HILL TER
4	Inda	Provines			20000909	13		U	0		290 W MOUNT PLEASANT AVE
5	Inderdeep	Laurence	D		20001228	344		M	0		44 PROSPECT PL
6	Chrystine	Mangiapane			19990827	315		F	0	19780306	1806 1ST AVE APT 8F
7	Adelene	Stock	R		20000827	252		M	0		1117 FARM RD
8	Mendy	Rufenblanchette			20000903	24		M	0		3 W 83RD ST APT 4C
9	Lannie	Amerantes	I		20001219	313		U	0		200 W 20TH ST APT 909
10	Tare	Gonyeau	T		19930807	48		F	0	19750801	6 CANDLE CT

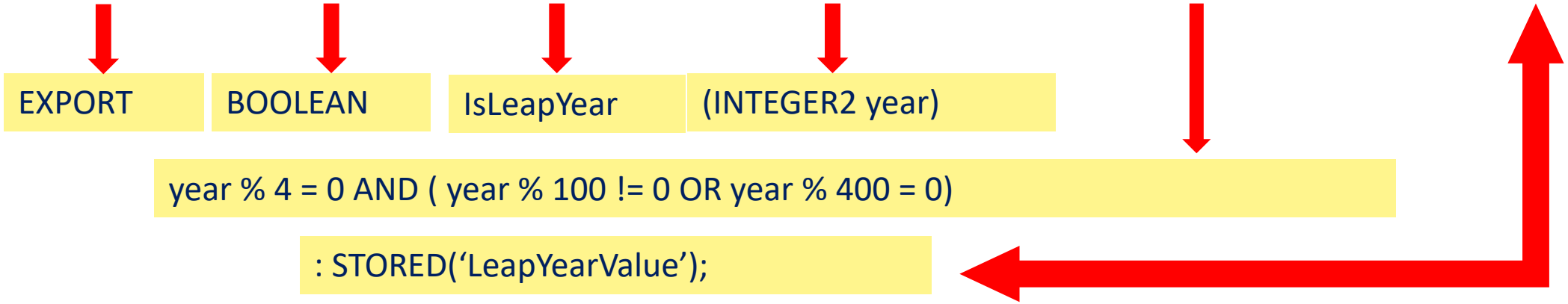
Conjunto de registros (*recordset*)

```
SeniorPeople := People(IsSeniorCitizen);
MalePeople := People(Gender = MaleValue);
FemaleMalePeople := People(Gender IN GenderValues);
```

Sintaxe Completa de uma Definição ECL

Nome := Expressão ;

[Escopo] [TipoValor] Nome [(parâmetros)] := Expressão [:ServiçoWorkflow] ;



Escopo da definição (Visibilidade)

Global –

A palavra-chave **EXPORT** torna a definição disponível "globalmente" no repositório

```
EXPORT PeopleCount := COUNT(People);
```

Módulo –

A palavra-chave **SHARED** torna a definição disponível somente no modulo/diretório que a contém

```
SHARED StateCount := 50;
```

Local –

A **ausência dessas palavras-chave** torna a definição disponível somente no arquivo que a contém e até a próxima definição ECL que contenha EXPORT ou SHARED

```
Num5 := 5;
EXPORT NumTotal := Num5 + 10 + StateCount;
```

Escopo da definição (Visibilidade)

IMPORT listadiretorios

- listadiretorios – Uma lista de diretórios separados por vírgula.

A palavra-chave **IMPORT** define uma lista de diretórios cujos arquivos de definições exportados tornam-se diponíveis para uso no código.

```
IMPORT Companies;          // Definições Exportadas em Companies estão disponíveis
FloridaCompanies := Companies.File_Company(state='FL');

IMPORT $;                  // Definições Exportadas no Módulo atual estão disponíveis
FloridaCompanies := $.File_Company(state='FL');
```

Estrutura MODULE

A estrutura **MODULE** permite agrupar e fornecer parâmetros para um conjunto de definições ECL relacionadas.

nome [(*parametros*)] := **MODULE**

definições;

END;

- *Nome* – O nome da definição ECL do modulo.
- *parametros* – Os parâmetros disponíveis para todas as *definições*.
- *definições* – As definições ECL que compõem o módulo.

Começando a trabalhar com dados

Antes de começar a trabalhar com qualquer arquivo de dados na plataforma HPCC Systems, três passos devem ser executados:

Exemplo de estrutura de dados

```
EXPORT File_Persons := MODULE
  EXPORT Layout := RECORD
    UNSIGNED8 ID;
    STRING15     FirstName;
    STRING25     LastName;
    STRING15     MiddleName;
    STRING2  NameSuffix;
    STRING8  FileDate;
    UNSIGNED2 BureauCode;
    STRING1  MaritalStatus;
    STRING1  Gender;
    UNSIGNED1 DependentCount;
    STRING8  BirthDate;
    STRING42     StreetAddress;
    STRING20     City;
    STRING2  State;
    STRING5  ZipCode;
    END;
  EXPORT File := DATASET('~CLASS::hmw::Intro::Persons', Layout,FLAT);
END;
```

##	id	firstname	lastname	middl...	n...	filedate	bureaucode	marit...	gender	dep...	birthdate	streetaddress	city	state	zipcode
1	9108218085885411565	Cherianne	Khatchatourian	N		19990922	24		M	0		69 BOULDER RIDGE RD # 25A	HAWKINS	WI	54530
2	16505326057200398078	Muyesser	Raplee	X		20001111	353		F	0		55 SWAMP RD	DISTRICT HEIGHT	MD	20747
3	2454818069645923666	Roselin	Viceconte			19990325	344		F	0	19800113	107 HILL TER	ENTERPRISE	OR	97828
4	15880908289586509107	Inda	Provines			20000909	13		U	0		290 W MOUNT PLEASANT AVE	LAVACA	AR	72941
5	6512705660523829539	Inderdeep	Laurence	D		20001228	344		M	0		44 PROSPECT PL	GREENSBORO	FL	32330
6	9193989543268753887	Chrystine	Mangiapane			19990827	315		F	0	19780306	1806 1ST AVE APT 8F	ARVADA	CO	80007
7	12286552293562700162	Adelene	Stock	R		20000827	252		M	0		1117 FARM RD	DOVER	DE	19901
8	11459575736386985069	Mendy	Rufenblanchette			20000903	24		M	0		3 W 83RD ST APT 4C	WILLIAMSTON	SC	29697
9	8053906447536575038	Lannie	Amerantes	I		20001219	313		U	0		200 W 20TH ST APT 909	CHARLESTON	WV	25312
10	484768759680234166	Tare	Gonyeau	T		19930807	48		F	0	19750801	6 CANDLE CT	EL PASO	TX	79924
11	16156125023194932930	Finney	Aristilde	P		19900621	344		M	0	19560920	222 1ST AVE APT 2B	MACON	GA	31220
12	13804468446718957143	Oreoluwa	Marthaler			19931006	358		F	0	19731201	176 CLAREMONT GDNS	AUBURN	ME	04210
13	11995825474648190448	Surge	Abbottkrepp	D		20000308	13		F	0		22 LE PARC CT	TWINSBURG	OH	44087
14	15714117310244664573	Dave	Mcjury			20001129	238		U	0		510 COOPER RD # 1	TACOMA	WA	98402
15	12587451362606486546	Ramsay	Ping			20001129	238		M	0		404 AVENUE L	MESQUITE	NV	89024

Estrutura RECORD

Uma estrutura **RECORD** define o layout de campos do DATASET.

Nome **:= RECORD**
    campos;
END;

- *Nome* – O nome da estrutura RECORD.
- *campos* – O tipo e o nome de cada campo.

Nota: As palavras-chave RECORD e END podem ser substituídas com chaves ({}) e os delimitadores de campos (;) podem ser substituídos por vírgulas (,).

Declaração DATASET

DATASET introduz um novo arquivo de dados no sistema com o layout *record* especificado.

nome **:= DATASET(** *arquivo, record,* **FLAT[THOR] [***opções***]);**
nome **:= DATASET(***arquivo, record,* **CSV [(** *opções* **)]);**
nome **:= DATASET(***arquivo, record,* **XML(** *caminho,***[***opções***]));**
nome **:= DATASET(***arquivo, record,* **JSON(** *caminho,***[***opções***]));**

- ✓ *nome* – O nome da definição pelo qual o arquivo passará a ser referenciado.
- ✓ *arquivo* – Uma constante string contendo o nome do arquivo lógico.
- ✓ *record* – A estrutura RECORD do dataset.

Nota: Um conjunto de registros pode ser definido inline entre colchetes (indicando uma definição set). Dentro dos colchetes, cada registro é delimitado por chaves ({}) e separado por virgulas. Os campos dentro de cada registo são delimitados por virgula.

```
Names := DATASET([{'John','Jones'}, {'Jane','Smith'}], {STRING first_name, STRING last_name});
```

Atenção! Escopo e Nomes de arquivos lógicos

- Nomes de arquivos sempre começam com um escopo (estrutura de diretórios) e terminando com o nome do arquivo.

- O HPCC busca por **arquivos cujos nomes começam com um escopo padrão (THOR)**:

```
'DIR1::DIR2::NomeArquivo'          //dado isso, HPCC procura por:
'THOR::DIR1::DIR2::NomeArquivo'    //esse arquivo
```

- O sinal de "til" (~) indica a **supressão do escopo padrão**:

```
'~DIR1::DIR2::NomeArquivo'  //dado isso, HPCC procura por:
'DIR1::DIR2:: NomeArquivo'  //esse arquivo
```

Já posso ver os meus dados?

A ação **OUTPUT** grava o *recordset* em um arquivo e formatos especificados.

OUTPUT(*recordset* [,*formato*] [,*arquivo* [,OVERWRITE]]**)**

- *recordset* – O conjunto de registros a processar.
- *formato* – O formato de saída dos registros: uma estrutura RECORD previamente definida, ou um layout de registros "on-the-fly" entre chaves ({ }).
- *arquivo* – Nome opcional do arquivo onde os registros serão gravados. Caso seja omitido, os dados formatados são mostrados na linha de comando ou no ECL IDE.
- OVERWRITE – Permite sobreescrever o arquivo, caso ele já exista.

Exemplos de OUTPUT:

```
OUTPUT(File_Accounts.File);

OUTPUT(Persons,{FirstName, LastName}, NAMED('Names_Only'));

OUTPUT(MyRecordset,,'~CLASS::BMF::NewData', OVERWRITE);

//THOR é o formato padrão, mas também é possível gerar saída como:
OUTPUT(MyRecordset,,'~CLASS::BMF::NewData', CSV);
OUTPUT(MyRecordset,,'~CLASS::BMF::NewData', XML);
OUTPUT(MyRecordset,,'~CLASS::BMF::NewData', JSON);
```

Filtragem simples de dados

- Uma expressão booleana entre parênteses após um Dataset/Recordset é um **filtro**
- Múltiplos filtros podem ser especificados usando uma vírgula (,) ou usando “AND”

```
ValidNames := People(Lastname >= 'T', Lastname < 'U');

ValidTrades := Trades(Rate >= 7);

ValidPeople := People(NOT IsSeniorCitizen AND Lastname < 'U');

ValidPeople2 := People(state IN ['FL','NY']);
```

Operadores de comparação

Equivalência	=
Diferente de	<>
Diferente de	!=
Menor que	<
Maior que	>
Menor ou igual que	<=
Maior ou igual que	>=
Comparação de equivalência	<=> retorna -1, 0, or 1

Operadores aritméticos:

Divisão	/
Divisão Inteira	DIV
Divisão Módulo	%
Multiplicação	*
Adição	+
Subtração	-

Nota: Qualquer divisão por (0) resulta em zero (0).

Esse comportamento pode ser alterado especificando-se

#OPTION ('divideByZero', ‘fail'); //Aborta e reporta erro

Funções de agregação

COUNT(*recordset***)**

COUNT(*listavalores***)**

MAX(*recordset , campo* **)**

MAX(*listavalores***)**

MIN(*recordset , campo* **)**

MIN(*listavalores***)**

SUM(*recordset , campo* **)**

SUM(*listavalores***)**

AVE(*recordset , campo* **)**

AVE(*listavalores***)**

- *recordset* – O set ou conjunto de registros a serem processados.
- *campo* – O campo ou expressão a partir dos quais o valor deve ser calculado.
- *listavalores* – Uma lista de expressões separadas por vírgula a partir dos quais o valor deve ser calculado. Também pode ser um SET de valores.

```
OldCount:=COUNT(People(IsSeniorCitizen));
MaxVal := MAX(People, People.age);
MinVall := MIN(People, People.age);
```

Desafio: Lending Club

Lending Club

- LendingClub é uma empresa americana de empréstimos peer-to-peer, com sede em São Francisco, Califórnia.

- A empresa afirma que US $ 15,98 bilhões em empréstimos foram originados por meio de sua plataforma até 31 de dezembro de 2015.

- Dados: https://www.kaggle.com/ethon0426/lending-club-20072020q1

O que esperamos descobrir

- Identificar um modelo de predição baseado em atributos de operações financeiras que permitam aferir o risco de um pedido de crédito.

- As principais perguntas a serem respondidas são:

  - Quais variáveis podem ser utilizadas para estabelecer com precisão conhecida o risco de crédito?

  - Quais as caracterísiticas do modelo estatístico capaz de prever o risco de crédito?

Fundação Vanzolini

Exercício prático:

Faça a extração do dataset do Lending Club

- Spray
- Estrutura RECORD
- Declaração DATASET

Até a próxima aula!!!