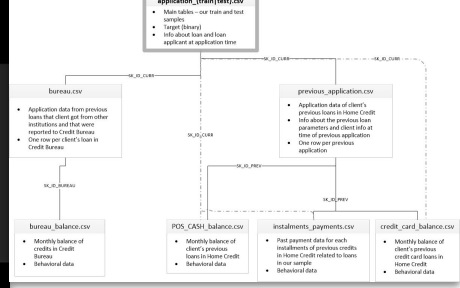
```
## SQL in Rmarkdown

```{sql, connection = con}
-- SQL in Rmarkdown
SELECT *
FROM applications_train
LIMIT 10
```
```

SK_ID_CURR <dbl>	TARGET <dbl>	NAME_CONTRACT_TYPE <chr>	CODE_GENDER <chr>	FLAG_OWN_CAR <chr>	FLAG_OWN_REALTY <chr>
100002	1	Cash loans	M	N	Y

1–10 of 10 rows | 1–6 of 122 columns



SQL for Data Science

Analyzing Home Loans using SQL, R, & dplyr

Difficulty: **Beginner**

Matt Dancho & David Curry

Business Science Learning Lab

Learning Lab Structure

- **Presentation**
  (20 min)
- **Demo's**
  (30 min)
- **Pro-Tips**
  (15 mins)

Matt Dancho

Founder of Business Science, Matt designs and executes educational courses and workshops that deliver immediate value to organizations. His passion is **up-leveling future data scientists** coming from **untraditional backgrounds**.

David Curry

Founder of Sure Optimize, David works with businesses to help improve website performance and SEO using data science. His passion is **ethical Machine Learning initiatives**.

Success Story

Diego Usai

- Past Year, began studying Data Science
- Took Business Science Courses
- Created own website with Project Portfolio

#Business Science Success

“Your courses were a godsend.”

#MACHINE LEARNING #TIDYVERSE #CLASSIFICATION #CHURN #API

Modelling with Tidymodels and Parsnip - A Tidy Approach to a Classification Problem

Recently I have completed the online course Business Analysis With R focused on applied data and business science with R, which introduced me to a couple of new modelling concepts and approaches. One that especially captured my attention is parsnip and its attempt to implement a unified modelling and analysis interface (similar to python’s scikit-learn) to seamlessly access several modelling platforms in R. parsnip is the brainchild of RStudio’s Max Khun (of caret fame) and Davis Vaughan and forms part of tidymodels, a growing ensemble of tools to explore and iterate modelling tasks that shares a common philosophy (and a few libraries) with the tidyverse. ...

DIEGO USAI

#UNSUPERVISED LEARNING #K MEANS CLUSTERING #CUSTOMER SEGMENTATION #MACHINE LEARNING #MARKETING STRATEGIES

A gentle Introduction to Customer Segmentation - Using K-Means Clustering to Understand Marketing Response

Market segmentation refers to the process of dividing a consumer market of existing and/or potential customers into groups (or segments) based on shared attributes, interests, and behaviours. For this mini-project I will use the popular K-Means clustering algorithm to segment customers based on their response to a series of marketing campaigns. The basic concept is that consumers who share common traits would respond to marketing communication in a similar way so that companies can reach out for each group in a relevant and effective way. ...

DIEGO USAI

#MACHINE LEARNING #DATA MINING #DATA CLEANING #BIG DATA #DATA SCIENCE

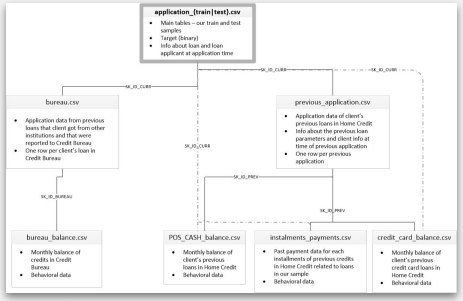
Market Basket Analysis - Part 3 of 3: A Shiny Product Recommender with Improved Collaborative Filtering

My objective for this piece of work is to carry out a Market Basket Analysis as an end-to-end data science project. I have split the output into three parts, of which this is the THIRD and last, that I have organised as follows: In the first chapter, I will source, explore and format a complex dataset suitable for modelling with recommendation algorithms. For the second part, I will apply various machine learning algorithms for Product Recommendation and select the best performing model. ...

DIEGO USAI

diegousai.io

Agenda



- **Business Case Study**
  - Loan Applications
  - Loan Defaults cost BILLIONS
- **Relational Database**
  - Key Concepts
- **Feature Engineering**
  - Core Concepts
- **SQL For Data Science**
  - Databases are fast
  - SQL is Painful
  - Better Way?
- **30-Min Demo**
  - Home Loans
  - SQL in R
  - Feature Engineering
- **Pro-Tips & Learning Guide**
  - Recap + Pro-Tips
  - Learning Plan

Learning Labs PRO

Every 2-Weeks

1-Hour Course

Recordings + Code + Slack

$19/month

university.business-science.io

Lab 20
Explainable Machine Learning

Lab 19
Using Customer Credit Card History for Networks Analysis

Lab 18
Time Series Anomaly Detection with anomalize

Lab 17
Anomaly Detection with H2O Machine Learning

Lab 16
R's Optimization Toolchain, Part 2 - Nonlinear Programming

Lab 15
R's Optimization Toolchain, Part 1 - Linear Programming

Continuous Learning
Jet Fuel for your Brain

Home Loan Applications

Business Case

Loan Defaults Cost Billions

Home Loans

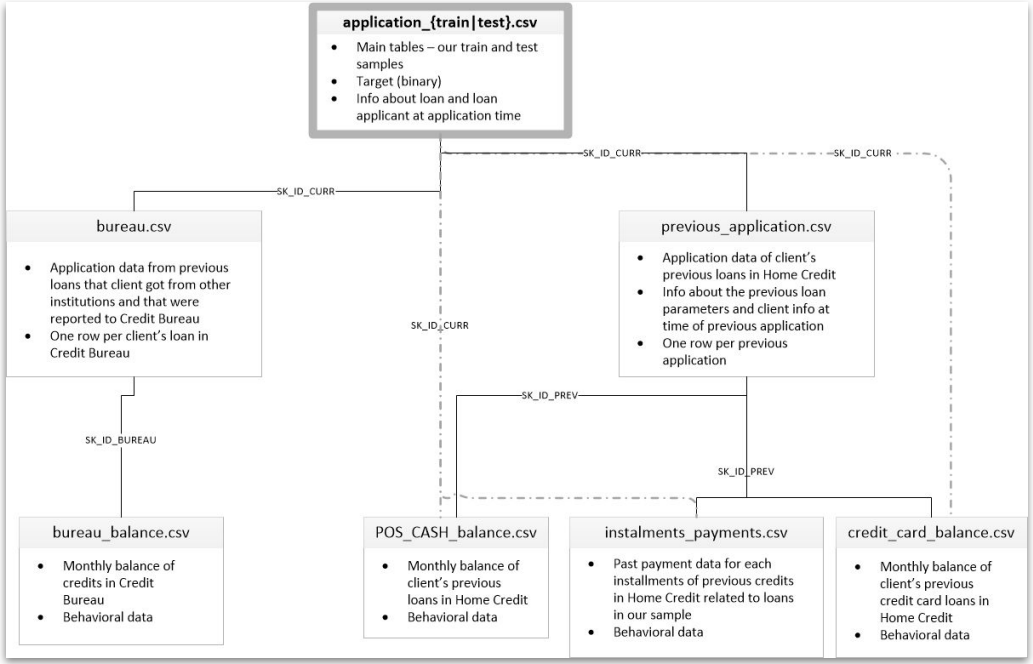
1. Banks lend **Billions**
2. **Approval process** - Want strict, but not too strict
3. If we can better **predict** defaults, we can **save** billions

SQL Database + Feature Engineering

SQL Database

1. Data Scientists interact with data stored in **SQL for their Analysis 99.9% of time**

2. This complex Network is SUPER Useful if we know how to use it… **Feature Engineering!**



Relational Database Basics

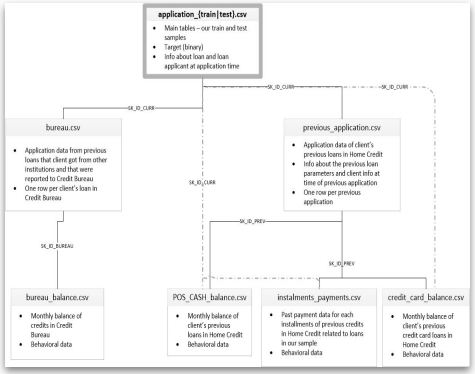
80/20 Concepts

Types of Databases

1

SQL

Structured (Most BI Databases)



MS SQL
MySQL
PostGreSQL

2

NoSQL

Unstructured Data (Apps)

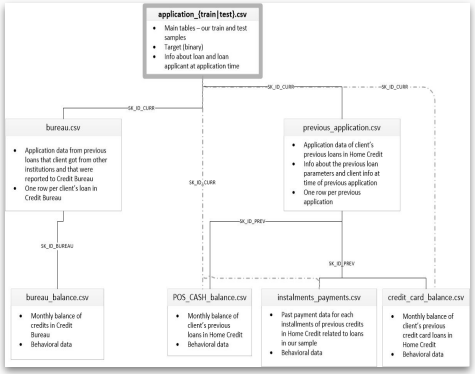
MongoDB
AWS DynamoDB

Types of Databases

1

SQL

Structured (Most BI Databases)

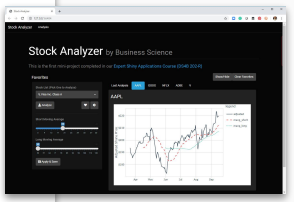


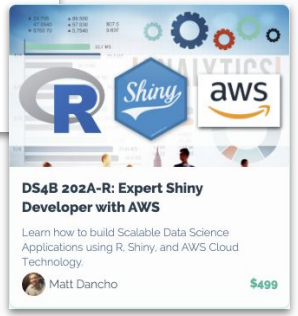
MS SQL
MySQL
PostGreSQL

2

NoSQL

Unstructured Data (Apps)



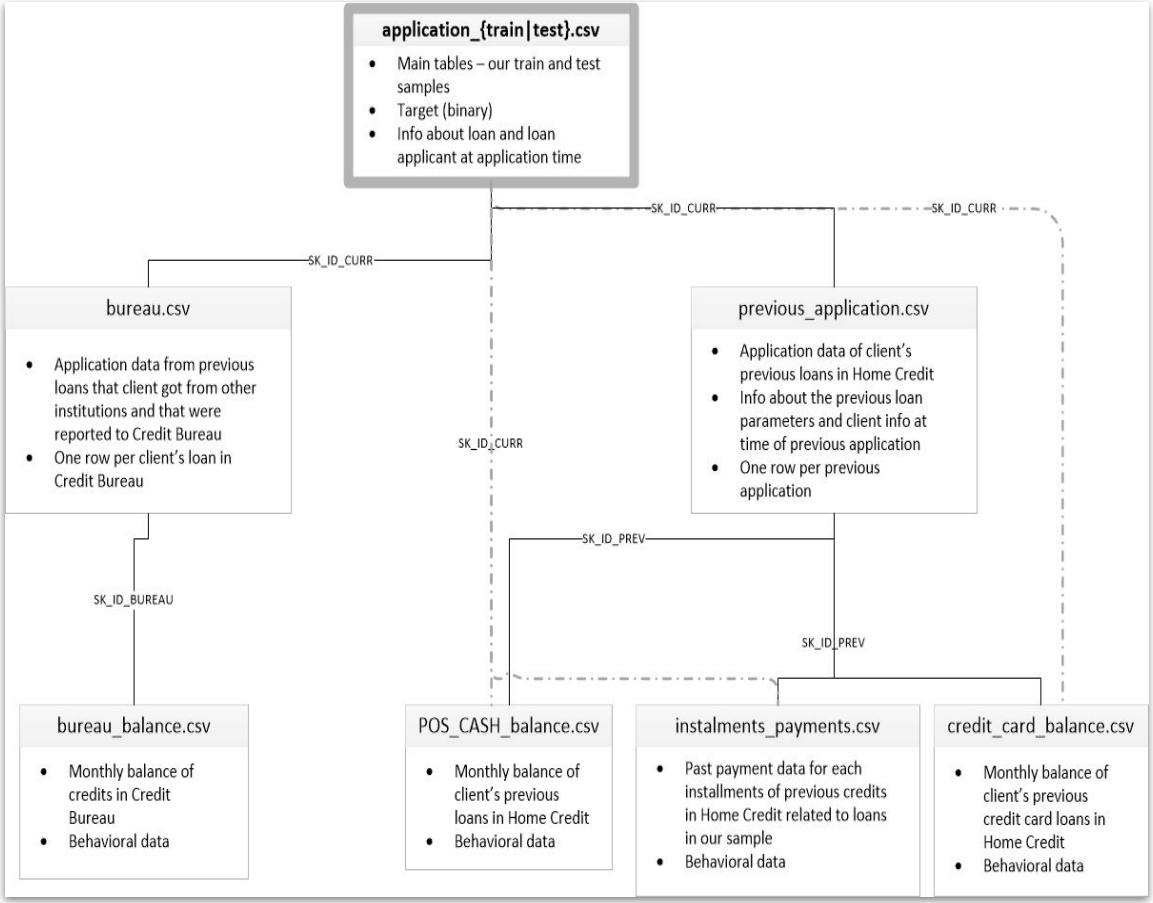


MongoDB
AWS DynamoDB

SQL Database

1. Data stored in **SQL Tables**
2. **Relationships** between tables linked with common field called an ID (Primary Key)
3. Data Scientists can use tables to **generate features** & model business problem

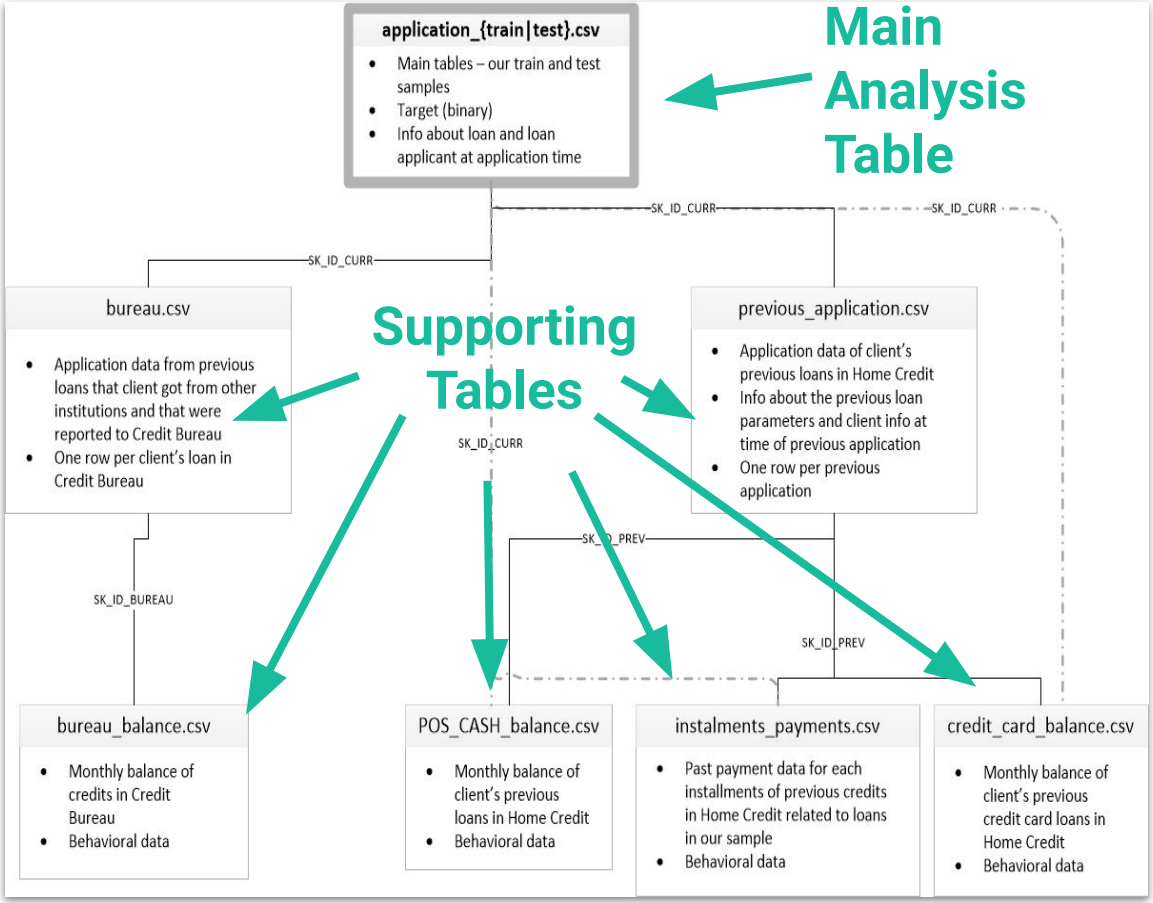
Key Point - Can use relationships between tables to **generate critical features** to higher level data model



SQL Database

1. Data stored in **SQL Tables**
2. **Relationships** between tables linked with common field called an ID (Primary Key)
3. Data Scientists can use tables to **generate features** & model business problem

Key Point - Can use relationships between tables to **generate critical features** to higher level data model



Feature Engineering

Core Concepts

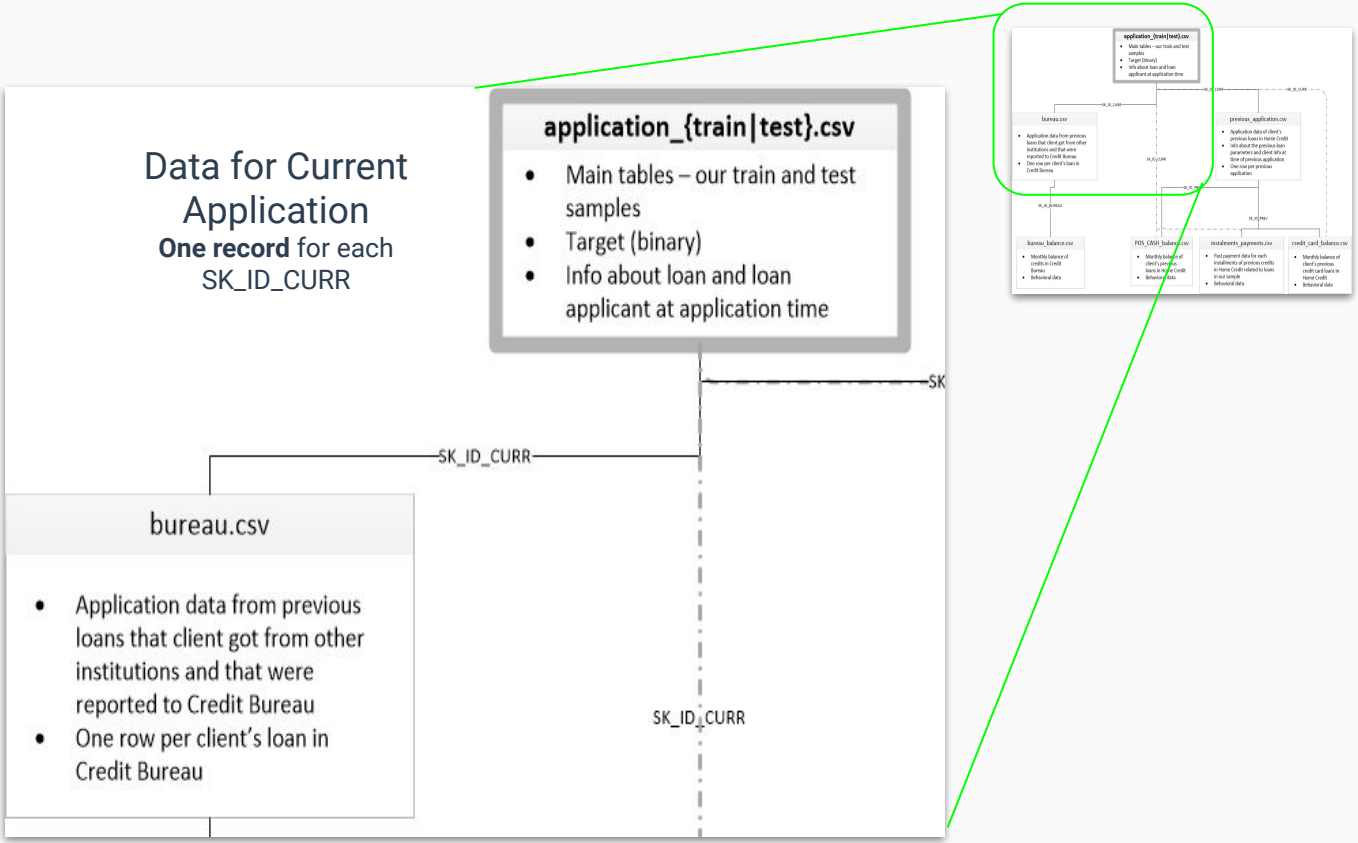
Feature Engineering

Data for Current Application

One record for each SK_ID_CURR

Data from Credit Bureau

Multiple records (previous applications) for each SK_ID_CURR



Feature Engineering

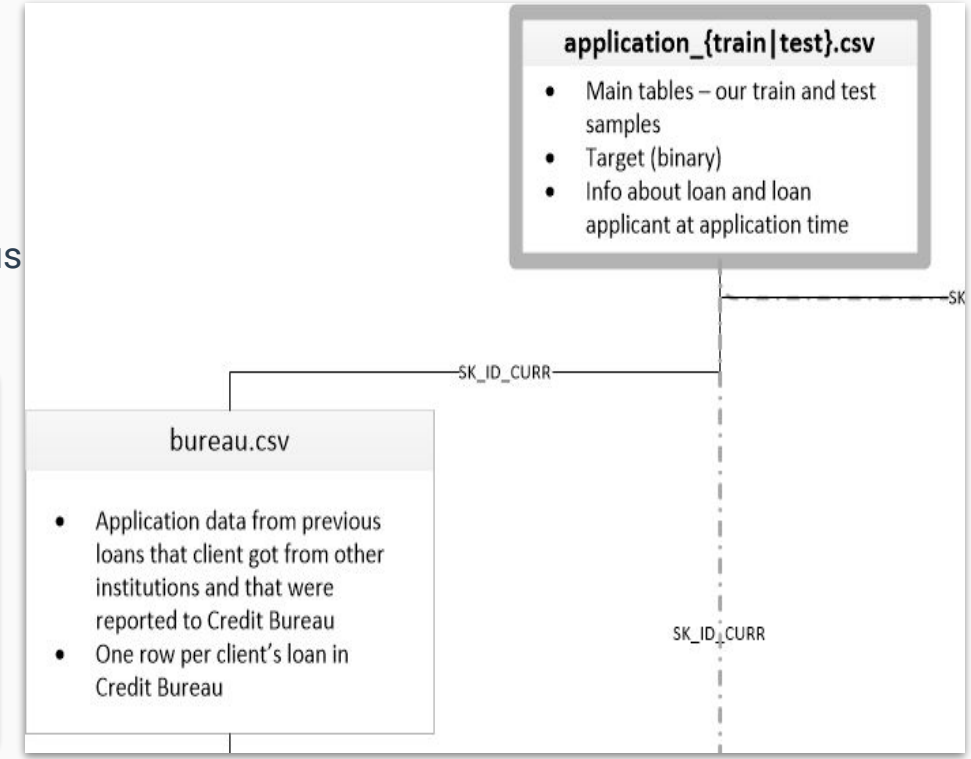
Question:
Does “Days Credit” for previous loan applications influence Loan Default?

Days Credit for Previous Loan Applications

Multiple Prior applications

	SK_ID_CURR	DAYS_CREDIT
	<dbl>	<dbl>
1	215354	-497
2	215354	-208
3	215354	-203
4	215354	-203
5	215354	-629
6	215354	-273
7	215354	-43
8	162297	-1896
9	162297	-1146
10	162297	-1146

… with more rows



Feature Engineering

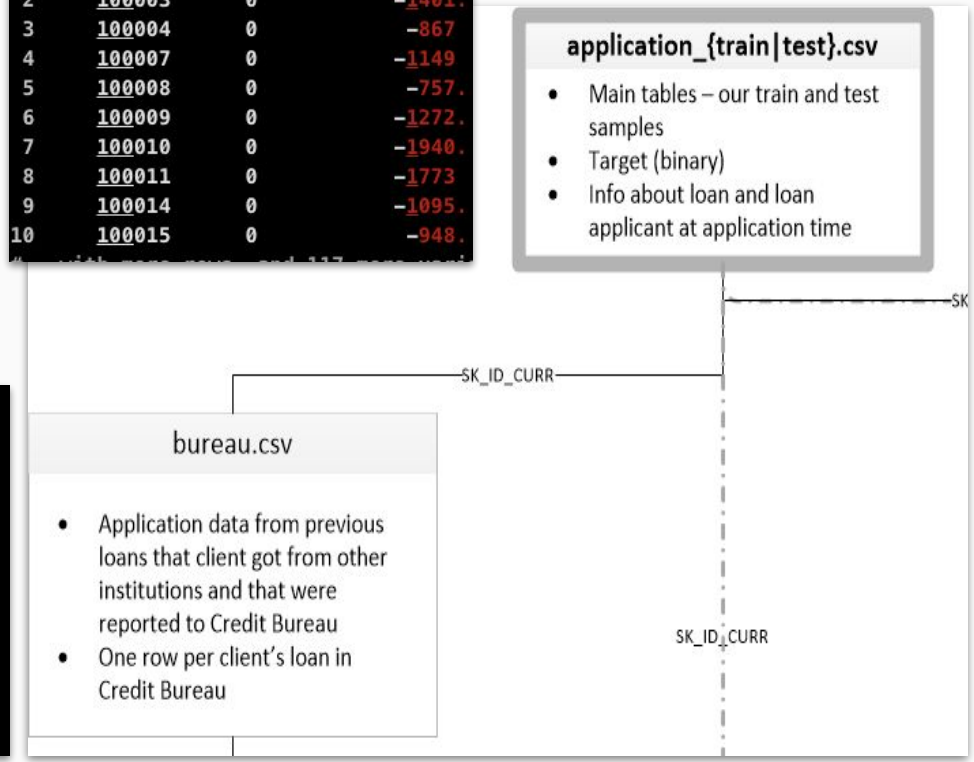
Compute Average

Mean Days Credit & Join with Application_Train

SK_ID_CURR	TARGET	mean_days_credit
<dbl>	<dbl>	<dbl>
1 100002	1	-874
2 100003	0	-1401.
3 100004	0	-867
4 100007	0	-1149
5 100008	0	-757.
6 100009	0	-1272.
7 100010	0	-1940.
8 100011	0	-1773
9 100014	0	-1095.
10 100015	0	-948.

SK_ID_CURR	DAYS_CREDIT
<dbl>	<dbl>
1 215354	-497
2 215354	-208
3 215354	-203
4 215354	-203
5 215354	-629
6 215354	-273
7 215354	-43
8 162297	-1896
9 162297	-1146
10 162297	-1146

… with more rows



Feature Engineering

Compute Average

Mean Days Credit & Join with Application_Train

SK_ID_CURR	TARGET	mean_days_credit
<dbl>	<dbl>	<dbl>
1 100002	1	-874
2 100003	0	-1401.
3 100004	0	-867
4 100007	0	-1149
5 100008	0	-757.
6 100009	0	-1272.
7 100010	0	-1940.
8 100011	0	-1773
9 100014	0	-1095.
10 100015	0	-948.

Correlation

	TARGET	mean_days_credit
TARGET	1.00000000	0.08972897
mean_days_credit	0.08972897	1.00000000

SK_ID_CURR	DAYS_CREDIT
<dbl>	<dbl>
1 215354	-497
2 215354	-208
3 215354	-203
4 215354	-203
5 215354	-629
6 215354	-273
7 215354	-43
8 162297	-1896
9 162297	-1146
10 162297	-1146

… with more rows

application_{train|test}.csv

- Main tables – our train and test samples
- Target (binary)
- Info about loan and loan applicant at application time

SK_ID_CURR

SK_ID_CURR

bureau.csv

- Application data from previous loans that client got from other institutions and that were reported to Credit Bureau
- One row per client's loan in Credit Bureau

SQL for Data Science

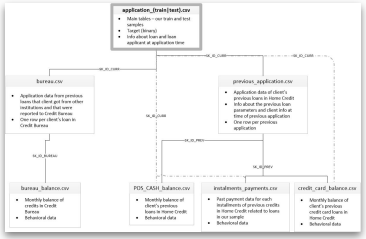
Why Use Databases for Data Science?

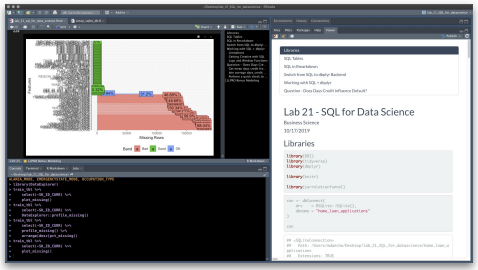
Databases are Fast

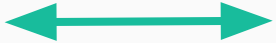
SQL & NoSQL are **optimized** to handle data & perform aggregations, filtering, etc.

Data takes a long time to **transfer** between machines.

SQL for Data Science Workflow



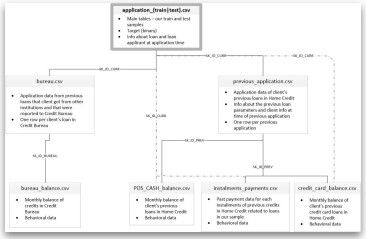


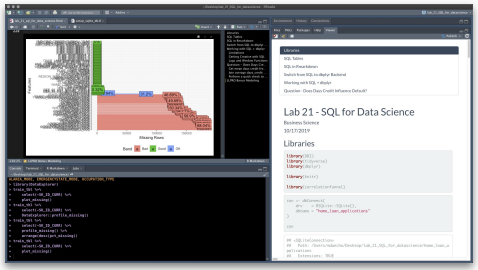


Home Loans Database
(SQL Database)

Matt's Computer
(MacBook Pro)

SQL for Data Science Workflow

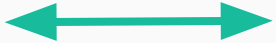




Maximize
Simple but
Expensive
Operations

Home Loans
Database
(SQL Database)

Minimize
Data
Transfer via
Aggregation



Matt's Computer
(MacBook Pro)

Perform
Complex
Data Science

SQL is painful

It costs you **time** to write SQL,

it's prone to **errors** & **difficult** to learn

If only there was...
A better way


Wise owl says,
“Hoot. Hoot.
Rewrite your previous
statement with **dplyr**.”

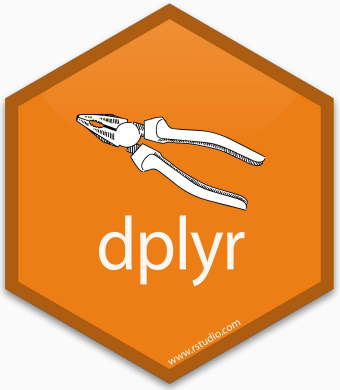
If only there was...
A dplyr way

dplyr

SQL Translation

If you know **dplyr**, you know SQL

(You just may not know this yet)



Goal: Feature Engineering

Compute Average

Mean Days Credit & Join with Application_Train

SK_ID_CURR	TARGET	mean_days_credit
<dbl>	<dbl>	<dbl>
1 100002	1	-874
2 100003	0	-1401.
3 100004	0	-867
4 100007	0	-1149
5 100008	0	-757.
6 100009	0	-1272.
7 100010	0	-1940.
8 100011	0	-1773
9 100014	0	-1095.
10 100015	0	-948.

Correlation

	TARGET	mean_days_credit
TARGET	1.00000000	0.08972897
mean_days_credit	0.08972897	1.00000000

SK_ID_CURR	DAYS_CREDIT
<dbl>	<dbl>
1 215354	-497
2 215354	-208
3 215354	-203
4 215354	-203
5 215354	-629
6 215354	-273
7 215354	-43
8 162297	-1896
9 162297	-1146
10 162297	-1146

… with more rows

application_{train|test}.csv

- Main tables – our train and test samples
- Target (binary)
- Info about loan and loan applicant at application time

SK_ID_CURR

SK_ID_CURR

bureau.csv

- Application data from previous loans that client got from other institutions and that were reported to Credit Bureau
- One row per client's loan in Credit Bureau

SQL Translation with dplyr

Database Table Connection

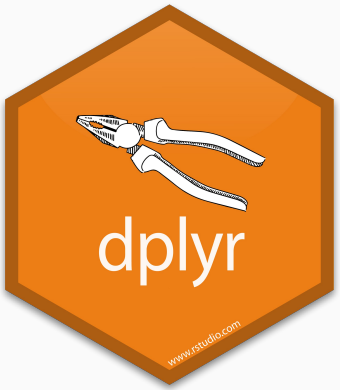
```
days_credit_query <- tbl(con, "bureau") %>%

    # Selet columns
    select(SK_ID_CURR, DAYS_CREDIT) %>%

    # Group by SK_ID_CURR and calculate average days credit
    group_by(SK_ID_CURR) %>%
    summarise(mean_days_credit = mean(DAYS_CREDIT, na.rm = T)) %>%
    ungroup() %>%

    # Arrange Descending by mean
    arrange(desc(mean_days_credit))
```

dplyr operations



SQL Translation with dplyr

Database Table Connection

```
days_credit_query <- tbl(con, "bureau") %>%

    # Selet columns
    select(SK_ID_CURR, DAYS_CREDIT) %>%

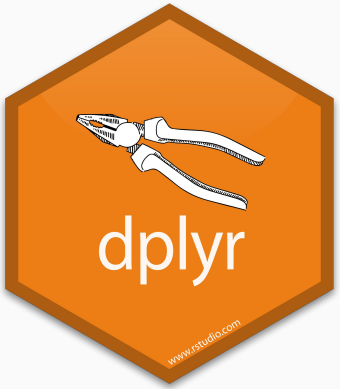
    # Group by SK_ID_CURR and calculate average days credit
    group_by(SK_ID_CURR) %>%
    summarise(mean_days_credit = mean(DAYS_CREDIT, na.rm = T)) %>%
    ungroup() %>%

    # Arrange Descending by mean
    arrange(desc(mean_days_credit))
```

dplyr operations

dplyr translates to SQL

```
<SQL>
SELECT `SK_ID_CURR`, AVG(`DAYS_CREDIT`) AS `mean_days_credit`
FROM (SELECT `SK_ID_CURR`, `DAYS_CREDIT`
FROM `bureau`)
GROUP BY `SK_ID_CURR`
ORDER BY `mean_days_credit` DESC
```



SQL Translation with dplyr

	SK_ID_CURR	DAYS_CREDIT
	<dbl>	<dbl>
1	215354	-497
2	215354	-208
3	215354	-203
4	215354	-203
5	215354	-629
6	215354	-273
7	215354	-43
8	162297	-1896
9	162297	-1146
10	162297	-1146

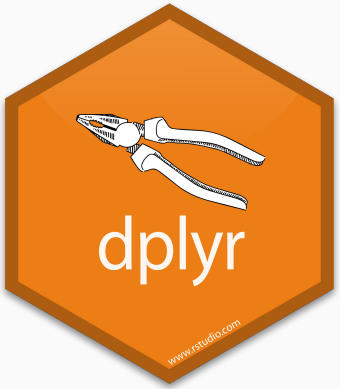
… with more rows

	SK_ID_CURR	mean_days_credit
	<dbl>	<dbl>
1	182735	0
2	418331	0
3	231365	-2
4	301765	-3
5	136099	-4
6	150407	-4
7	342009	-4
8	396255	-4
9	175871	-5
10	208781	-5

… with more rows

Not Impressed?

We still didn't do the join

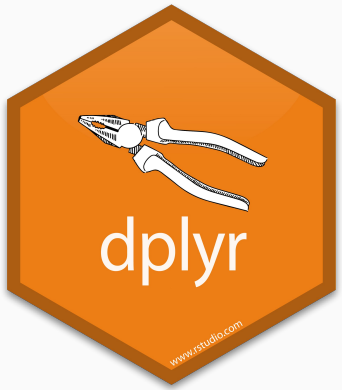


SQL Translation with dplyr

```
`LHS`.`FLOORSMAX_MODE` AS `FLOORSMAX_MODE`, `LHS`.`FLOORSMIN_MODE` AS `FLOORSMIN_MODE`, `LHS`.`LANDAREA_MODE`
AS `LANDAREA_MODE`, `LHS`.`LIVINGAPARTMENTS_MODE` AS `LIVINGAPARTMENTS_MODE`, `LHS`.`LIVINGAREA_MODE` AS
`LIVINGAREA_MODE`, `LHS`.`NONLIVINGAPARTMENTS_MODE` AS `NONLIVINGAPARTMENTS_MODE`, `LHS`.`NONLIVINGAREA_MODE`
AS `NONLIVINGAREA_MODE`, `LHS`.`APARTMENTS_MEDI` AS `APARTMENTS_MEDI`, `LHS`.`BASEMENTAREA_MEDI` AS
`BASEMENTAREA_MEDI`, `LHS`.`YEARS_BEGINEXPLUATATION_MEDI` AS `YEARS_BEGINEXPLUATATION_MEDI`,
`LHS`.`YEARS_BUILD_MEDI` AS `YEARS_BUILD_MEDI`, `LHS`.`COMMONAREA_MEDI` AS `COMMONAREA_MEDI`,
`LHS`.`ELEVATORS_MEDI` AS `ELEVATORS_MEDI`, `LHS`.`ENTRANCES_MEDI` AS `ENTRANCES_MEDI`,
`LHS`.`FLOORSMAX_MEDI` AS `FLOORSMAX_MEDI`, `LHS`.`FLOORSMIN_MEDI` AS `FLOORSMIN_MEDI`, `LHS`.`LANDAREA_MEDI`
AS `LANDAREA_MEDI`, `LHS`.`LIVINGAPARTMENTS_MEDI` AS `LIVINGAPARTMENTS_MEDI`, `LHS`.`LIVINGAREA_MEDI` AS
`LIVINGAREA_MEDI`, `LHS`.`NONLIVINGAPARTMENTS_MEDI` AS `NONLIVINGAPARTMENTS_MEDI`, `LHS`.`NONLIVINGAREA_MEDI`
AS `NONLIVINGAREA_MEDI`, `LHS`.`FONDKAPREMONT_MODE` AS `FONDKAPREMONT_MODE`, `LHS`.`HOUSETYPE_MODE` AS
`HOUSETYPE_MODE`, `LHS`.`TOTALAREA_MODE` AS `TOTALAREA_MODE`, `LHS`.`WALLSMATERIAL_MODE` AS
`WALLSMATERIAL_MODE`, `LHS`.`EMERGENCYSTATE_MODE` AS `EMERGENCYSTATE_MODE`, `LHS`.`OBS_30_CNT_SOCIAL_CIRCLE`
AS `OBS_30_CNT_SOCIAL_CIRCLE`, `LHS`.`DEF_30_CNT_SOCIAL_CIRCLE` AS `DEF_30_CNT_SOCIAL_CIRCLE`,
`LHS`.`OBS_60_CNT_SOCIAL_CIRCLE` AS `OBS_60_CNT_SOCIAL_CIRCLE`, `LHS`.`DEF_60_CNT_SOCIAL_CIRCLE` AS
`DEF_60_CNT_SOCIAL_CIRCLE`, `LHS`.`DAYS_LAST_PHONE_CHANGE` AS `DAYS_LAST_PHONE_CHANGE`,
`LHS`.`FLAG_DOCUMENT_2` AS `FLAG_DOCUMENT_2`, `LHS`.`FLAG_DOCUMENT_3` AS `FLAG_DOCUMENT_3`,
`LHS`.`FLAG_DOCUMENT_4` AS `FLAG_DOCUMENT_4`, `LHS`.`FLAG_DOCUMENT_5` AS `FLAG_DOCUMENT_5`,
`LHS`.`FLAG_DOCUMENT_6` AS `FLAG_DOCUMENT_6`, `LHS`.`FLAG_DOCUMENT_7` AS `FLAG_DOCUMENT_7`,
`LHS`.`FLAG_DOCUMENT_8` AS `FLAG_DOCUMENT_8`, `LHS`.`FLAG_DOCUMENT_9` AS `FLAG_DOCUMENT_9`,
`LHS`.`FLAG_DOCUMENT_10` AS `FLAG_DOCUMENT_10`, `LHS`.`FLAG_DOCUMENT_11` AS `FLAG_DOCUMENT_11`,
`LHS`.`FLAG_DOCUMENT_12` AS `FLAG_DOCUMENT_12`, `LHS`.`FLAG_DOCUMENT_13` AS `FLAG_DOCUMENT_13`,
`LHS`.`FLAG_DOCUMENT_14` AS `FLAG_DOCUMENT_14`, `LHS`.`FLAG_DOCUMENT_15` AS `FLAG_DOCUMENT_15`,
`LHS`.`FLAG_DOCUMENT_16` AS `FLAG_DOCUMENT_16`, `LHS`.`FLAG_DOCUMENT_17` AS `FLAG_DOCUMENT_17`,
`LHS`.`FLAG_DOCUMENT_18` AS `FLAG_DOCUMENT_18`, `LHS`.`FLAG_DOCUMENT_19` AS `FLAG_DOCUMENT_19`,
`LHS`.`FLAG_DOCUMENT_20` AS `FLAG_DOCUMENT_20`, `LHS`.`FLAG_DOCUMENT_21` AS `FLAG_DOCUMENT_21`,
`LHS`.`AMT_REQ_CREDIT_BUREAU_HOUR` AS `AMT_REQ_CREDIT_BUREAU_HOUR`, `LHS`.`AMT_REQ_CREDIT_BUREAU_DAY` AS
`AMT_REQ_CREDIT_BUREAU_DAY`, `LHS`.`AMT_REQ_CREDIT_BUREAU_WEEK` AS `AMT_REQ_CREDIT_BUREAU_WEEK`,
`LHS`.`AMT_REQ_CREDIT_BUREAU_MON` AS `AMT_REQ_CREDIT_BUREAU_MON`, `LHS`.`AMT_REQ_CREDIT_BUREAU_QRT` AS
`AMT_REQ_CREDIT_BUREAU_QRT`, `LHS`.`AMT_REQ_CREDIT_BUREAU_YEAR` AS `AMT_REQ_CREDIT_BUREAU_YEAR`,
`RHS`.`mean_days_credit` AS `mean_days_credit`
FROM `applications_train` AS `LHS`
LEFT JOIN (SELECT `SK_ID_CURR`, AVG(`DAYS_CREDIT`) AS `mean_days_credit`
FROM (SELECT `SK_ID_CURR`, `DAYS_CREDIT`
FROM `bureau`)
GROUP BY `SK_ID_CURR`
ORDER BY `mean_days_credit` DESC) AS `RHS`
ON (`LHS`.`SK_ID_CURR` = `RHS`.`SK_ID_CURR`)
)
WHERE (NOT(((`mean_days_credit`) IS NULL)))
```

SQL LEFT JOIN

Using the
application_train
Table
& the newly created
mean_days_credit



SQL Translation with dplyr

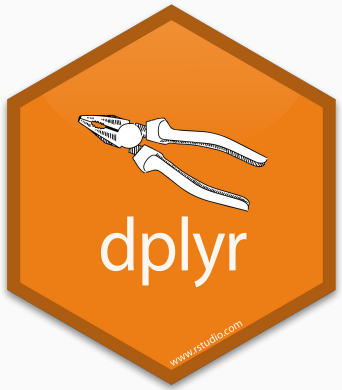
```r
applications_days_credit_joined_query <- tbl(con, "applications_train") %>%
    left_join(days_credit_query) %>%
    filter(!is.na(mean_days_credit)) %>%
    select(SK_ID_CURR, TARGET, mean_days_credit, everything())

applications_days_credit_joined_query %>% show_query()
```

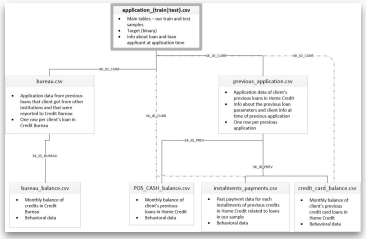
```
`LHS`.`FLOORSMAX_MODE` AS `FLOORSMAX_MODE`, `LHS`.`FLOORSMIN_MODE` AS `FLOORSMIN_MODE`, `LHS`.`LANDAREA_MODE`
AS `LANDAREA_MODE`, `LHS`.`LIVINGAPARTMENTS_MODE` AS `LIVINGAPARTMENTS_MODE`, `LHS`.`LIVINGAREA_MODE` AS
`LIVINGAREA_MODE`, `LHS`.`NONLIVINGAPARTMENTS_MODE` AS `NONLIVINGAPARTMENTS_MODE`, `LHS`.`NONLIVINGAREA_MODE`
AS `NONLIVINGAREA_MODE`, `LHS`.`APARTMENTS_MEDI` AS `APARTMENTS_MEDI`, `LHS`.`BASEMENTAREA_MEDI` AS
`BASEMENTAREA_MEDI`, `LHS`.`YEARS_BEGINEXPLUATATION_MEDI` AS `YEARS_BEGINEXPLUATATION_MEDI`,
`LHS`.`YEARS_BUILD_MEDI` AS `YEARS_BUILD_MEDI`, `LHS`.`COMMONAREA_MEDI` AS `COMMONAREA_MEDI`,
`LHS`.`ELEVATORS_MEDI` AS `ELEVATORS_MEDI`, `LHS`.`ENTRANCES_MEDI` AS `ENTRANCES_MEDI`,
`LHS`.`FLOORSMAX_MEDI` AS `FLOORSMAX_MEDI`, `LHS`.`FLOORSMIN_MEDI` AS `FLOORSMIN_MEDI`, `LHS`.`LANDAREA_MEDI`
AS `LANDAREA_MEDI`, `LHS`.`LIVINGAPARTMENTS_MEDI` AS `LIVINGAPARTMENTS_MEDI`, `LHS`.`LIVINGAREA_MEDI` AS
`LIVINGAREA_MEDI`, `LHS`.`NONLIVINGAPARTMENTS_MEDI` AS `NONLIVINGAPARTMENTS_MEDI`, `LHS`.`NONLIVINGAREA_MEDI`
AS `NONLIVINGAREA_MEDI`, `LHS`.`FONDKAPREMONT_MODE` AS `FONDKAPREMONT_MODE`, `LHS`.`HOUSETYPE_MODE` AS
`HOUSETYPE_MODE`, `LHS`.`TOTALAREA_MODE` AS `TOTALAREA_MODE`, `LHS`.`WALLSMATERIAL_MODE` AS
`WALLSMATERIAL_MODE`, `LHS`.`EMERGENCYSTATE_MODE` AS `EMERGENCYSTATE_MODE`, `LHS`.`OBS_30_CNT_SOCIAL_CIRCLE`
AS `OBS_30_CNT_SOCIAL_CIRCLE`, `LHS`.`DEF_30_CNT_SOCIAL_CIRCLE` AS `DEF_30_CNT_SOCIAL_CIRCLE`,
`LHS`.`OBS_60_CNT_SOCIAL_CIRCLE` AS `OBS_60_CNT_SOCIAL_CIRCLE`, `LHS`.`DEF_60_CNT_SOCIAL_CIRCLE` AS
`DEF_60_CNT_SOCIAL_CIRCLE`, `LHS`.`DAYS_LAST_PHONE_CHANGE` AS `DAYS_LAST_PHONE_CHANGE`,
`LHS`.`FLAG_DOCUMENT_2` AS `FLAG_DOCUMENT_2`, `LHS`.`FLAG_DOCUMENT_3` AS `FLAG_DOCUMENT_3`,
`LHS`.`FLAG_DOCUMENT_4` AS `FLAG_DOCUMENT_4`, `LHS`.`FLAG_DOCUMENT_5` AS `FLAG_DOCUMENT_5`,
`LHS`.`FLAG_DOCUMENT_6` AS `FLAG_DOCUMENT_6`, `LHS`.`FLAG_DOCUMENT_7` AS `FLAG_DOCUMENT_7`,
`LHS`.`FLAG_DOCUMENT_8` AS `FLAG_DOCUMENT_8`, `LHS`.`FLAG_DOCUMENT_9` AS `FLAG_DOCUMENT_9`,
`LHS`.`FLAG_DOCUMENT_10` AS `FLAG_DOCUMENT_10`, `LHS`.`FLAG_DOCUMENT_11` AS `FLAG_DOCUMENT_11`,
`LHS`.`FLAG_DOCUMENT_12` AS `FLAG_DOCUMENT_12`, `LHS`.`FLAG_DOCUMENT_13` AS `FLAG_DOCUMENT_13`,
`LHS`.`FLAG_DOCUMENT_14` AS `FLAG_DOCUMENT_14`, `LHS`.`FLAG_DOCUMENT_15` AS `FLAG_DOCUMENT_15`,
`LHS`.`FLAG_DOCUMENT_16` AS `FLAG_DOCUMENT_16`, `LHS`.`FLAG_DOCUMENT_17` AS `FLAG_DOCUMENT_17`,
`LHS`.`FLAG_DOCUMENT_18` AS `FLAG_DOCUMENT_18`, `LHS`.`FLAG_DOCUMENT_19` AS `FLAG_DOCUMENT_19`,
`LHS`.`FLAG_DOCUMENT_20` AS `FLAG_DOCUMENT_20`, `LHS`.`FLAG_DOCUMENT_21` AS `FLAG_DOCUMENT_21`,
`LHS`.`AMT_REQ_CREDIT_BUREAU_HOUR` AS `AMT_REQ_CREDIT_BUREAU_HOUR`, `LHS`.`AMT_REQ_CREDIT_BUREAU_DAY` AS
`AMT_REQ_CREDIT_BUREAU_DAY`, `LHS`.`AMT_REQ_CREDIT_BUREAU_WEEK` AS `AMT_REQ_CREDIT_BUREAU_WEEK`,
`LHS`.`AMT_REQ_CREDIT_BUREAU_MON` AS `AMT_REQ_CREDIT_BUREAU_MON`, `LHS`.`AMT_REQ_CREDIT_BUREAU_QRT` AS
`AMT_REQ_CREDIT_BUREAU_QRT`, `LHS`.`AMT_REQ_CREDIT_BUREAU_YEAR` AS `AMT_REQ_CREDIT_BUREAU_YEAR`,
`RHS`.`mean_days_credit` AS `mean_days_credit`
FROM `applications_train` AS `LHS`
LEFT JOIN (SELECT `SK_ID_CURR`, AVG(`DAYS_CREDIT`) AS `mean_days_credit`
FROM (SELECT `SK_ID_CURR`, `DAYS_CREDIT`
FROM `bureau`)
GROUP BY `SK_ID_CURR`
ORDER BY `mean_days_credit` DESC) AS `RHS`
ON (`LHS`.`SK_ID_CURR` = `RHS`.`SK_ID_CURR`)
)
WHERE (NOT(((`mean_days_credit`) IS NULL)))
```

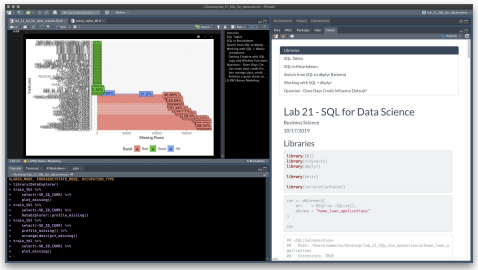
Dplyr SQL Translation

Using the
application_train
Table
& the newly created
mean_days_credit



SQL for Data Science Workflow

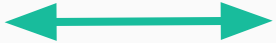




Maximize
Simple but
Expensive
Operations

Home Loans
Database
(SQL Database)

Minimize
Data
Transfer via
Aggregation



Matt's Computer
(MacBook Pro)

Perform
Complex
Data Science

30-Min Demo

Home Loan Applications

PRO-TIPS

Yeahhhhhh!

Pro-Tips

```
days_credit_query <- tbl(con, "bureau") %>%

    # Selet columns
    select(SK_ID_CURR, DAYS_CREDIT) %>%

    # Group by SK_ID_CURR and calculate average days credit
    group_by(SK_ID_CURR) %>%
    summarise(mean_days_credit = mean(DAYS_CREDIT, na.rm = T)) %>%
    ungroup() %>%

    # Arrange Descending by mean
    arrange(desc(mean_days_credit))

days_credit_query %>% show_query()
```

```
<SQL>
SELECT `SK_ID_CURR`, AVG(`DAYS_CREDIT`) AS `mean_days_credit`
FROM (SELECT `SK_ID_CURR`, `DAYS_CREDIT`
FROM `bureau`)
GROUP BY `SK_ID_CURR`
ORDER BY `mean_days_credit` DESC
```

#1. Don’t Memorize SQL

This will drive you MAD

#2. Learn dplyr, then translate to SQL

Using show_query()

3. Fill in the blanks by Googling SQL Commands

Use sql() to insert into your dplyr code

What We Just Did

And how WE did it!

SQL for Data Science Workflow

Step-By-Step

dplyr

Feature Engineering

SQL Query

Aggregations & Joins

Machine Learning, Visualization & Apps

Predict & Explain
Loan Default Risk

Data Manipulation & Visualization


```r
```{r, paged.print = FALSE}
days_credit_query <- tbl(con, "bureau") %>%

    # Selet columns
    select(SK_ID_CURR, DAYS_CREDIT) %>%

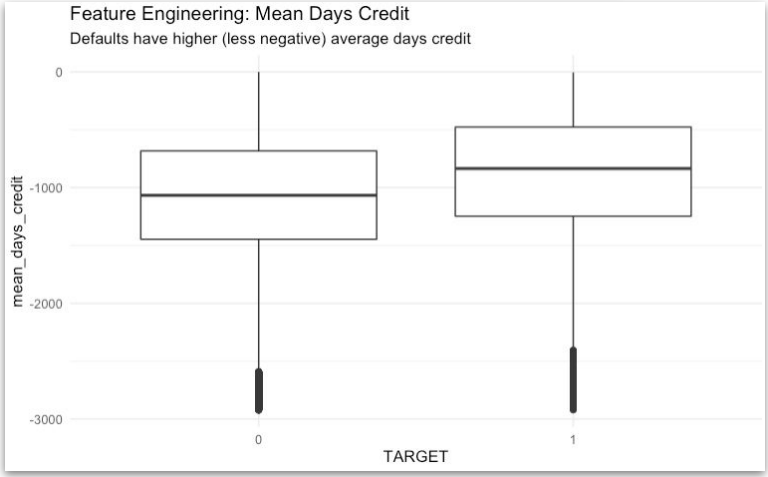
    # Group by SK_ID_CURR and calculate average days credit
    group_by(SK_ID_CURR) %>%
    summarise(mean_days_credit = mean(DAYS_CREDIT, na.rm = T)) %>%
    ungroup() %>%

    # Arrange Descending by mean
    arrange(desc(mean_days_credit))

days_credit_query %>% show_query()
```
```

```
<SQL>
SELECT `SK_ID_CURR`, AVG(`DAYS_CREDIT`) AS `mean_days_credit`
FROM (SELECT `SK_ID_CURR`, `DAYS_CREDIT`
FROM `bureau`)
GROUP BY `SK_ID_CURR`
ORDER BY `mean_days_credit` DESC
```

101 & 201



Preprocessing

101 & 201

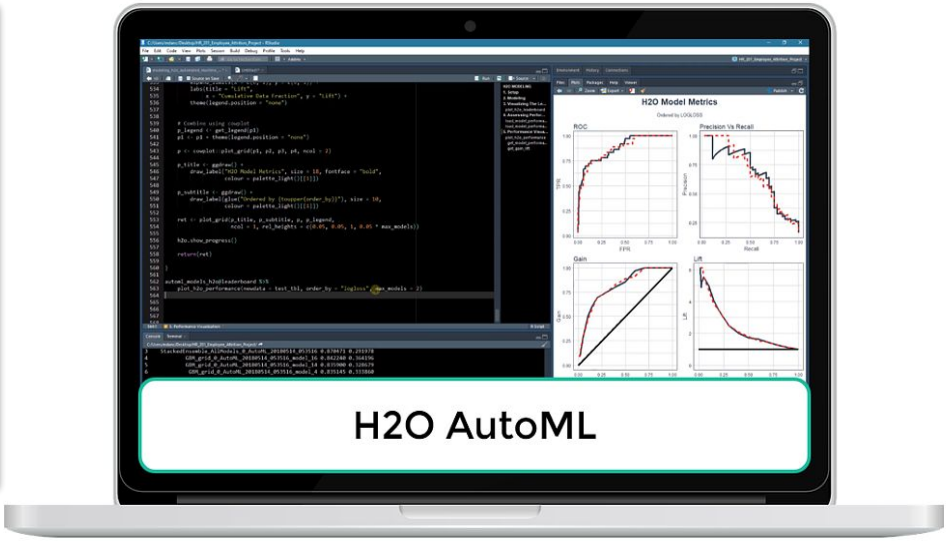
```
64  # 3.1 Preprocessing -----
65  set.seed(123)
66  rsample_splits <- initial_split(customer_churn_raw_tbl, prop = 0.8)
67
68  rec_obj <- recipe(Churn ~ ., data = training(rsample_splits)) %>%
69      step_mutate(TotalCharges = ifelse(is.na(TotalCharges), 0, TotalCharges)) %>%
70      step_rm(customerID) %>%
71      step_string2factor(all_nominal()) %>%
72      prep()
73
74  train_tbl <- bake(rec_obj, training(rsample_splits))
75  test_tbl  <- bake(rec_obj, testing(rsample_splits))
76
77  train_tbl
78
```

Advanced Machine Learning

H2O.ai

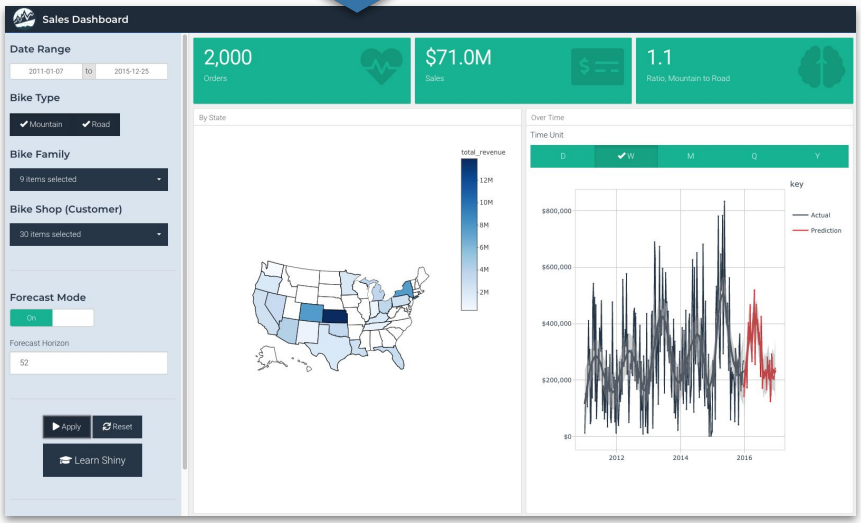
201

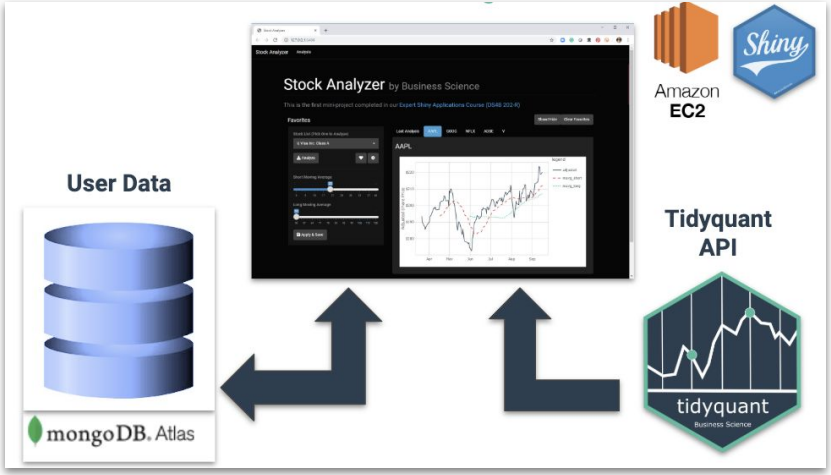
```
> h2o.predict(h2o_model, newdata = as.h2o(credit_card_group_tbl)) %>%
+   as_tibble()
  |======================================================================| 100%
  |======================================================================| 100%
# A tibble: 1,125 x 7
   predict      p1        p2        p3      p4        p5   Other
   <fct>     <dbl>     <dbl>     <dbl>   <dbl>     <dbl>   <dbl>
 1 Other   0       0.0000704 0.0228    0       0         0.977
 2 Other   0.0232  0.0000717 0.0000553 0       0.00376   0.973
 3 Other   0       0.0000737 0.0238    0       0.000107  0.976
 4 Other   0.00643 0.0000724 0.0000558 0.00343 0.000105  0.990
 5 Other   0       0.0000720 0.0000555 0       0.000104  1.000
 6 3       0       0.0000704 0.909     0       0.000102  0.0909
 7 3       0       0.0000761 0.995     0       0.000110  0.00491
 8 1       0.984   0.0000735 0.0000567 0.00349 0.000106  0.0127
 9 Other   0.195   0.0000602 0.0000464 0.00285 0.0000870 0.802
10 Other   0       0.0000737 0.0000568 0       0         1.000
# … with 1,115 more rows
```



Advanced Machine Learning

102 & 202A





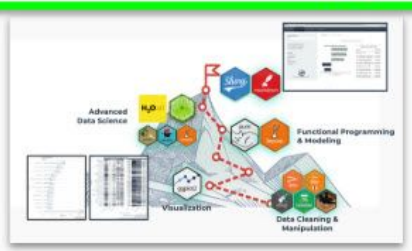
Business Science University

Data Science for Business Transformation in **6-Months**

The program that will deliver YOUR Transformation

R-TRACK

LEARNING LABS PRO



Do Business Projects
Climb the Hill

Build Production-Ready
Web Apps

Complete 1-Hour Courses
Continuous Education

Start

1

2

3

Finish

Analysis Courses
101 & 201

App Development Courses
102

Learning Labs PRO

Everything is **Taken Care of** For You in Our Platform

4-Course R-Track System

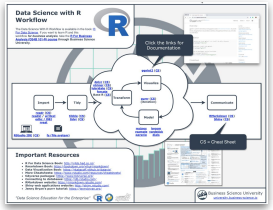
Business Analysis with R
(DS4B 101-R)

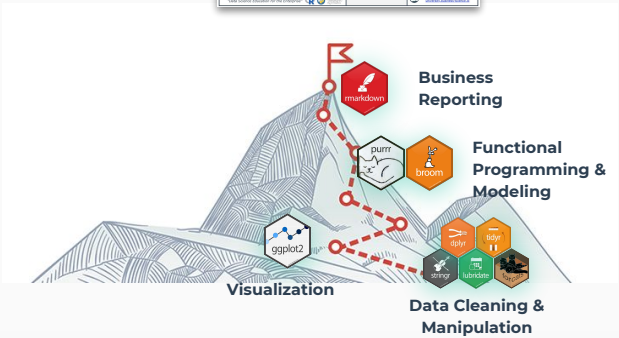
Data Science For Business with R
(DS4B 201-R)

Web Apps & Shiny Developer
(DS4B 102-R + DS4B 202A-R)

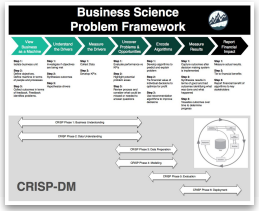
Project-Based Courses with Business Application

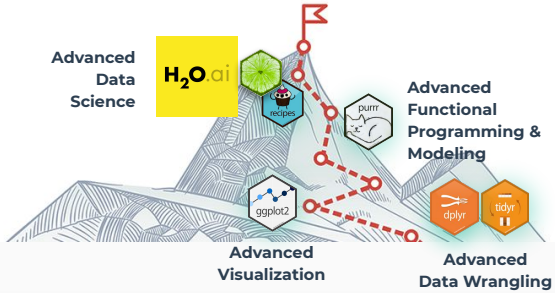
Data Science Foundations
7 Weeks



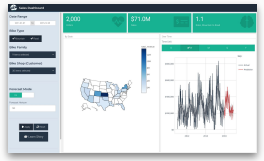


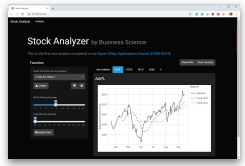
Machine Learning & Business Consulting
10 Weeks





Web Application Development
12 Weeks





Key Benefits

- Fundamentals - Weeks 1-5 (25 hours of Video Lessons)
  - Data Manipulation (dplyr)
  - Time series (lubridate)
  - Text (stringr)
  - Categorical (forcats)
  - Visualization (ggplot2)
  - Programming & Iteration (purrr)
  - 3 Challenges

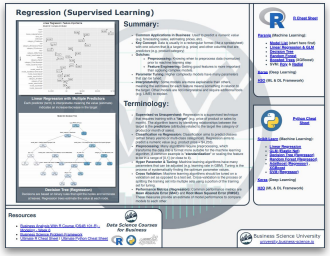
- **Machine Learning - Week 6 (8 hours of Video Lessons)**
  - **Clustering (3 hours)**
  - **Regression (5 hours)**
  - **2 Challenges**

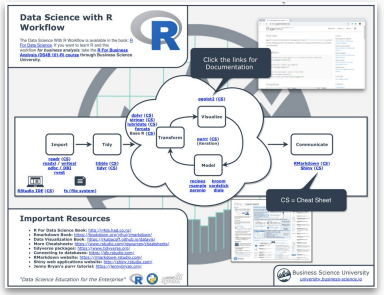
- Learn Business Reporting - Week 7
  - RMarkdown & plotly
  - 2 Project Reports:
    1. Product Pricing Algo
    2. Customer Segmentation

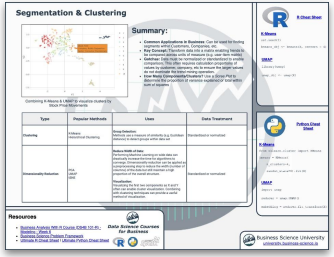
Business Analysis with R (DS4B 101-R)

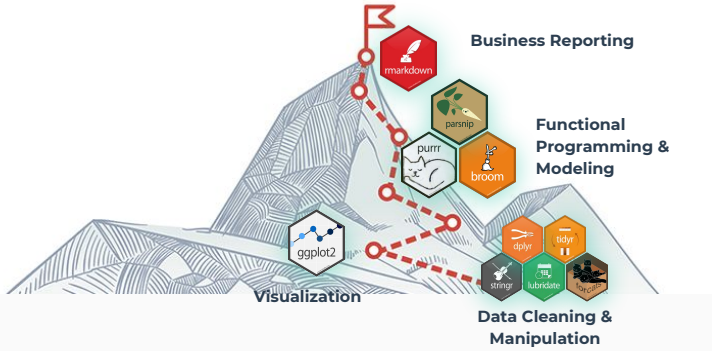
Data Science Foundations

7 Weeks









Key Benefits

End-to-End Churn Project

Understanding the Problem & Preparing Data - Weeks 1-4

- Project Setup & Framework
- Business Understanding / Sizing Problem
- Tidy Evaluation - rlang
- EDA - Exploring Data -GGally, skimr
- Data Preparation - recipes
- Correlation Analysis
- 3 Challenges

Machine Learning - Weeks 5, 6, 7

- **H2O AutoML - Modeling Churn**
- **ML Performance**
- **LIME Feature Explanation**

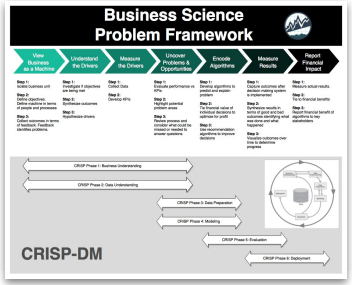
Return-On-Investment - Weeks 7, 8, 9

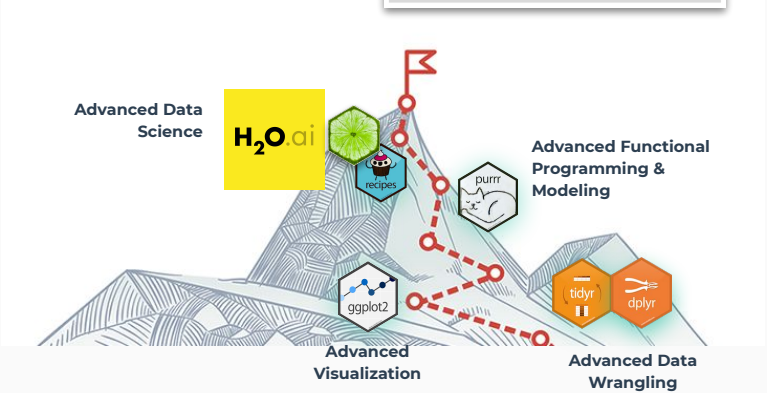
- **Expected Value Framework**
- **Threshold Optimization**
- **Sensitivity Analysis**
- **Recommendation Algorithm**

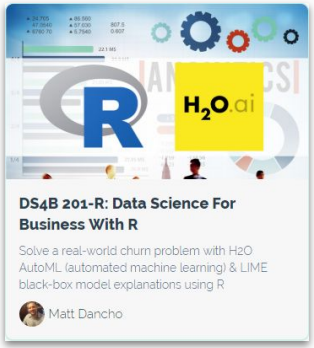
Data Science For Business (DS4B 201-R)

Machine Learning & Business Consulting

10 Weeks







Key Benefits

Learn Shiny & Flexdashboard
- Build Applications
- Learn Reactive Programming
- Integrate Machine Learning

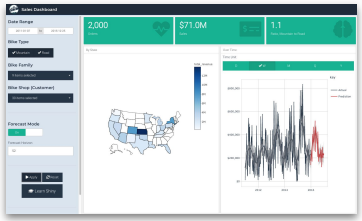
App #1: Predictive Pricing App
- Model Product Portfolio
- XGBoost Pricing Prediction
- Generate new products instantly

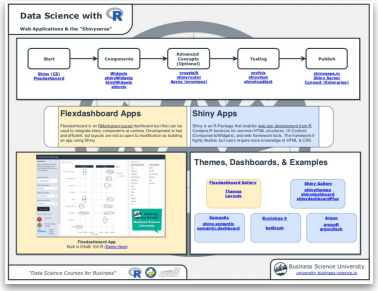
App #2: Sales Dashboard with Demand Forecasting
- Model Demand History
- Segment Forecasts by Product & Customer
- XGBoost Time Series Forecast
- Generate new forecasts instantly

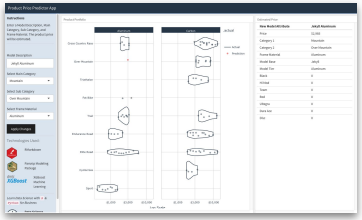
Shiny Apps for Business (DS4B 102-R)

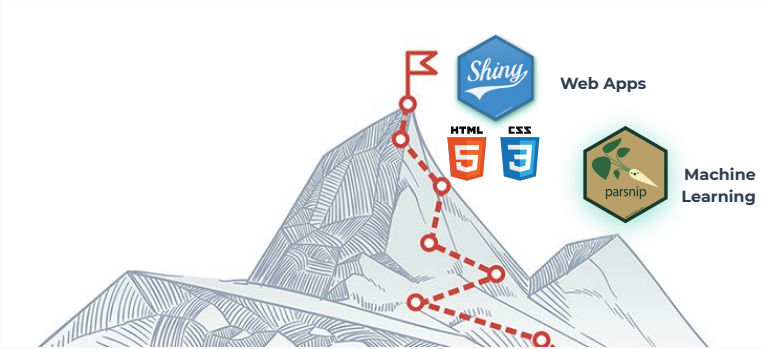
Web Application Development

4 Weeks









Key Benefits

Frontend + Backend + Production Deployment

Frontend for Shiny
- Bootstrap

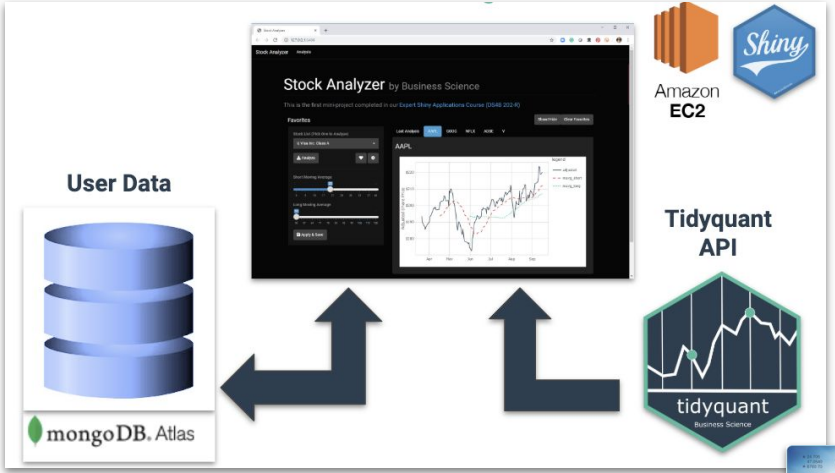
Backend for Shiny
- MongoDB
- Dynamic UI
- User Authentication
- Store & Write User Data

Production Deployment
- AWS
- EC2 Server
- VPC Connection
- URL Routing

Shiny Apps for Business (DS4B 202A-R)

Web Application Development

6 Weeks



aws

Bootstrap

mongoDB.

20% OFF PROMO Code: SHINYDEVLAUNCH

R-TRACK BUNDLE

4-Course Bundle - Machine Learning + Expert Web Applications (R-Track)

Go from Beginner to Expert Data Scientist & Shiny Developer in Under 6-Months

4 Course Bundle ~~$1,596~~

$119/mo
Expires Nov 1

DS4B 101-R: Business Analysis With R

Your Data Science Journey Starts Now! Learn the fundamentals of data science for business with the tidyverse.

Matt Dancho

DS4B 102-R: Shiny Web Applications For Business (Level 1)

Build a predictive web application using Shiny, Flexdashboard, and XGBoost

Matt Dancho

DS4B 201-R: Data Science For Business With R

Solve a real-world churn problem with H2O AutoML (automated machine learning) & LIME black-box model explanations using R

Matt Dancho

DS4B 202A-R: Expert Shiny Developer with AWS

Learn how to build Scalable Data Science Applications using R, Shiny, and AWS Cloud Technology.

Matt Dancho

○	**Paid Course** 20% COUPON DISCOUNT	~~$1,596~~ **$1,276.80**
◉	**12 Low Monthly Payments** 20% COUPON DISCOUNT 12X Payment Plan	~~12 payments of $149/m~~ **12 payments of $119.20/m**

Begin Learning Today

university.business-science.io