

# Data-driven Sales Leads Prediction

*for Everything-as-a-Service in the Cloud*

Data Scientists @ IBM Cloud | **Chul Sung, Bo Zhang, Chunhui Higgins**

{sungc, bozhang, ychunhui}@us.ibm.com

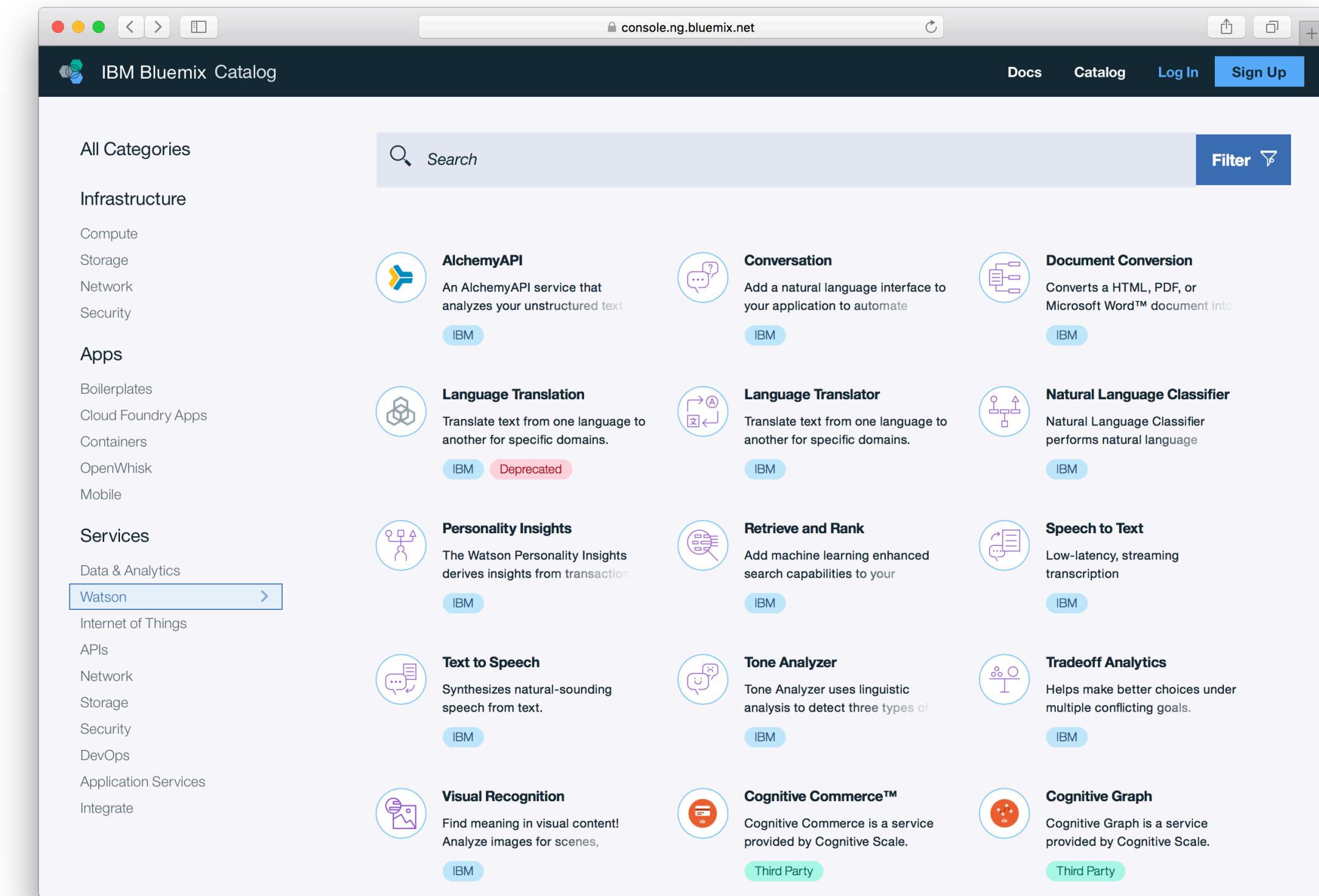
Texas A&M University | **Yoonsuck Choe**

[choe@cs.tamu.edu](mailto:choe@cs.tamu.edu)

October 19, 2016 at IEEE DSAA 2016

# IBM Bluemix

- A **hybrid cloud development platform** to access a catalog of services and APIs, on which customers can build and run their apps
- It is a **compelling** platform:
  - **Widest selection** of compute choices for scalable, enterprise production apps such as bare metal, virtual servers, cloud foundry, and containers
  - **Big data and analytics for insights** beyond all needs
  - **Watson** for instant cognitive capabilities into apps and services
  - A myriad of visiting customers



# Cognitive Era with IBM Bluemix

- **IBM Bluemix** helps to run scalable analytics solutions like Streaming Analytics or AlchemyData News to get results in seconds.
- You can improve decisions and outcomes with Retrieve and Rank.



Streaming  
Analytics



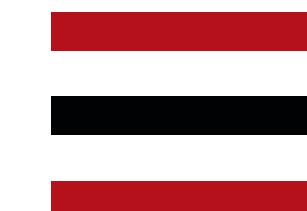
Retrieve and  
Rank



AlchemyData



Geospatial  
Analytics



# Challenge

**IBM Bluemix** is **growing rapidly** in popularity, but ...

- A key challenge is how to **engage with a huge number of customers** while optimizing sales and marketing performance.
- It is nearly impossible to understand behaviors of the customers without **an inter-team collaboration** among various stakeholders.

On top of these challenges ...

# DDBA Challenge

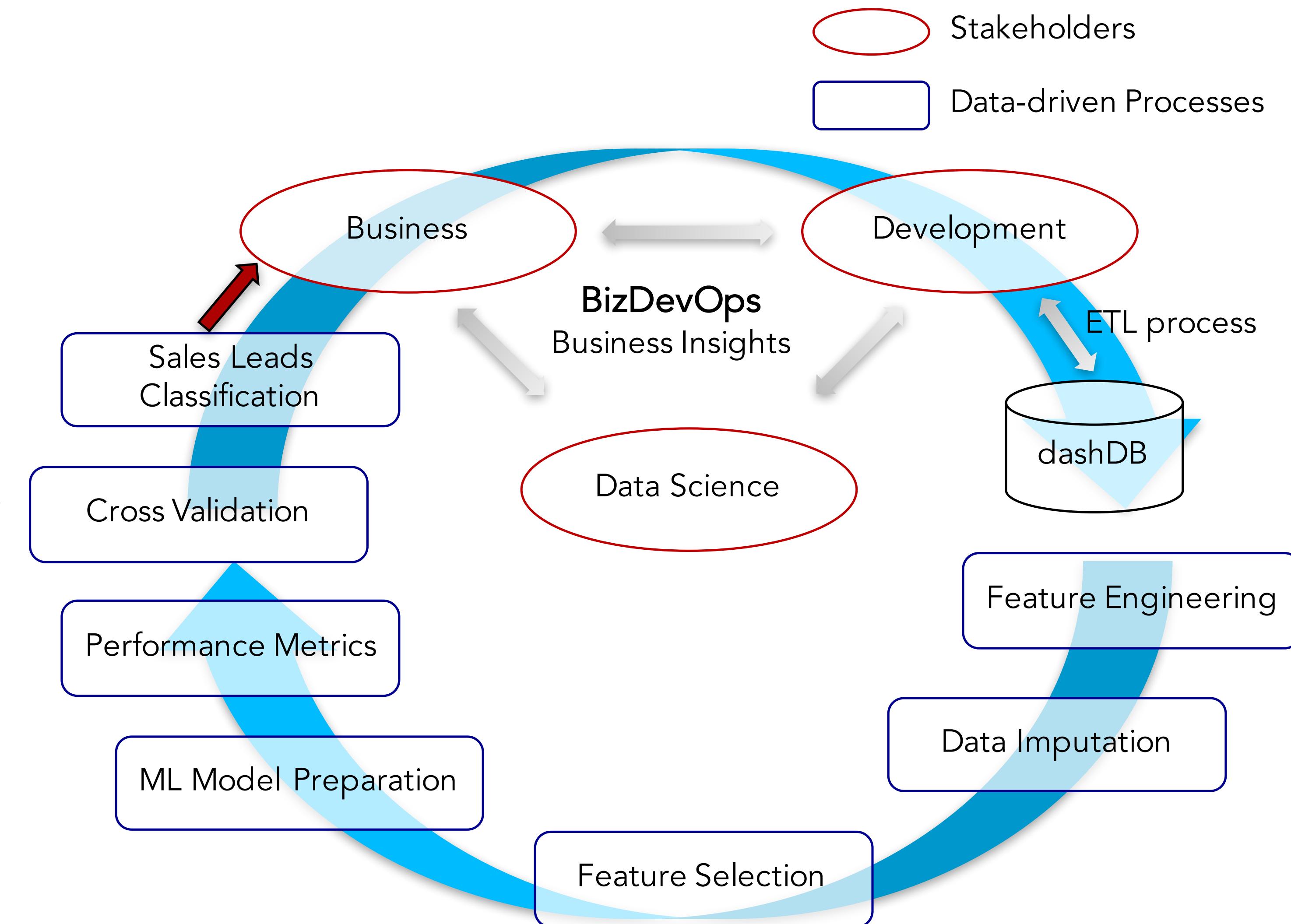
Additional challenges to **recognize high profile customers** in Data-driven Business Analysis (DDBA):

- **heavy customer traffic flows**
- conversion to **meaningful and consumable** features
- handling **missing values** of some customer behaviors
- **class imbalance problem** – the actual paying customer class represented by only a few tuples
- constantly **changing environment** – business models (providing services) and user behaviors

# Iterative Prediction Framework

**BizDevOps** – fills in the gap between the actual business goals and DevOps deliverables.

**Iterative** – to adapt to a continuously changing environment.

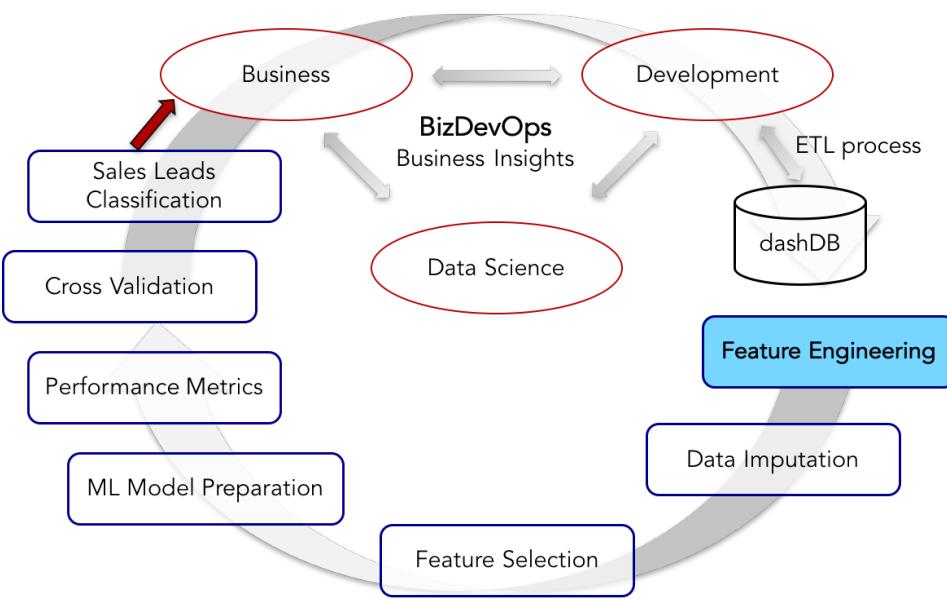
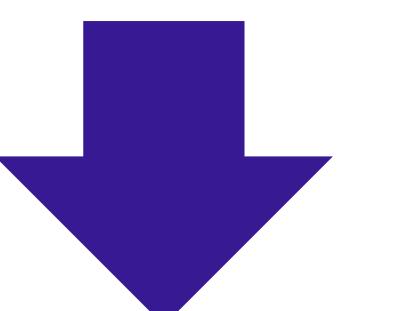


# Feature Engineering (1)

- application creation
- application update
- service creation
- service binding creation
- service binding deletion
- service usage
- runtime usage

## RFD Analysis

*R*ecency + *F*requency + *D*uration

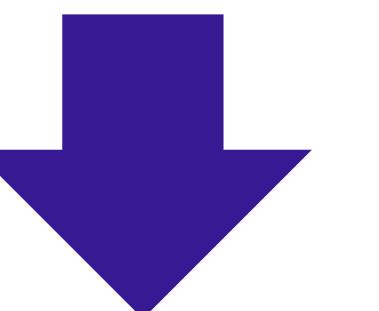


# Feature Engineering (2)

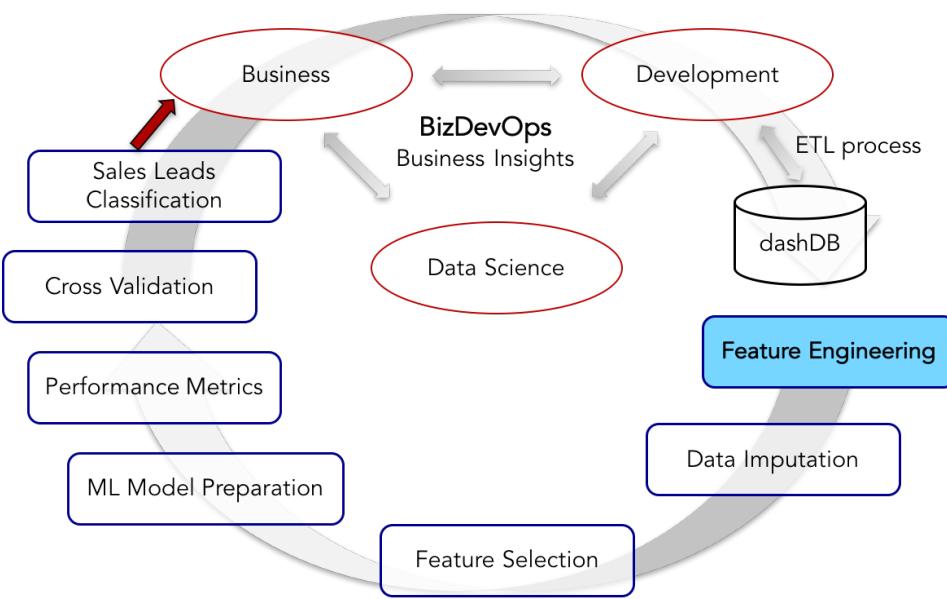
- customer account
  - applications
  - services

## L Analysis

*Lifetime*

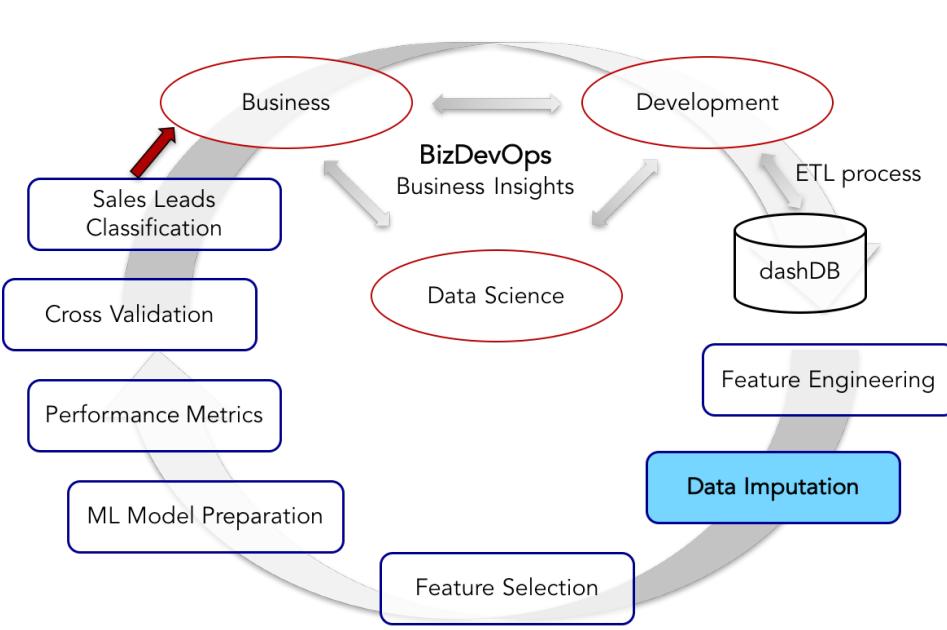


total **24** features



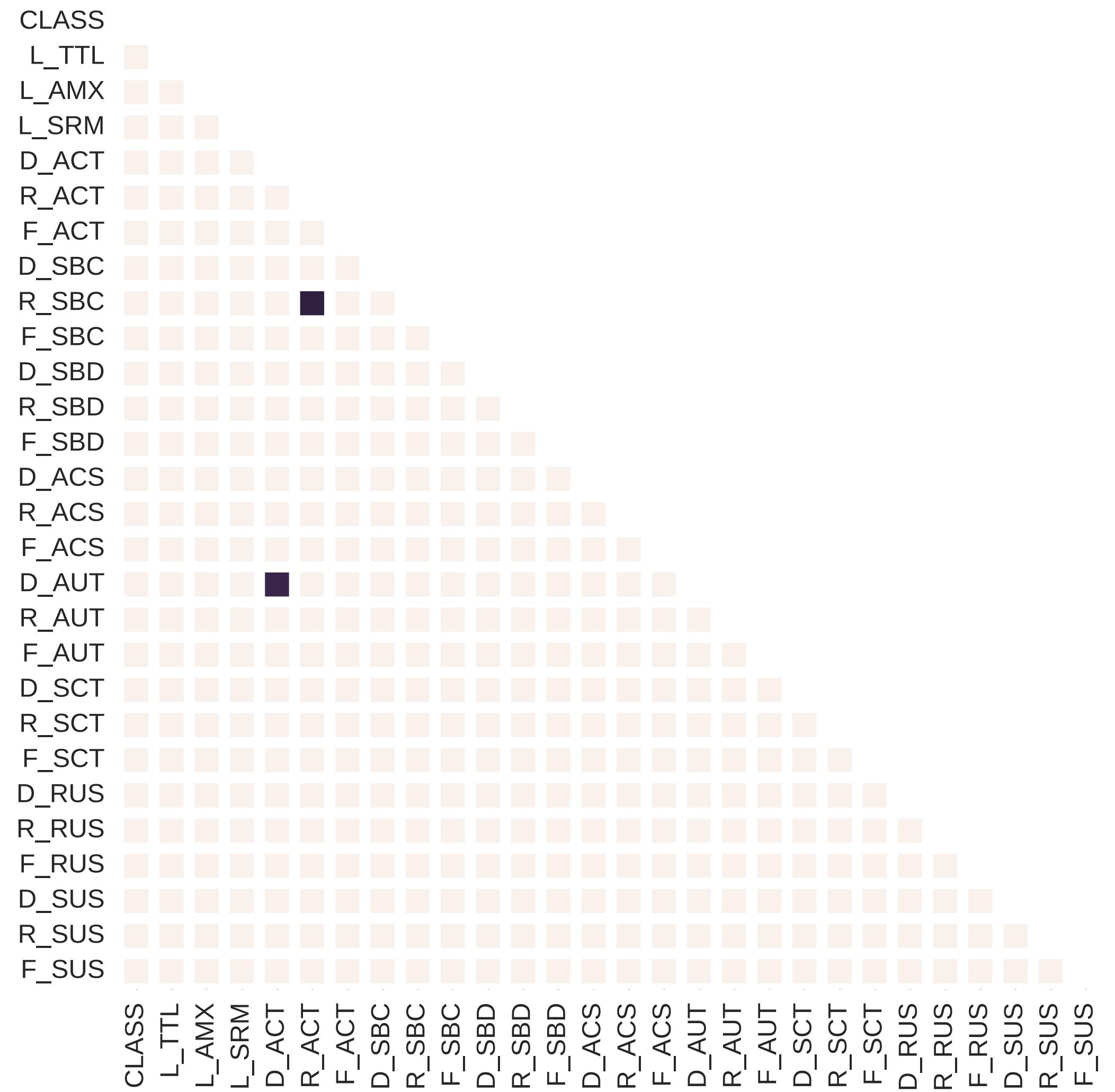
# Data Completion

- Ignoring instances having null values
- null values to the mean of close group
- **null values to max or min of values** – in the way which least affects the data
  - less Recency/Duration values more informative, so **max** of values for the attributes
  - bigger Frequency/Lifetime values more informative, so **min** of values for the attributes

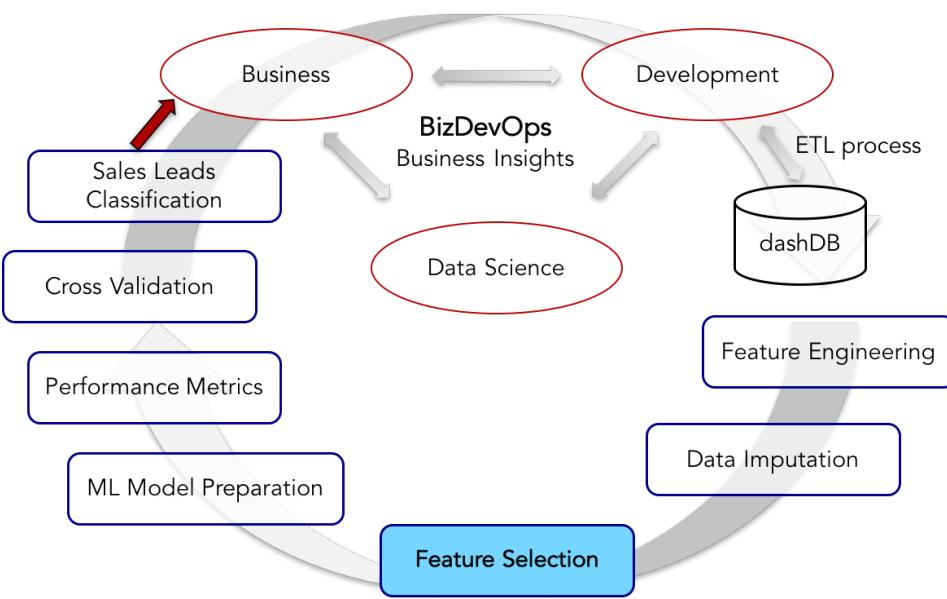


# Feature Selection - C\_95

Using **Pearson's correlation**, eliminate highly correlated features:  
(R\_SBC, R\_ACT\*) and  
(D\_ACT, D\_AUT\*)  
**over THR. |0.95|**



\* We dropped the red ones.



# Feature Selection - C\_90

Using **Pearson's correlation**,  
eliminate highly correlated  
features:

(R\_SBC, R\_SBD, R\_ACT\*),

(D\_ACT, D\_AUT\*),

(F\_SBC, F\_ACT\*),

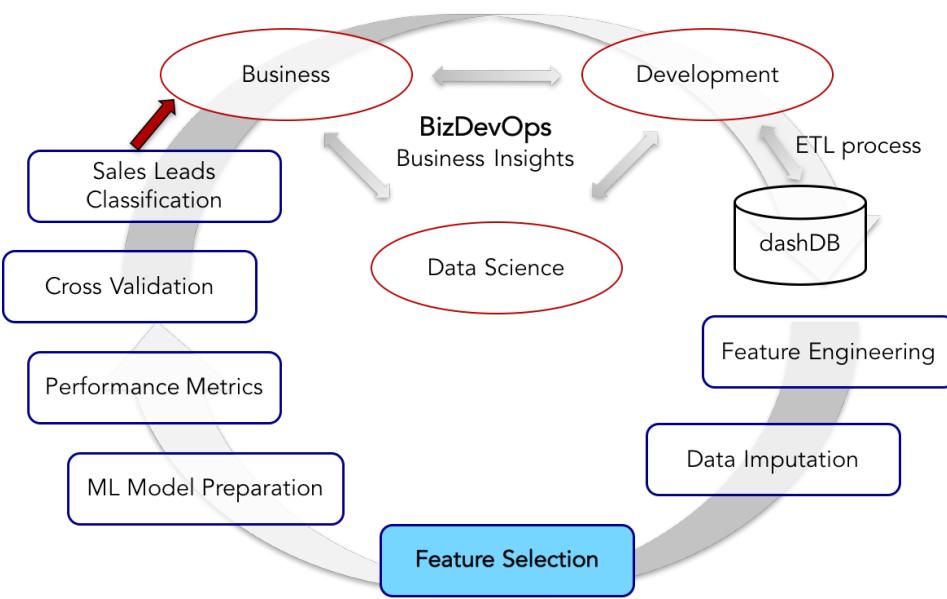
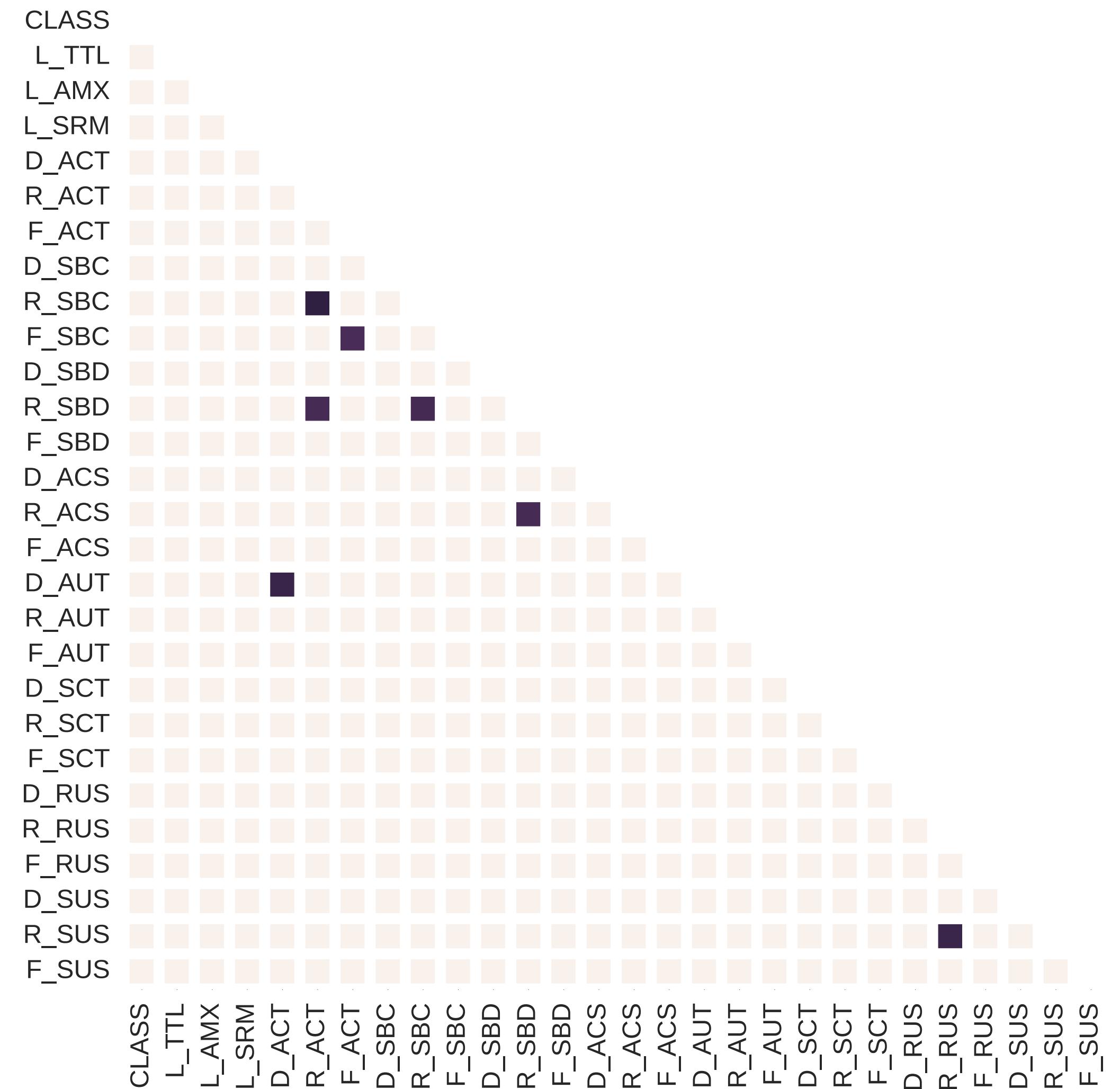
(R\_ACS\*, R\_SBD),

(D\_AUT, D\_ACT\*), and

(R\_SUS\*, R\_RUS)

**over THR. |0.90|**

\* We dropped the red ones.



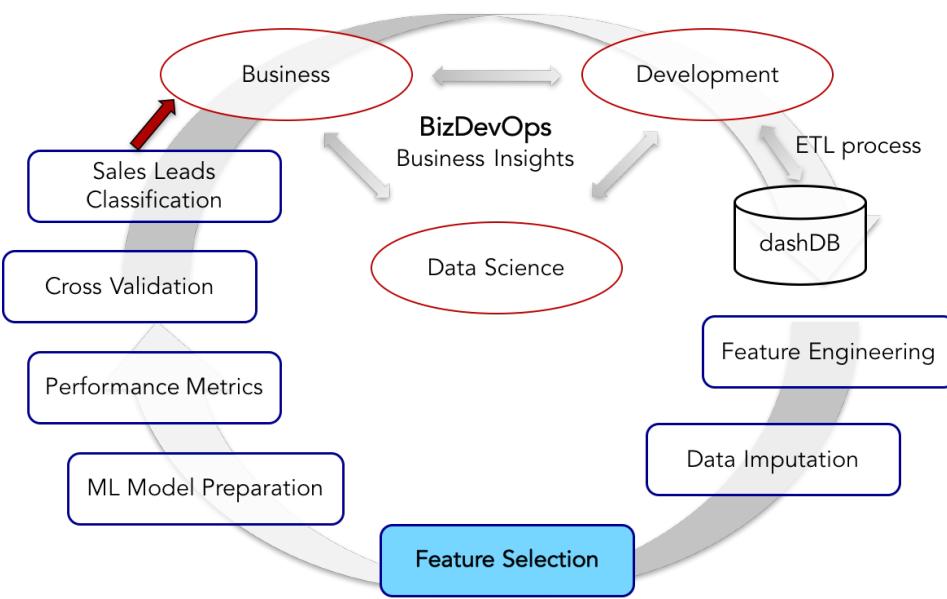
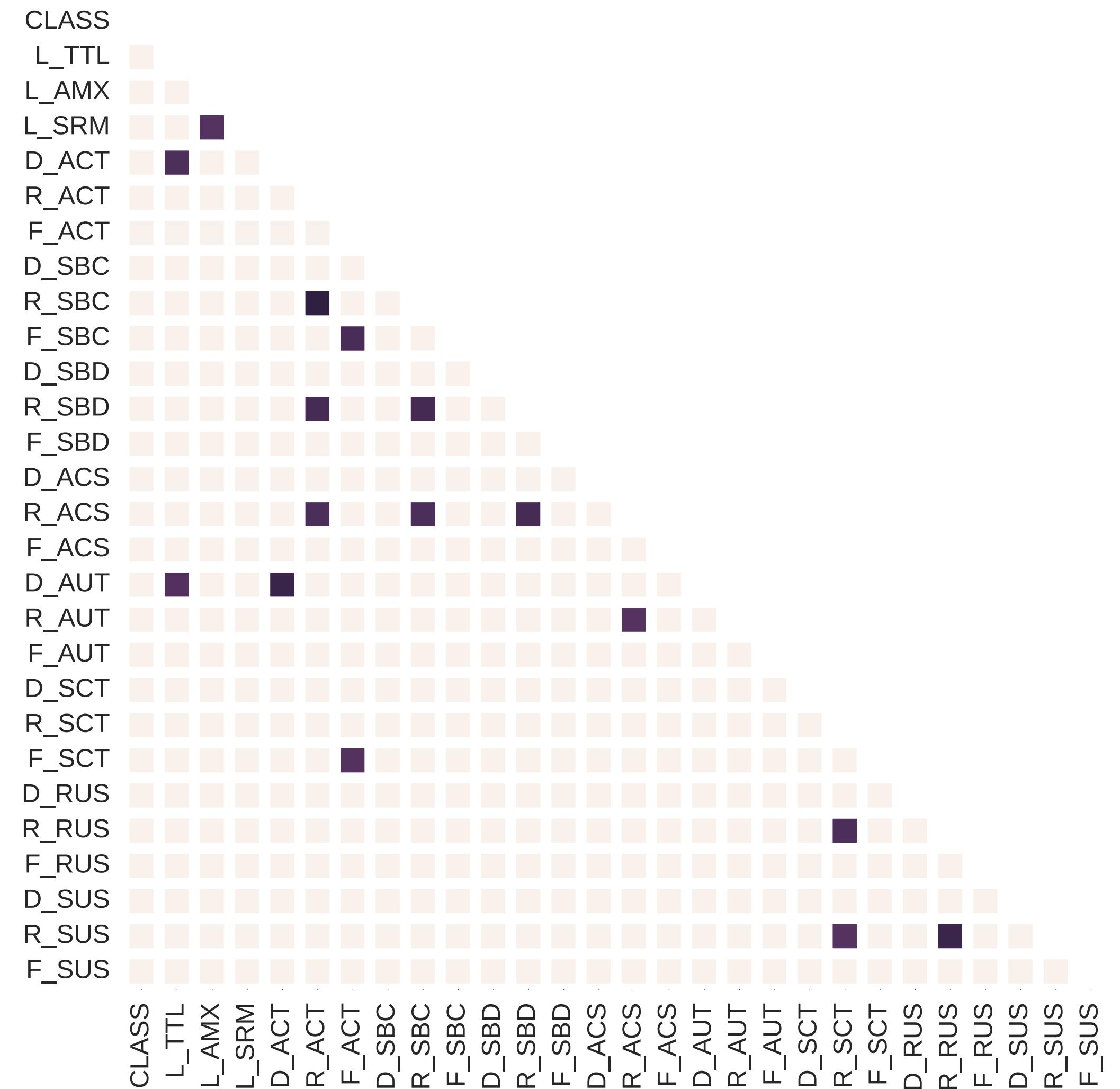
# Feature Selection - C\_85

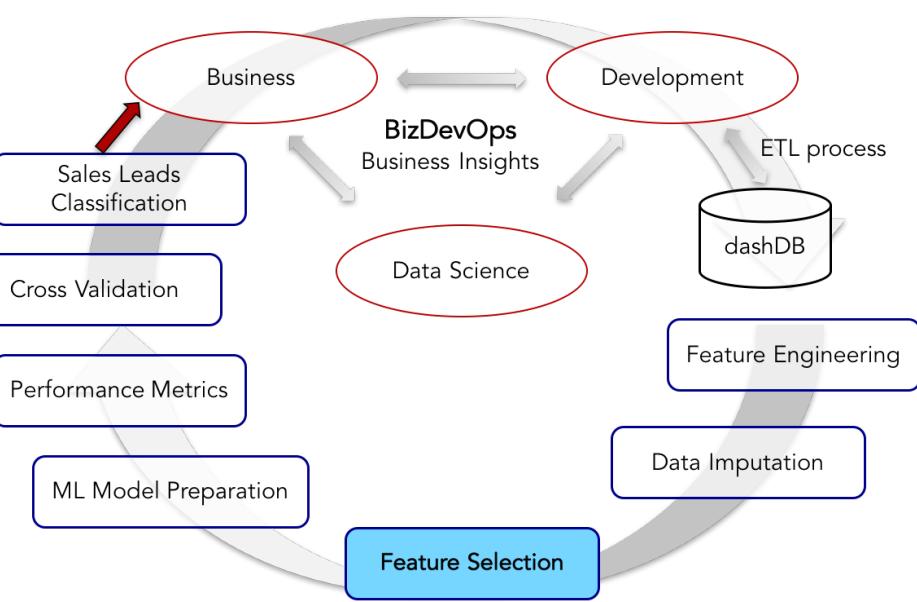
Using **Pearson's correlation**, eliminate highly correlated features:

(R\_SBC, R\_SBD, R\_ACT\*), (D\_ACT, D\_AUT\*),  
(F\_SBC, F\_ACT\*), (R\_ACS\*, R\_SBD),  
(D\_AUT, D\_ACT\*), (R\_SUS\*, R\_RUS),  
(L\_SRM, L\_AMX\*), (D\_ACT\*, L\_TTL),  
(D\_AUT, L\_TTL), (R\_AUT\*, R\_ACS),  
(F\_SCT\*, F\_ACT), (R\_ACS\*, R\_ACT),  
(R\_ACS\*, R\_SBC), and  
(R\_RUS\*, R\_SUS, R\_SCT)

**over THR. |0.85|**

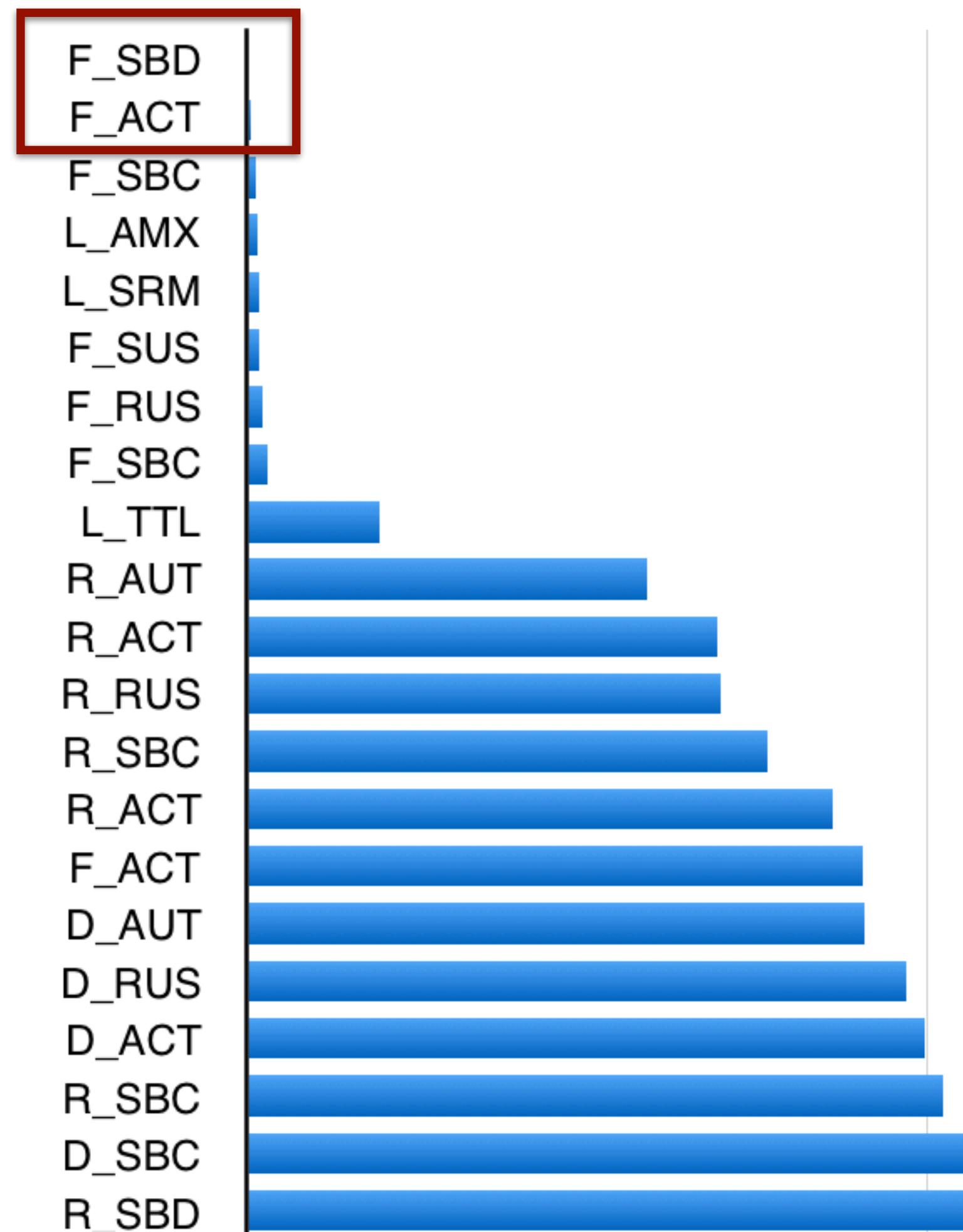
\* We dropped the red ones.



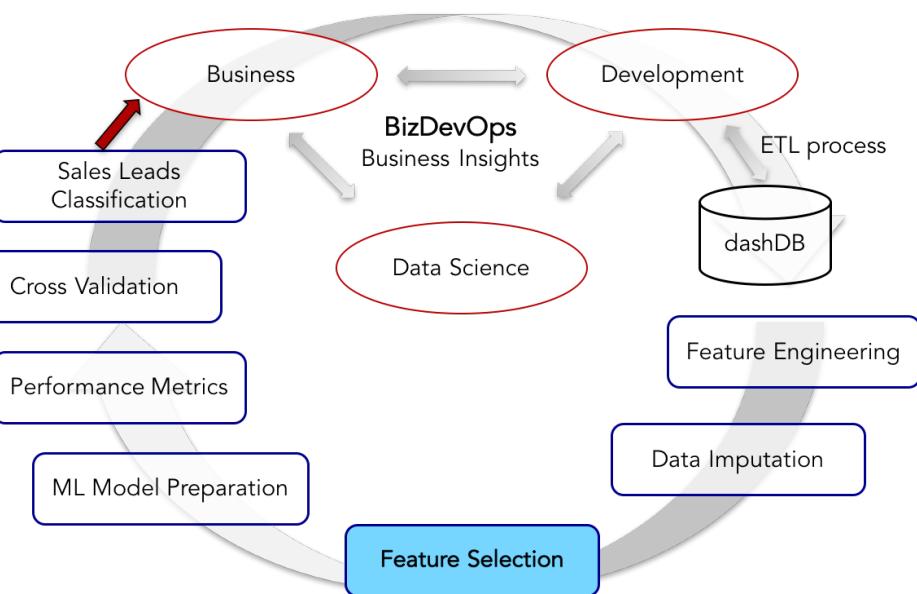


# Feature Selection - L\_V1

Eliminate very **low-variance** features:  
**F\_ACT\***, **F\_SBD\***  
 (less than 2000)

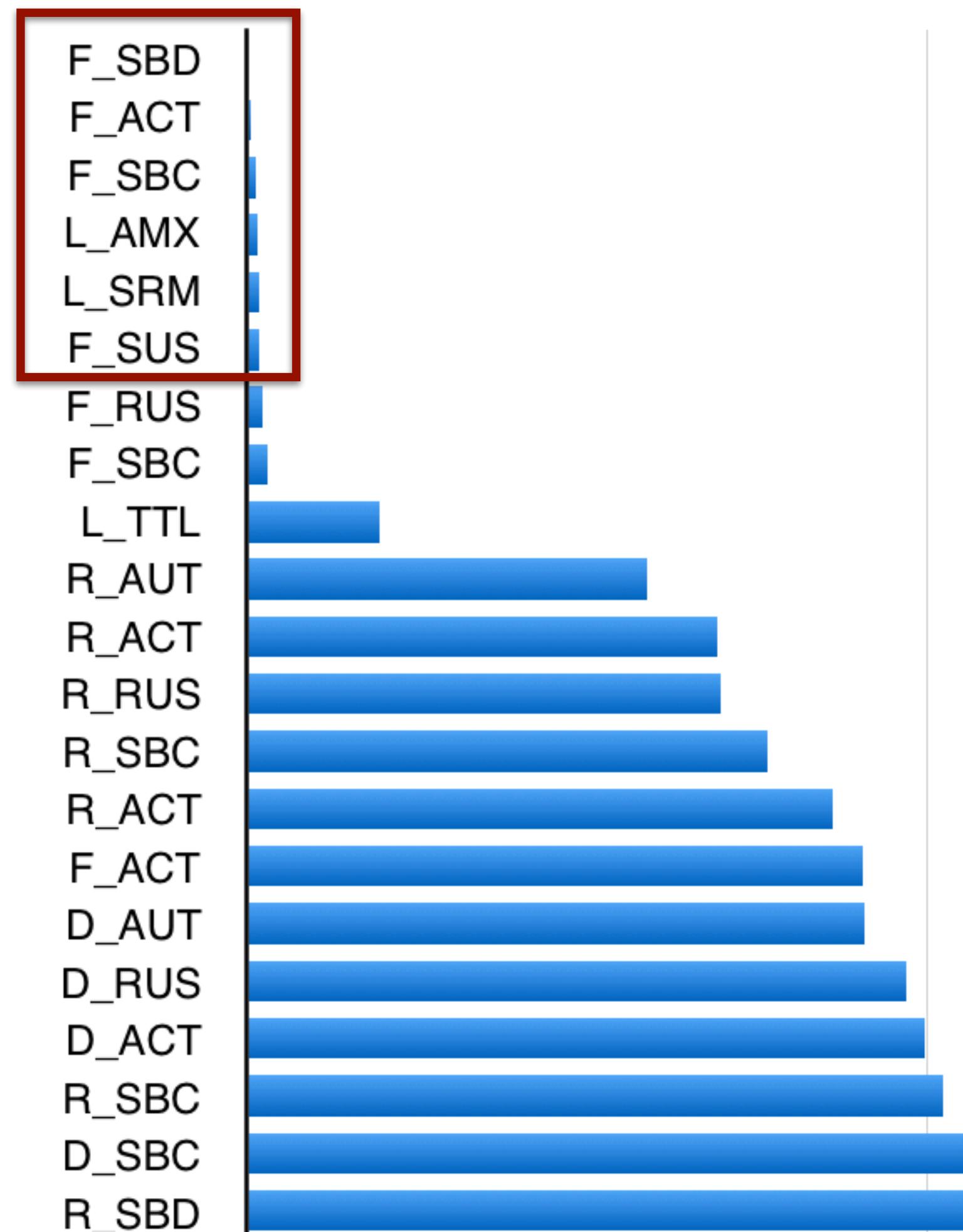


\* We dropped the red ones.



# Feature Selection - L\_V2

Eliminate very **low-variance** features:  
**F\_ACT\***, **F\_SBD\***,  
**F\_SUS\***, **L\_SRMR\***,  
**L\_AMX\***, and **F\_SBC**  
(less than 3500)

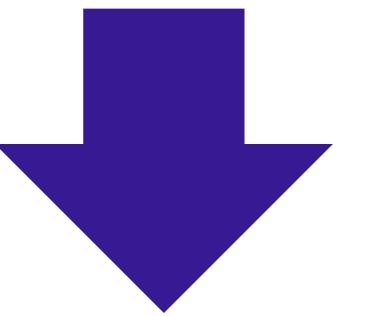


\* We dropped the red ones.

# Model Preparation

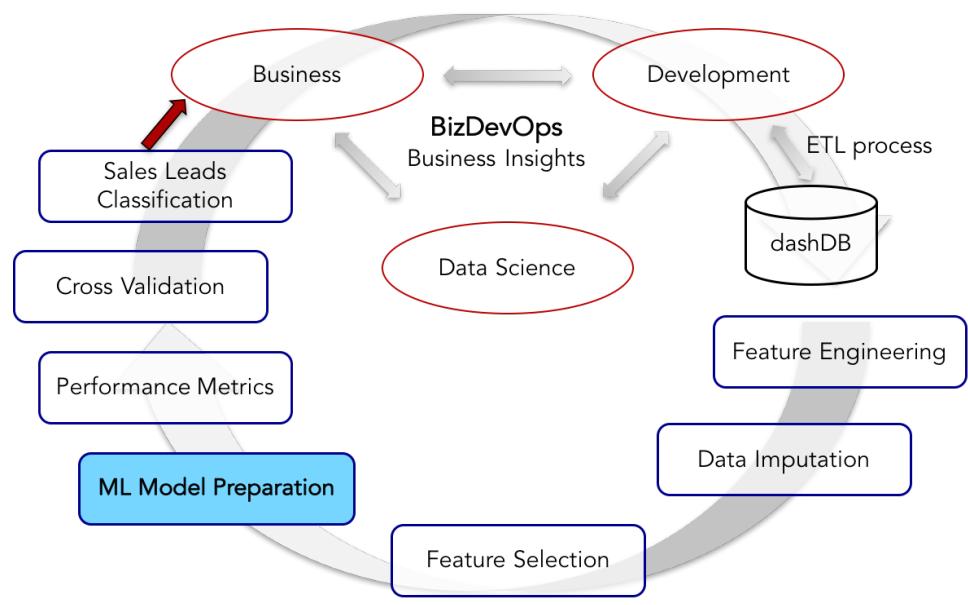
five groups of features:

ALL, C\_95, C\_90, C\_85, L\_V1, L\_V2



- logistic regression (**LR**)
- k-nearest neighbor (**KNN**)
- naive Bayes (**NB**)
- decision tree (**DT**)
- random forest (**RF**)
- adaptive boosting (**AB**)
- gradient boosting (**GB**)

- weighted RF (**WRF**)
- weighted AB (**WAB**)
- weighted GB (**WGB**)



# Performance metrics

On our **class-imbalance situation** where a few customers vs. the majority of non-paying customers, we have to consider:

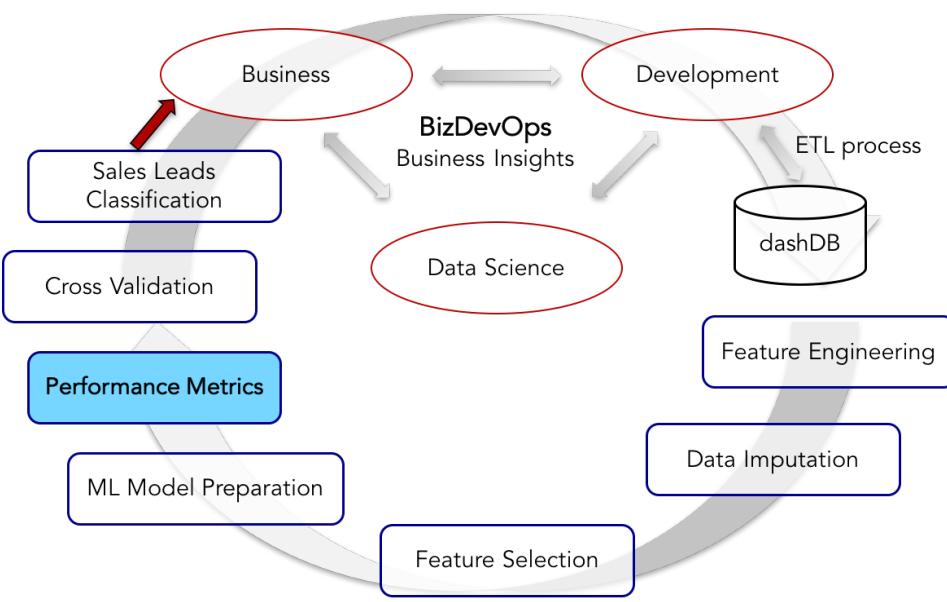
1. accuracy of positive classification:
2. accuracy of true-positive value:

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

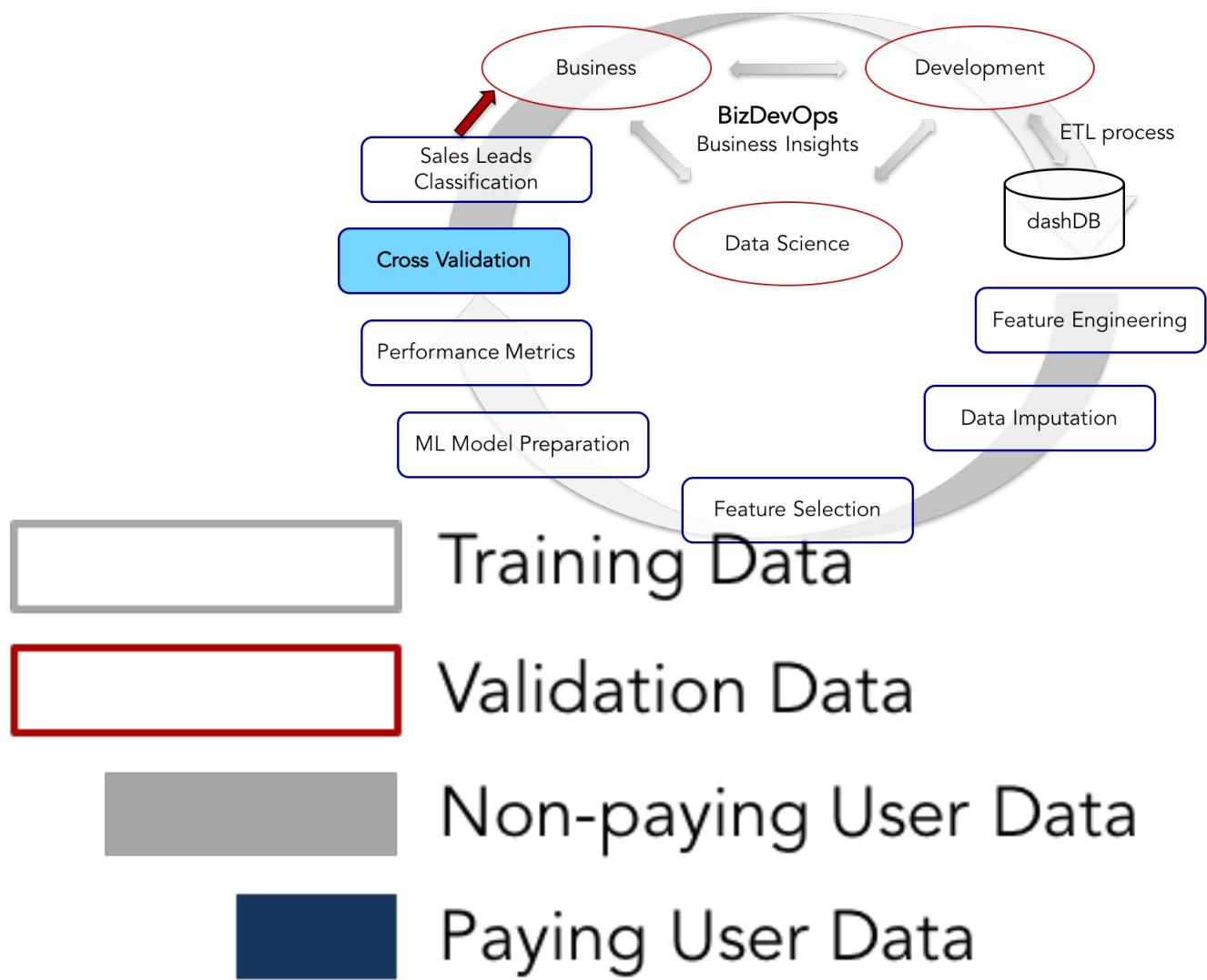
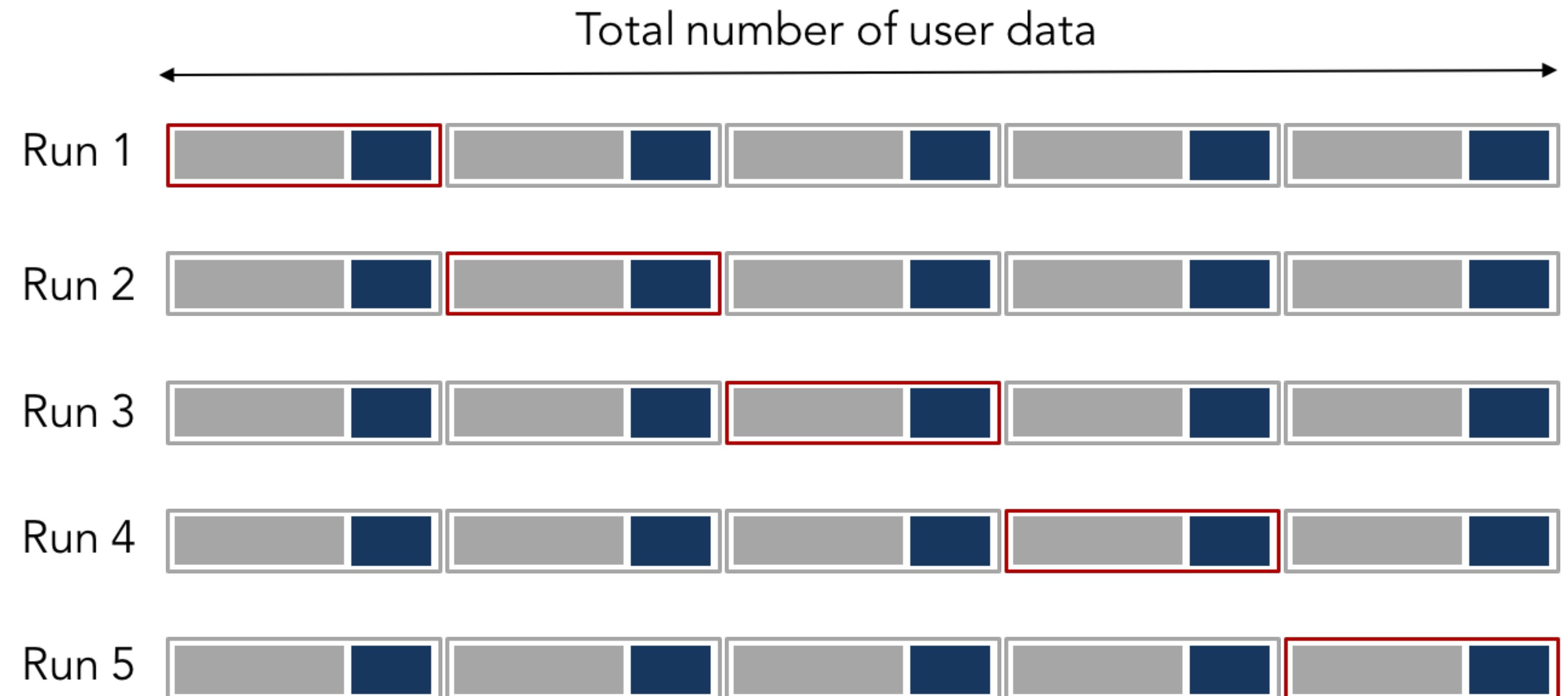
harmonic average

(Positive) **F1 Score**



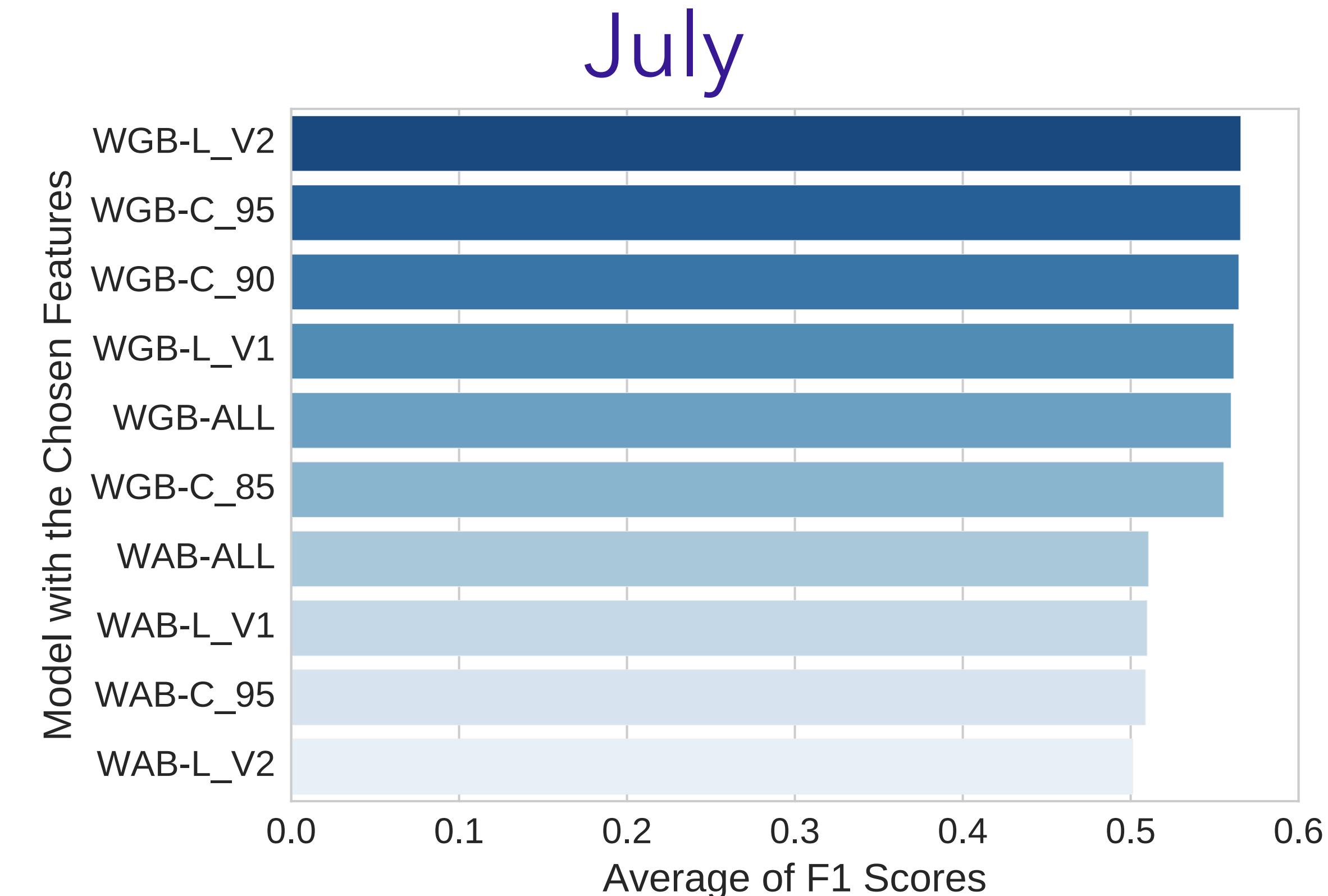
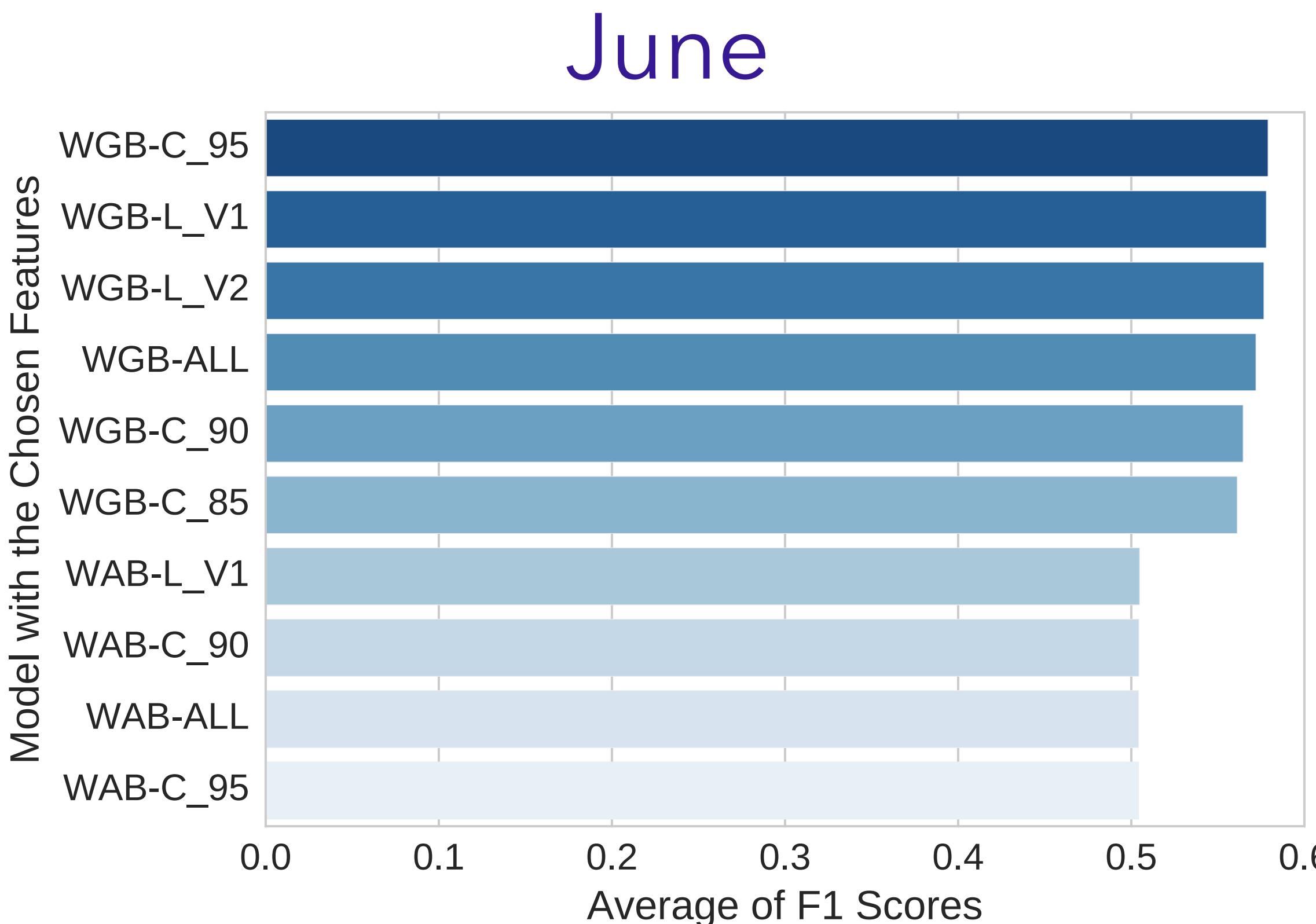
# Cross Validation

**Stratified** 5-fold cross validation to deal with our **class-imbalance situation.**



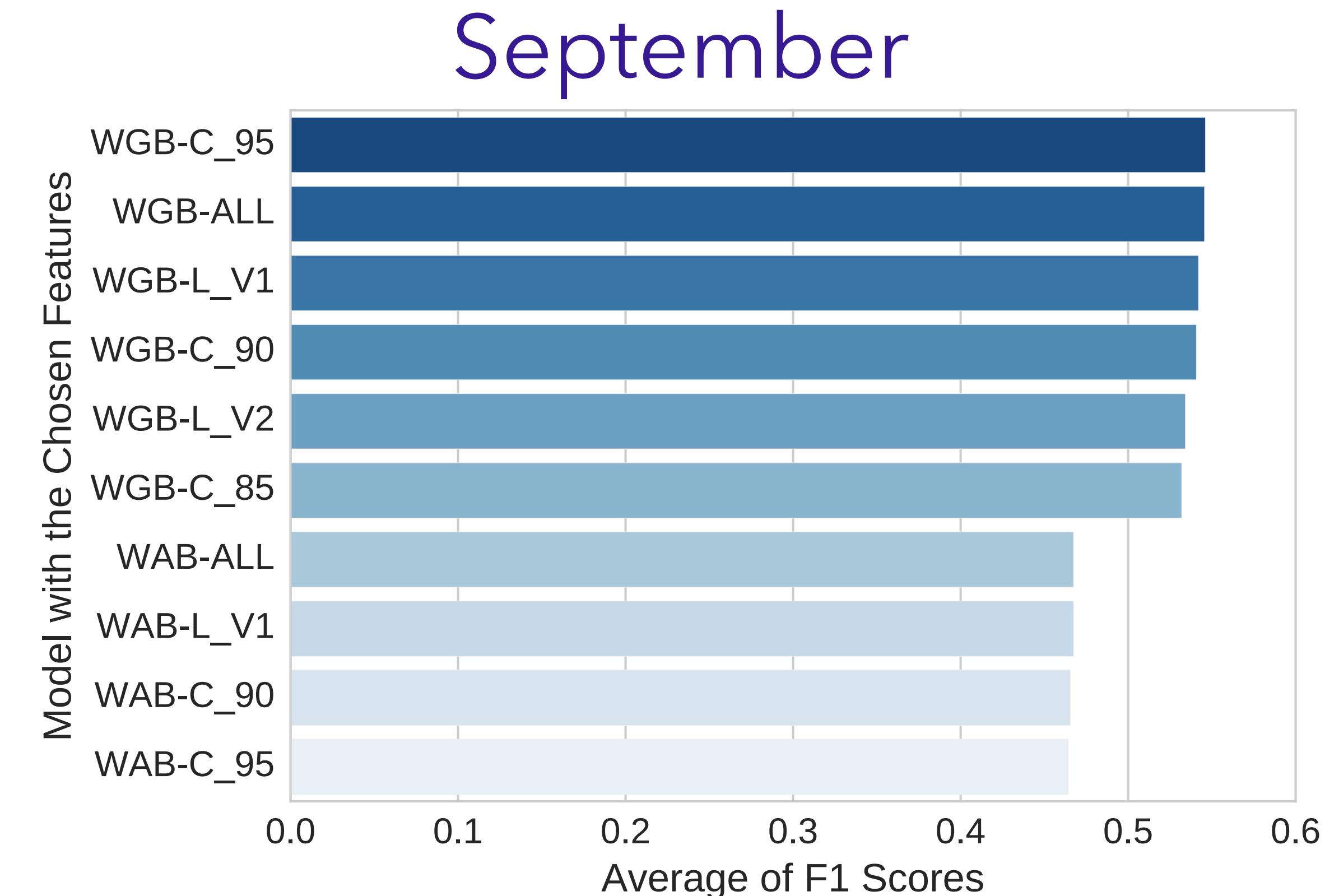
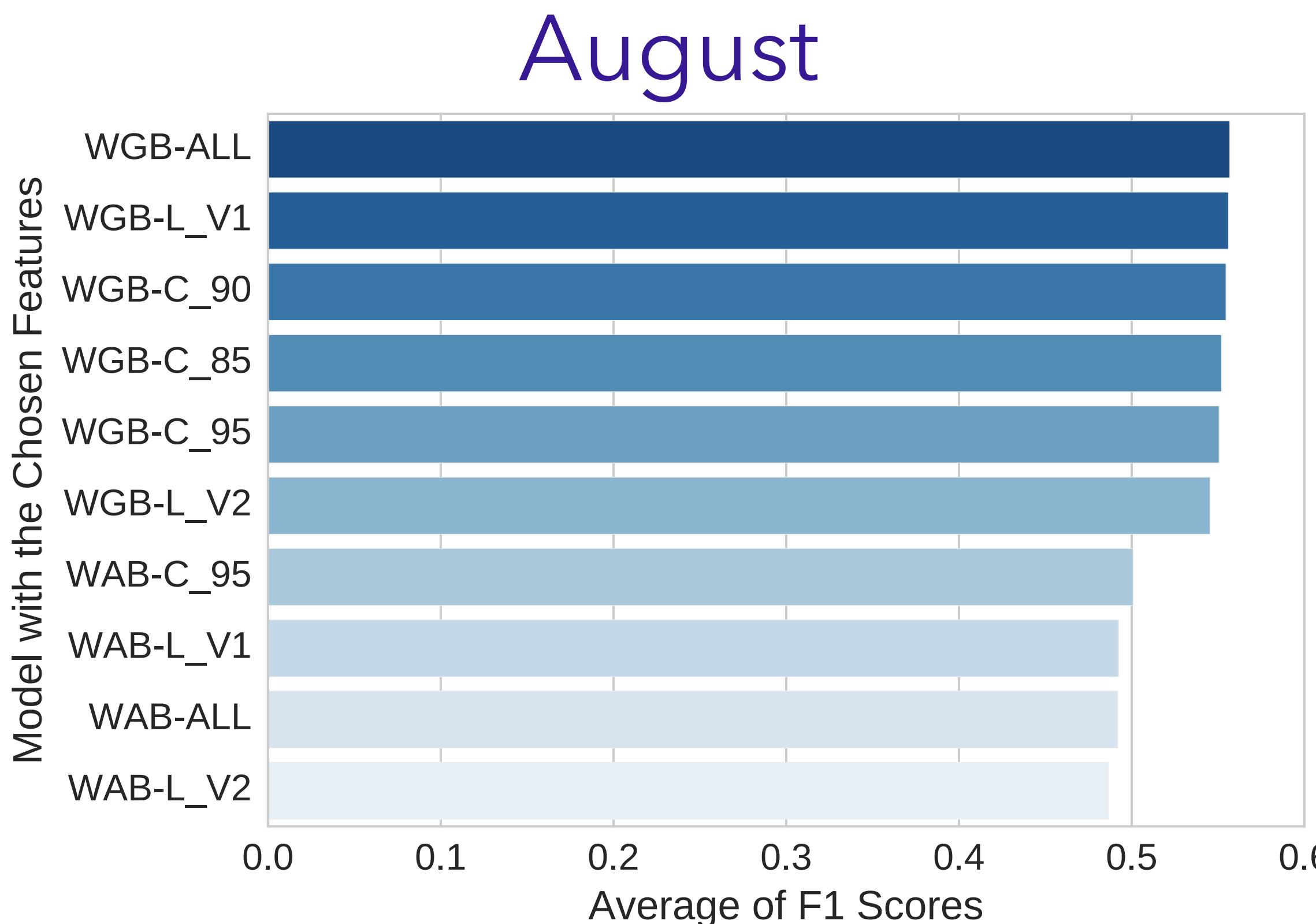
# Performance Comparison (1)

**Weighted Gradient Boosting (WGB)** outperformed others.

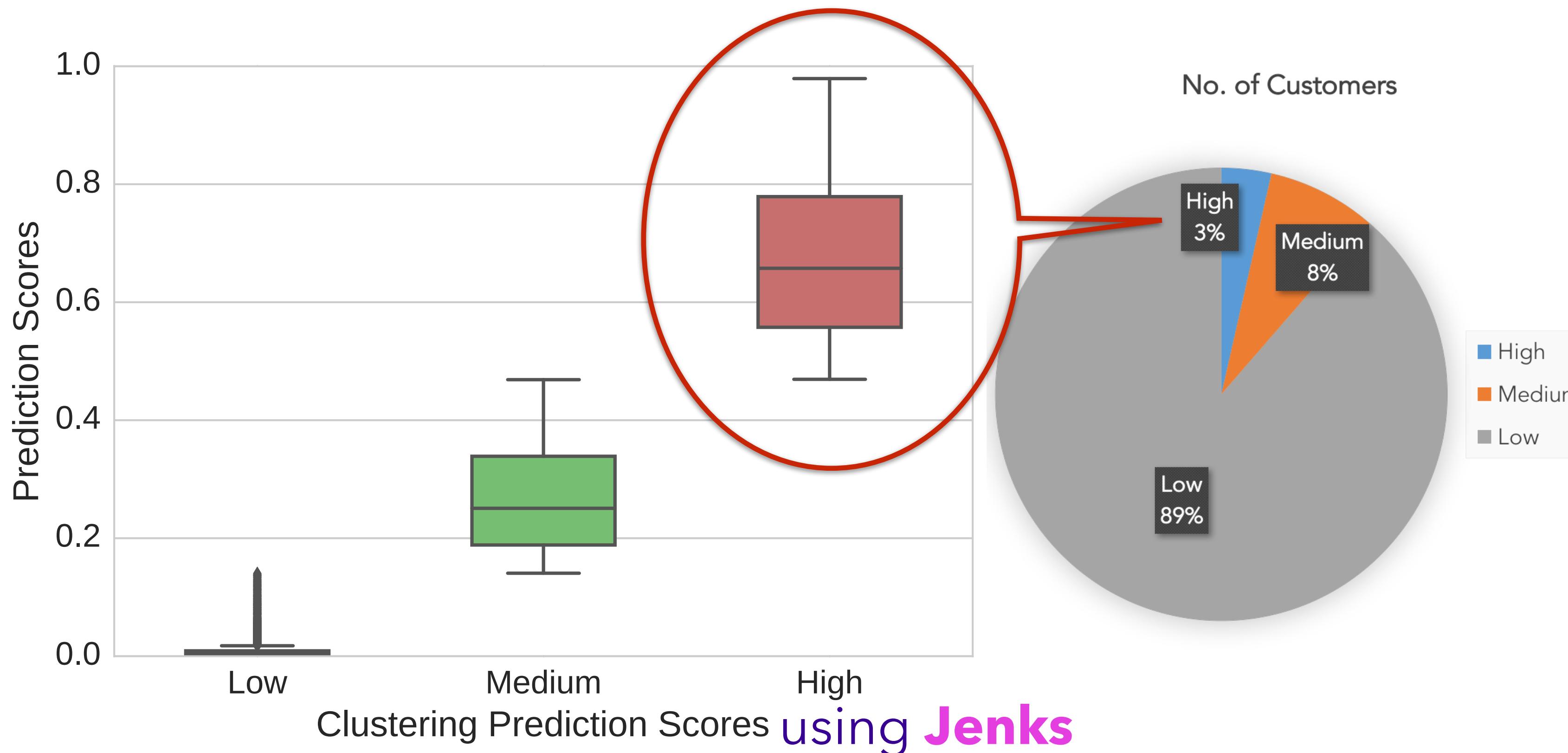


# Performance Comparison (2)

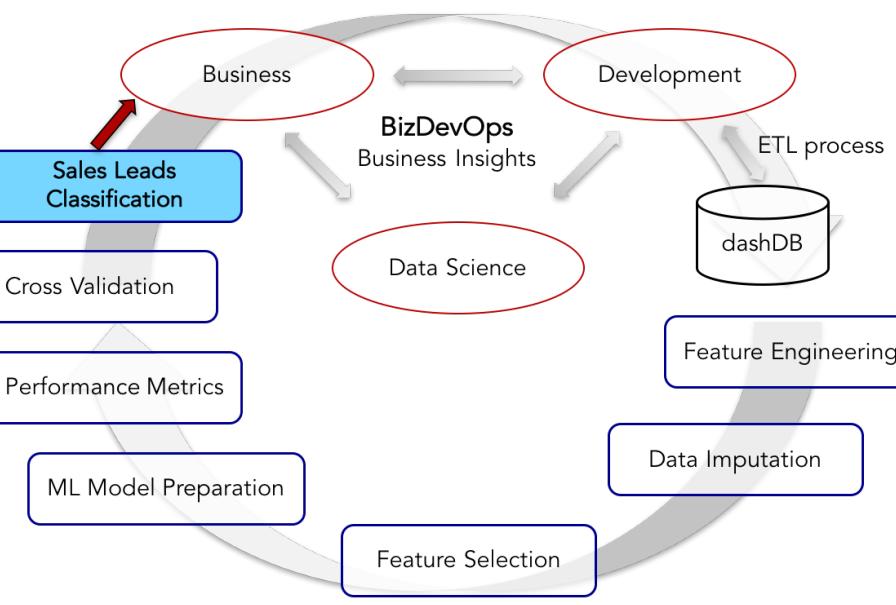
**Weighted Gradient Boosting (WGB)** outperformed others.



# Sales Leads Clustering

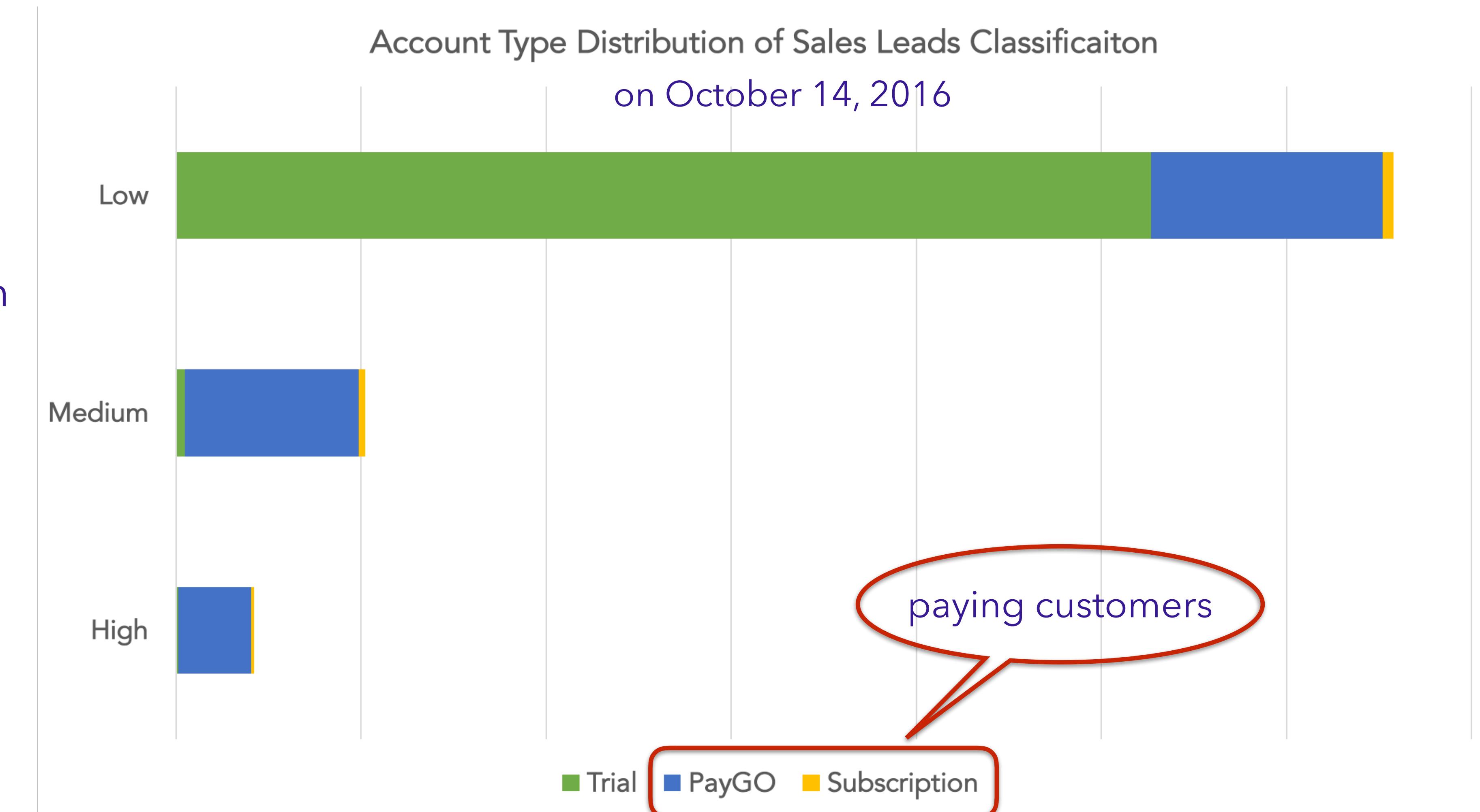


- Once we have the prediction scores from our trained model, we cluster three groups using **Jenks** algorithm.
- We deliver the **High-class** group (even 3% overall) to the Business team.



# Evaluation of Sales Leads Clustering

- **Account types** in October from Sales Leads three classes in September
- In the High & Medium classes, **a few trial accounts** and the majority of paying customers
- Still necessary to handle the paying customers in the low class



# Discussion

- Proposed **RFDL** (Recency, Frequency, Duration, and Lifetime) analysis to measure customer behaviors.
- **Iterative** prediction framework widely applicable to an online or cloud business domain with a constant **flux of customer traffic**
- Effective **prioritization** of new customers, targeting the high profile customers
- Necessary to handle the actual high profile customers in the low class

# Questions

Thank you!

<https://github.com/csung7/ieee-dsaa2016>

Chul Sung, Bo Zhang, Chunhui Y. Higgins, and Yoonsuck Choe. Data-driven Sales Leads Prediction for Everything-as-a-Service in the Cloud,  
The 3rd IEEE International Conference on Data Science and Advanced Analytics (DSAA), 2016.

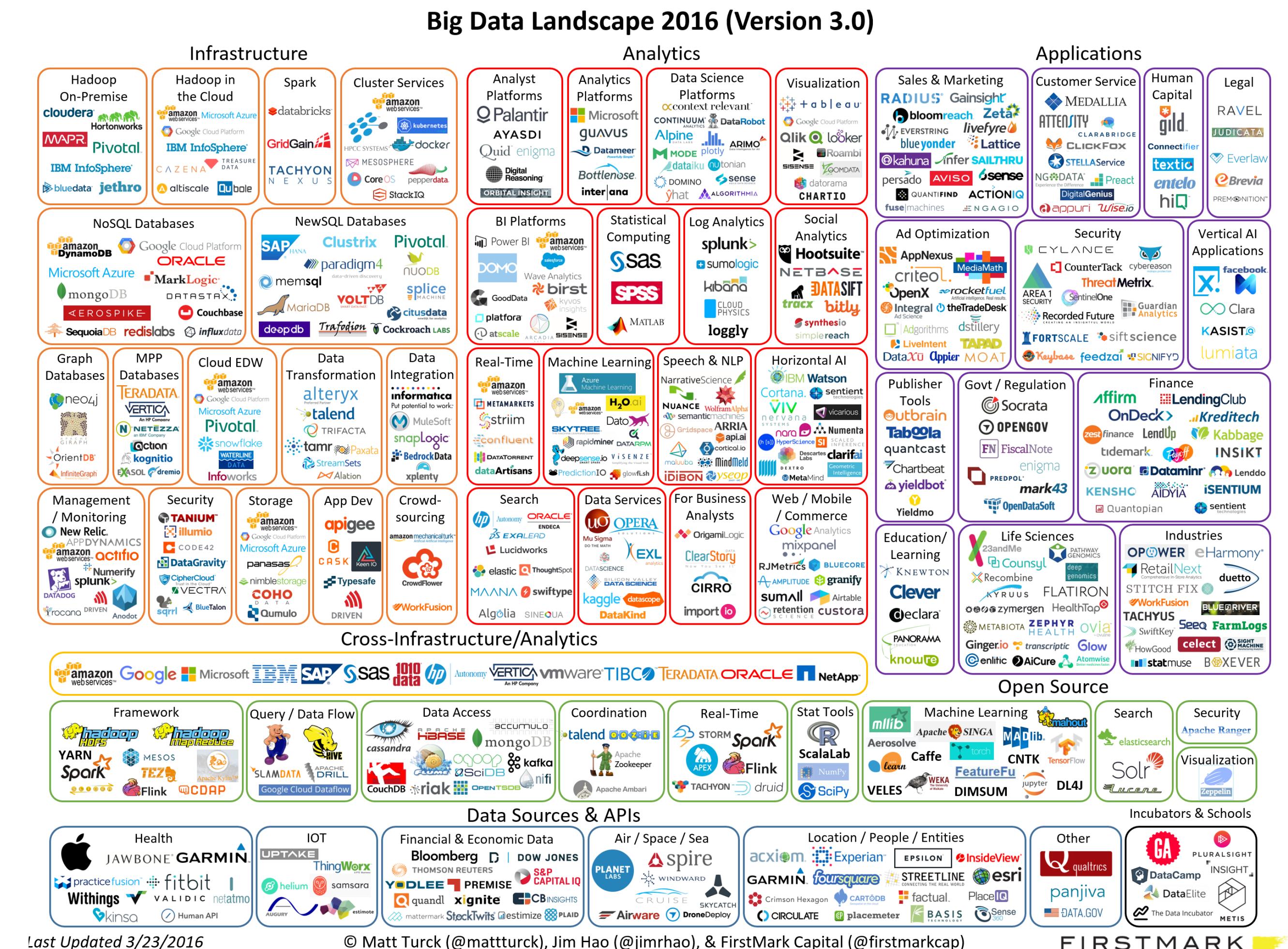
# Acknowledgements

Thank **Damion Heredia** (VP of IBM Cloud Product Management),  
**Janine Sneed** (VP of IBM Cloud GTM),  
**Adam M Gunther** (Director of Bluemix Offering),  
**Dave Lindquist** (VP and IBM Fellow of IBM Cloud Development),  
**Donald Cronin** (Program Director of IBM Cloud Development), and  
**Bill Higgins** (Distinguish Engineer of IBM CIO DevOps).

In addition, we would like to express our special thanks for the great collaboration to **Alexander Sobran**, **Ticard Bowser**, **Pete Marshall**, and **Pete Parente**.

# Big Data Landscape

- Big Data Landscape in 2016
  - For interest of **Big Data** at fever pitch, big data companies standing out
  - A variety of applications on **across cloud platforms** within the enterprise

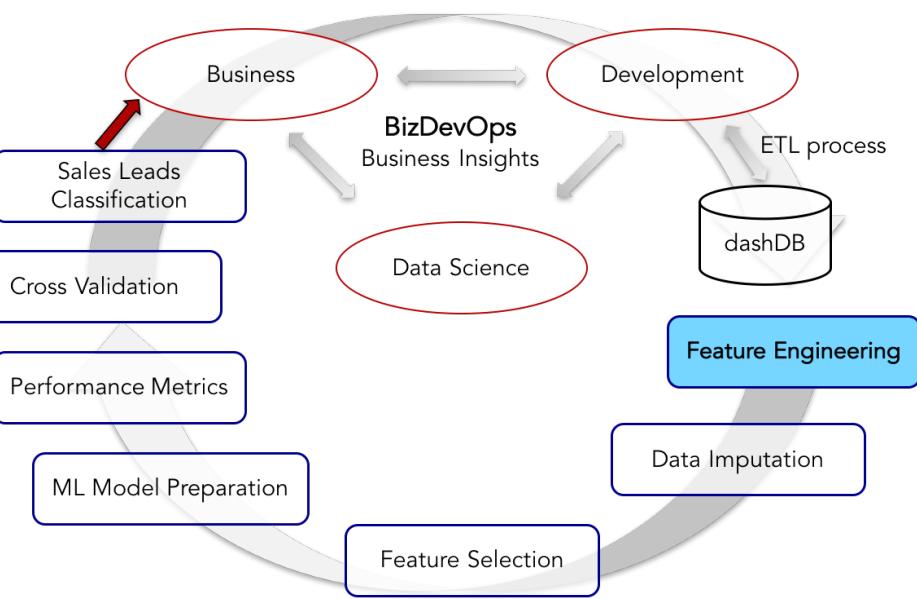


Last Updated 3/23/2016

© Matt Turck (@mattturck), Jim Hao (@jimrhao), & FirstMark Capital (@firstmarkcap)

FIRSTMARK

source: <http://mattturck.com/2016/02/01/big-data-landscape/>



# Feature Code Description

RFDL	Features (Abbr.)	Descriptions
R/F/D	{R/F/D}_ACT	Recency, Frequency, or Duration of application creation.
	{R/F/D}_ACS	Recency, Frequency, or Duration of application crash.
	{R/F/D}_AUT	Recency, Frequency, or Duration of application update.
	{R/F/D}_SCT	Recency, Frequency, or Duration of service creation.
	{R/F/D}_SBC	Recency, Frequency, or Duration of service binding creation.
	{R/F/D}_SBD	Recency, Frequency, or Duration of service binding deletion.
	{R/F/D}_SUS	Recency, Frequency, or Duration of service usage.
	{R/F/D}_RUS	Recency, Frequency, or Duration of runtime usage.
L	L_TTL	Lifetime of customer accounts.
	L_AMX	Lifetime of applications.
	L_SRM	Lifetime of services.
	CLASS	Labels for paying and non-paying customers.