

Quick Intro

Hiram Fleitas

Senior Customer Engineer Microsoft Data & Al

aka.ms/hiram

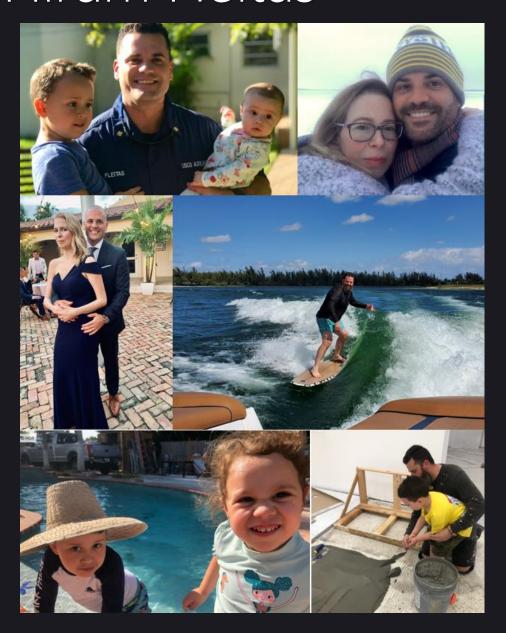


About Me

aka.ms/hiram

- 1. Machine Learning
- 2. Synapse Analytics
- 3. Power Platform
- 4. DevOps

Hiram Fleitas



Who I Am:

Cuban and live in Miami since 93.

Wrote my first app at 13 and work with Computers since.

Recently, I've been laying tile at the home that I'm building for my family.

What I Do:

New Senior CE at Microsoft.

Worked on the SQL Big Data Cluster and ML Extension as an Early Adopter.

~25 years in Tech.

Analyze Me:

Father of 2, Ocean (5) and Skylar (3).

Resilient

Perseverance

Problem solver

Jar Opener

To Work With Me, You Should Know:

- I speak English and Spanish.
- You can reach me at any time.
- I'm here to help.

What I Like:

Construction
Breakfast with my kids
Watersports, wakesurf, wakeskate
Cuban coffee, hello!

The Last Word:

"Dad, you want to make something awesome with me. It shall be so crazy! IOL"

"Believe in people."



2000s































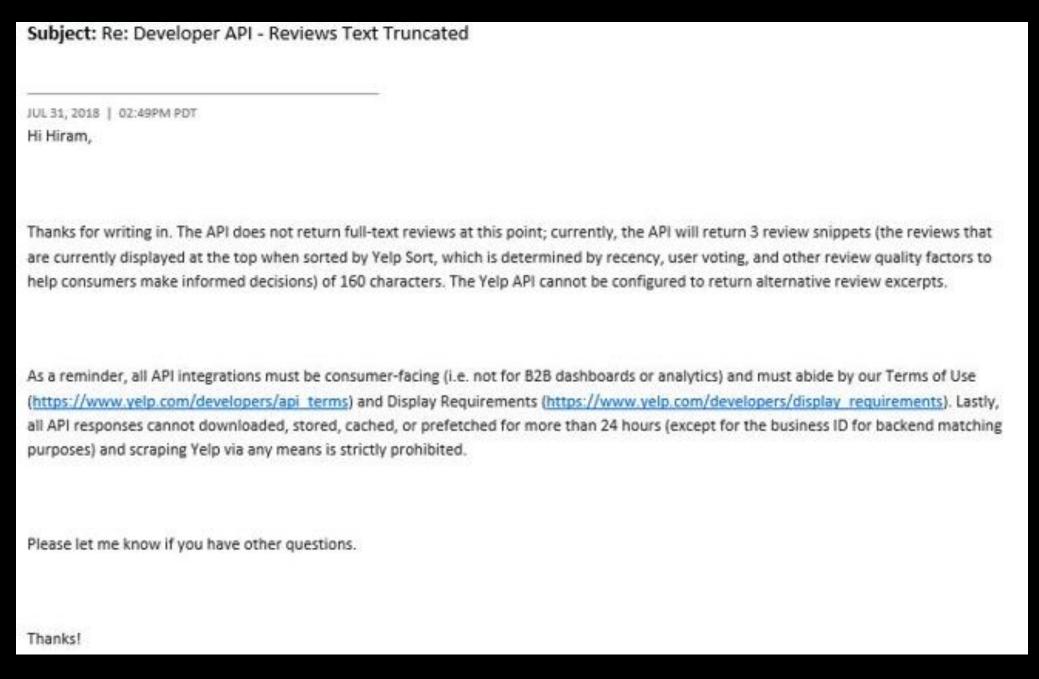








- Brand Awareness
- Fraud Detection Customer Churn
- Ethics
- Sentiment
- QnA Chat Bots
- **✓** Forms
- Enterprise Capital Modeling Financial Risk
- Length of Stay
- Object Detection



Learn more:

https://www.yelp.com/developers/api_terms https://www.yelp.com/developers/display_requirements

Support \vee

Partners 🗸

Extract information from your text

Marketplace 🗸

Use the demo below to experiment with the Text Analytics API. Pick one of our examples or provide your own. Identify the language, sentiment, key phrases, and entities (Preview) of your text by clicking "Analyze".

See it in action

Destiny is a gift. Some go their entire lives, living existence as a quiet desperation. Never learning the truth that what feels as though a burden pushing down upon our shoulders, is actually, a sense of purpose that lifts us to greater heights. Never forget that fear is but the precursor to valor, that to strive and triumph in the face of fear, is what it means to be a hero. Don't think, Master Jim. Become!

Documentation

Pricing

Training

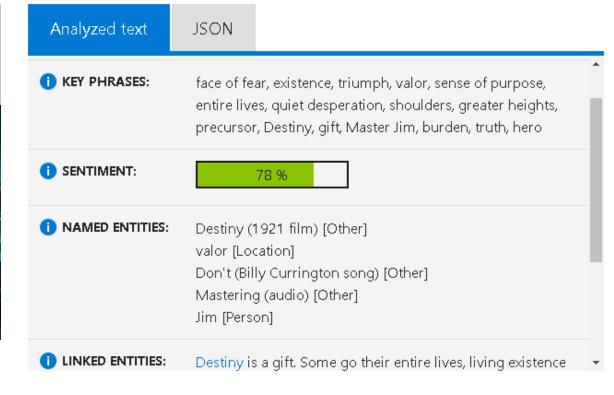
<u>Products</u> ∨

Overview

Analyze

Solutions





More ∨

Blog

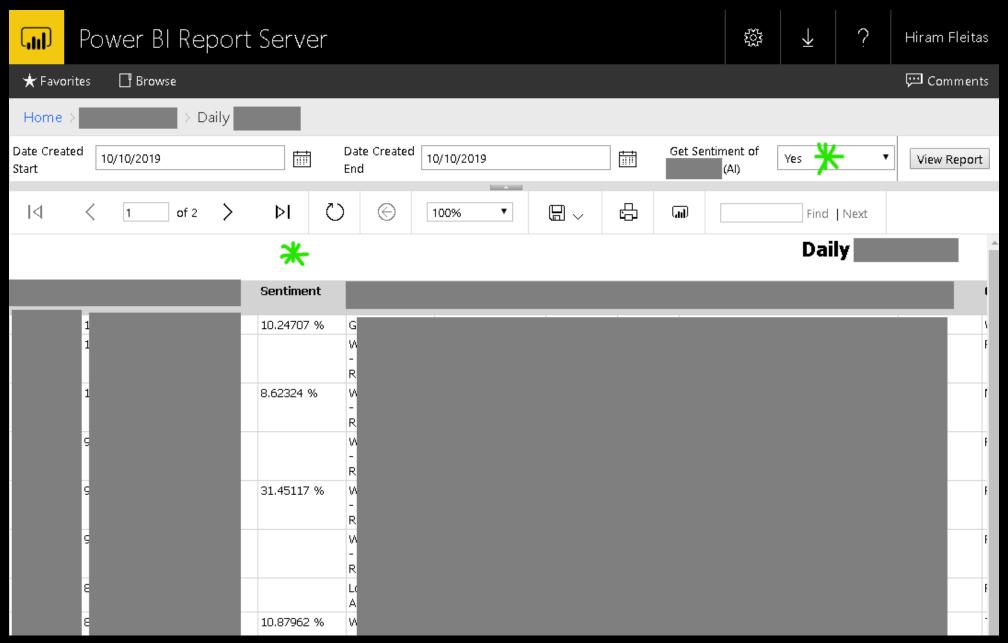
Free account >

```
create or alter proc GetCognitiveAPIQuoteSentiment
as
    set nocount on;
    declare @py nvarchar(max);
    set @py = N'import requests, pprint as pr
from pandas.io.json import json normalize
apikey = "mykey"
api = "https://eastus2.api.cognitive.microsoft.com/text/analytics/v2.0/"
url = api + "sentiment"
df = jsondocs
headers = {
     "Ocp-Apim-Subscription-Key": apikey,
     "content-type": "application/json"
response = requests.post(
     url,
     headers = headers,
     data = df.iloc[0][0].encode()
rds = response.json()
df2 = json normalize(rds, "documents")
pr.pprint(rds)
print(type(df2),df2,sep="\n")
```

Learn more:

github.com/hfleitas/seattle2019/blob/master/2SentimentPrediction/Trollhunters.ipynb

```
drop table if exists apiresults;
    create table apiresults (id int, score float);
    insert into apiresults
    exec sp_execute_external_script @language = N'Python'
       ,@script = @py
        "@input data 1 = N'select * from JsonQuotes'
        "@input data 1 name = N'jsondocs'
        ,@output data 1 name = N'df2'
    select * from apiresults;
   update q
       set q.Sentiment = a.Score
    from
           Quotes q
    inner join apiresults a
       on q.quoteid = a.id
    where q.Sentiment is null;
go
```















Power Virtual Agents - Publish X Power Virtual Agents

Learn more about Power Virtual Agents

Try out the chatbot we made!

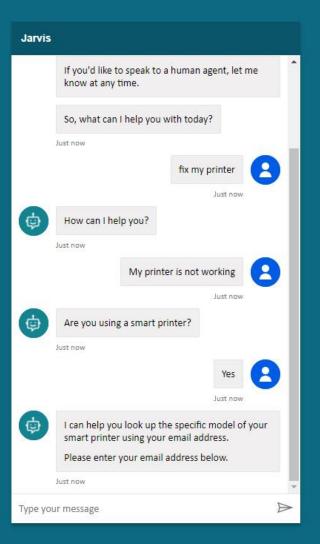
Here are some things my bot can help you with:

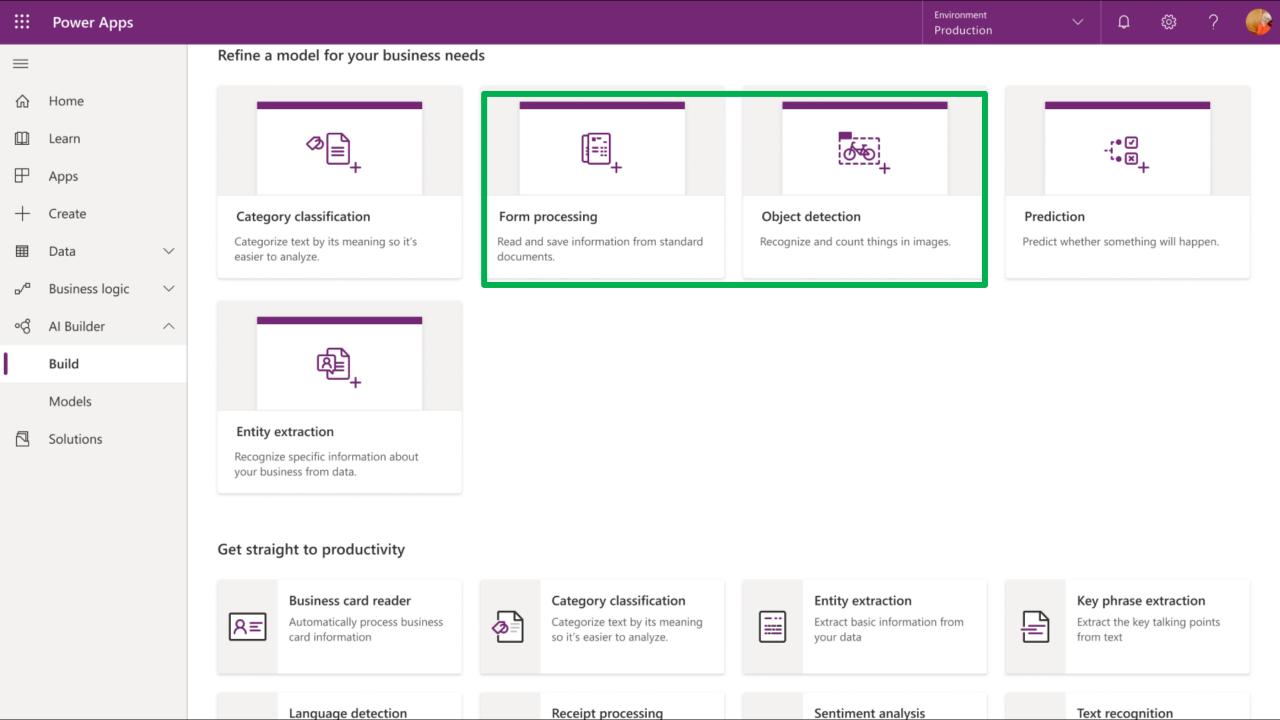
Hello

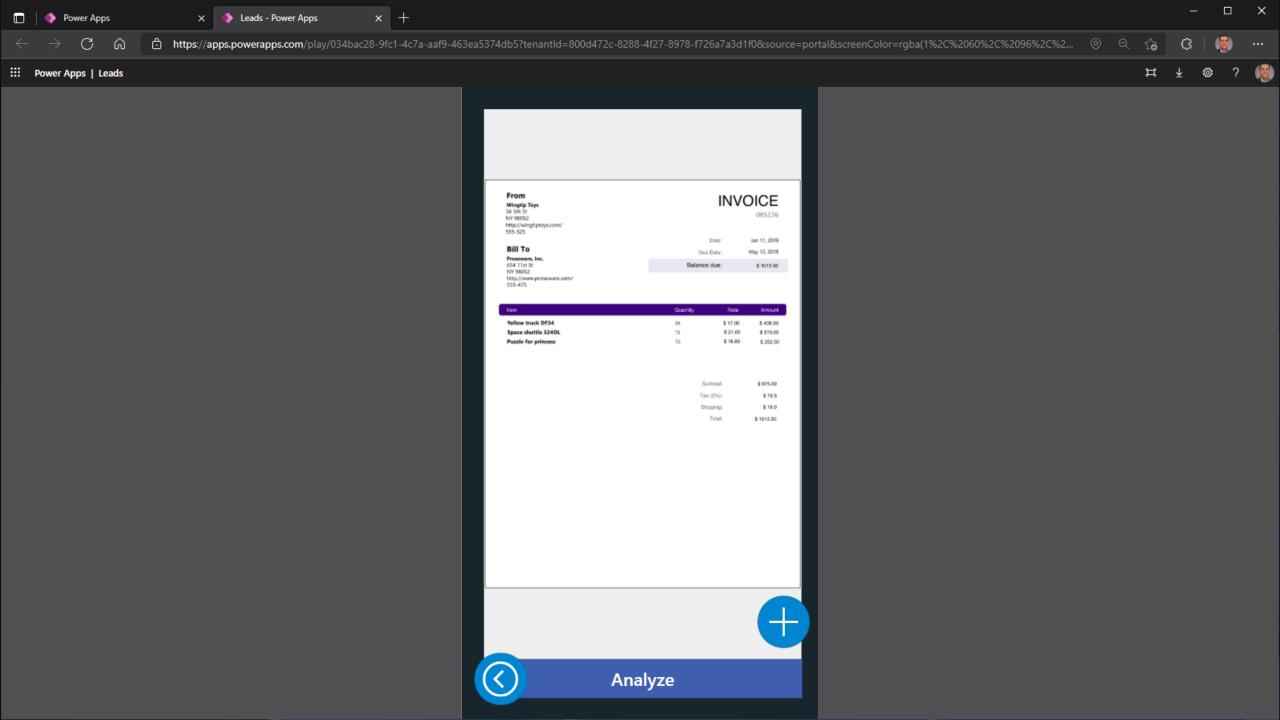
Start over

Talk to a person









The Transformational Digitization: We are people, not interfaces

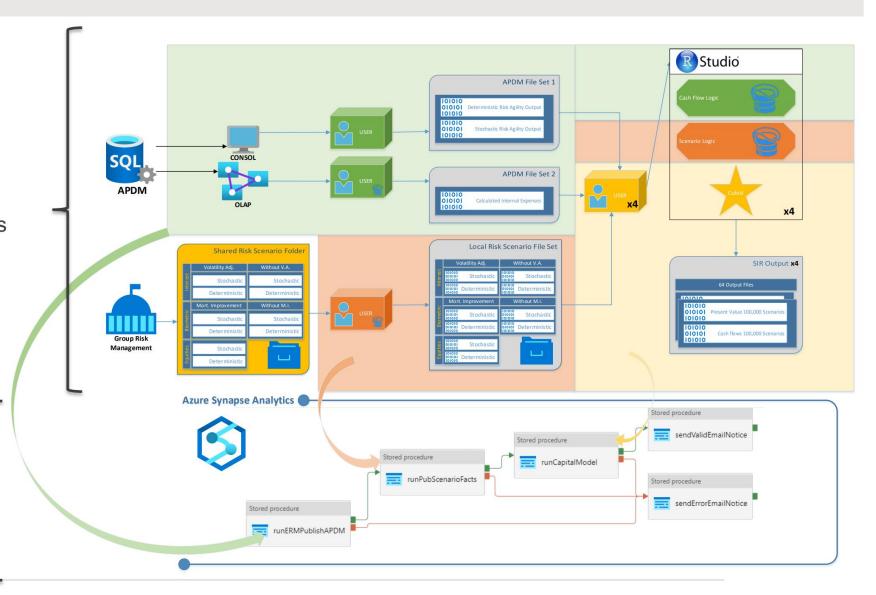
From:

7 user touchpoints
4 unique manual processes
Many, many different files and folders
Significant User Down Time

To:

- 1 Synapse Pipeline
- 3 Primary Stored Procedures
- 1 On Premise SQL Server

Low User Down Time



The Transformational Digitalization: R for R

From: Loop Based Logic

WET Code (Write Everything Twice)

Inefficient and Frequent R Troubleshooting

To: Functional Programming

DRY Code (Don't Repeat Yourself)
Little Need for R Troubleshooting

```
fitA:-cubist(y-cashSeedsA[,j],x=seedModVars,committees=committeesBase,control=cubistControl(extrapolation=extrapolation
scenCashAPredict[,j]<-round(predict(fitA,newdata=scenModVars),6)
(sum(cashSeedsD[j]|!=0)>0)(
fith-cubist(y-cashSeedsD[j])_x-seedbodvarsInt,committees-committeesInt,control-cubistControl(extrapolation-extrapolation)
seccashDerac(t[j]:-round(predict(fitb,newdata-scenbodvarsInt),co
(sum(cashSeeds(j,j)!=0)>0)(
fits-cubist(y-cashSeeds(j,j),x-seedModvarsint,committees-5,neighbors-3,control-cubistControl(extrapolation-extrapolation)
seecashSered(t,j,j)-round(predict(fits,newdata-scenModvarsint),6)
fitIntExp-<ublst(y=intExpSeedsDef[,j],x-seedModVars,committees=committeesInt,neighbors=5,control=cubistControl(extrapolation=extrapolation) scenCashIntExpPredict[,j]<-round(predict(fitIntExp,newdata=scenModVars),6)
f(type[i]=="PVFP"){
   f(sum(cashGed38[,]]1=0)>00]
f(sum(cashGed38[,]]1=0)>00]
f(sum(cashGed38[,]]1=0)>00]
fith-...duist()-cashGed38[,]],x-seedbodvarses,committees.committees.int,control-cubistControl(extrapolation-extrapolation)
seccashBredGit[,]]:-round[predict(fith_newdata-scenbodvarshes),6)
   (sum(cashSeedsc[,j]|=0)>0)(
fitc <cubist(y-cashSeedsc[,j])_x-seedbodvarsRes,committees-committeesInt,control-cubistControl(extrapolation-extrapolation)
seccashCreaf(t,j]</pre>-round(predict(fitc,mendata-scenbodvarsRes),com
   (Sum(cashseeds[,j]|-d))=0)

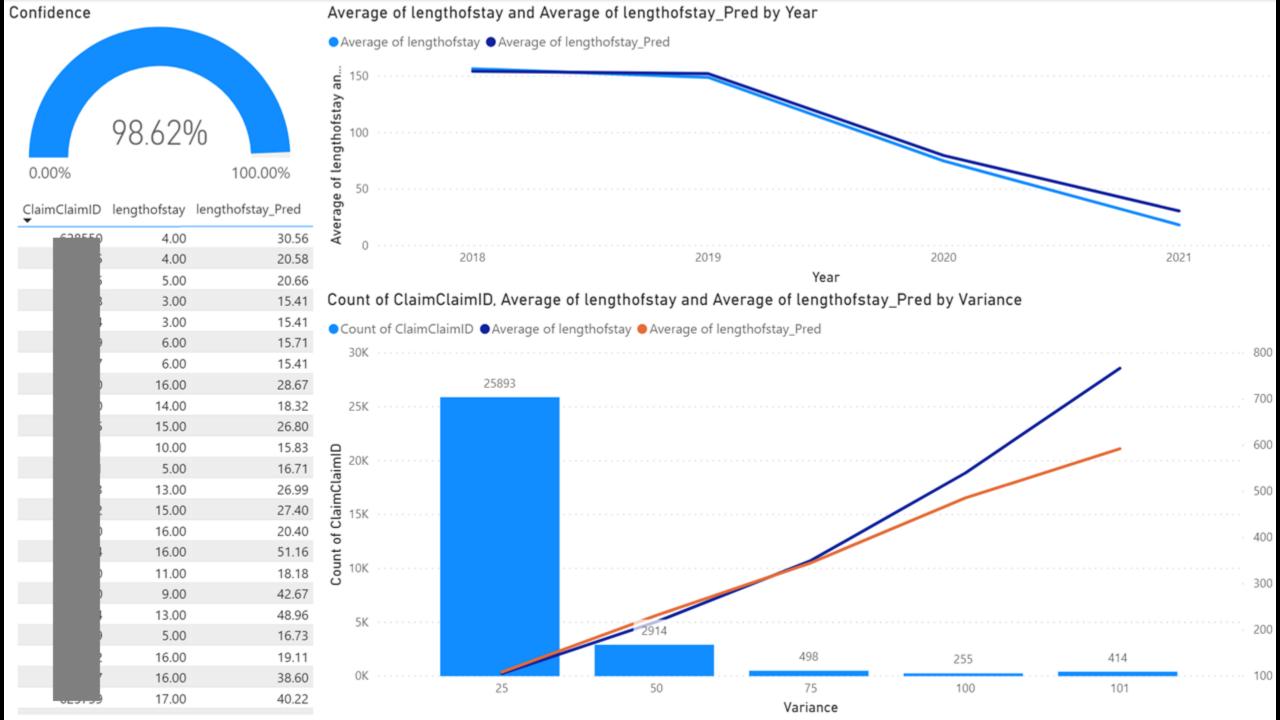
(Sum(cashseeds[,j]|-d))=0)

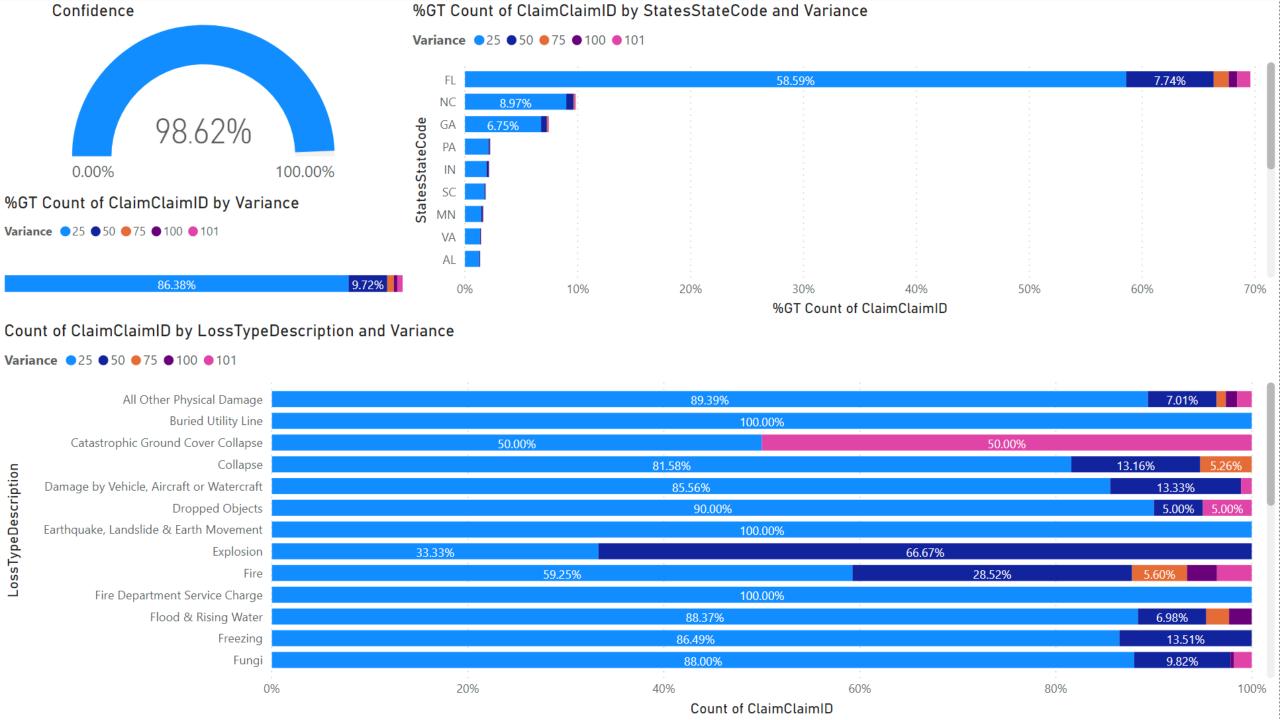
(Sum(cashseeds[,j],-seedModVarsRes,comittees-comitteesInt,control-cubistControl(extrapolation-extrapolation)

sec(GashFred(ci,j)-cnund(predict(fits_newdata-scenbodVarsRes),c)
   (sum(cashSeedsF[,j]!=0)>0)(
fitF < cubist(y-cashSeedsF[,j])_x-seedWodVarsRes,committees-committeesInt,control-cubistControl(extrapolation-extrapolation)
secreashFrend(t,j]</pre>-round(predict(fitF,nendata-scerWodVarsRes),c)
```

```
trainCubist <- function(dt, com = 15, extp = 90){Cubist::cubist(y = getY(dt),x = getX(dt),committees = getCom(dt),control = Cubist::cubistControl(extrapolation = extp))}

predictCubist <- function(aModel, dt){stats::predict(aModel, newdata = getX(dt))}
```







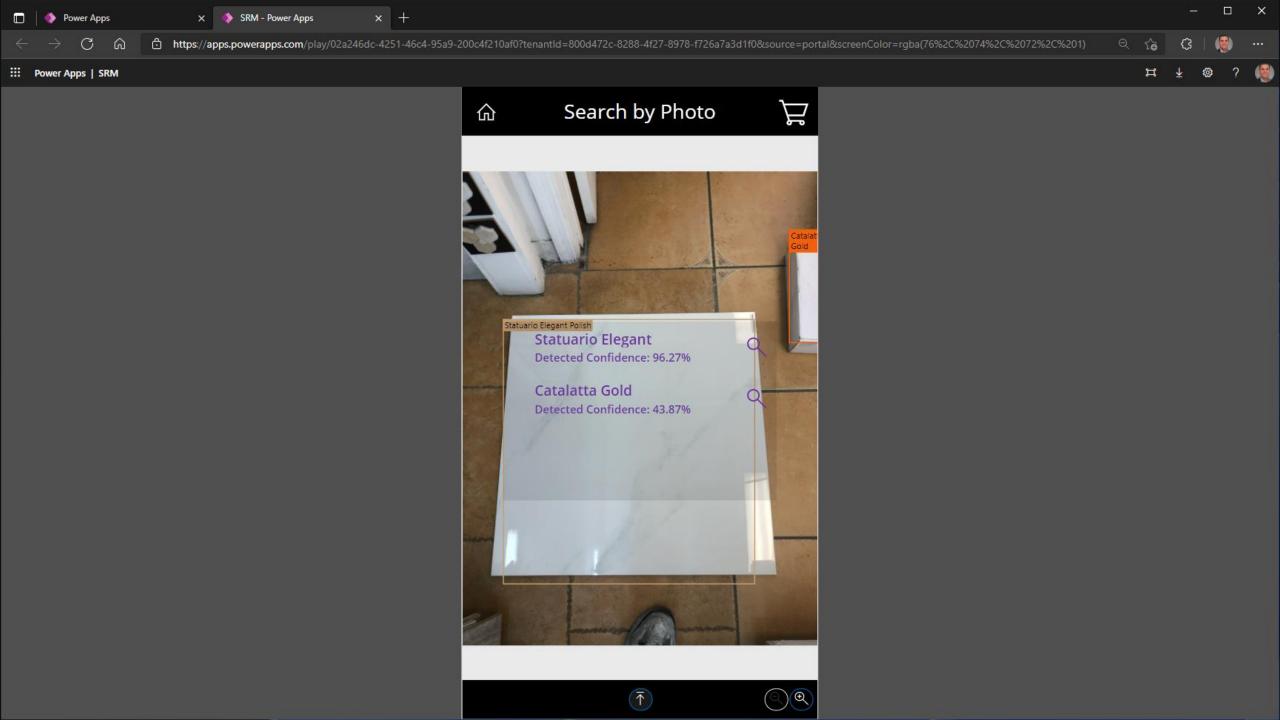




Realtime Scoring

Proc (errors), Train, Score

```
In [3]: -- My simple proc to serialize the model bin, cause train model real time scoring errors.
        Use Hospital Pv
        create or alter proc [GetRTSModelRF]
        declare @info varbinary(max);
        select @info = info from dbo.ColInfo;
        declare @info varbinary(max);
        select @info = info from dbo.ColInfo;
        exec sp_execute_external_script @language = N'Python', @script = N'
        import dill
        from numpy import sqrt
        from pandas import DataFrame
        from revoscalepy import rx set compute context, RxSqlServerData, rx dforest, RxOdbcData, rx serialize model, rx write object, RxLocalSeq
        from microsoftml import adadelta_optimizer
        connection string = "Driver=SQL Server; Server=localhost; Database=Hospital Py; Trusted Connection=true;"
        column info = dill.loads(info)
                Set training dataset, set features and types.
        variables_all = [var for var in column_info]
        #variables to remove = ["eid", "vdate", "discharged", "facid"]
        variables to remove = ["ClaimClaimID", "ClaimDateClosed", "ClaimReportedDate"]
        training variables = [x for x in variables all if x not in variables to remove]
        LoS_Train = RXSqlServerData(sql_query = "SELECT ClaimClaimID, {} FROM LoS WHERE ClaimClaimID IN (SELECT ClaimClaimID from Train_Id)".form
        at(", ".join(training_variables)),
                                    connection_string = connection_string,
                                    column info = column info)
                Specify the variables to keep for the training
        #variables_to_remove = ["eid", "vdate", "discharged", "facid", "lengthofstay"]
        variables_to_remove = ["ClaimClaimID", "ClaimDateClosed", "ClaimReportedDate", "lengthofstay"]
        training variables = [x for x in variables all if x not in variables to remove]
        formula = "lengthofstay ~ " + " + ".join(training_variables)
        ## Train RF Model
        dest = RxOdbcData(connection_string, table = "RTS")
        model = rx_dforest(formula=formula,
                            data=LoS_Train,
                            n tree=40,
                            cp=0.00005,
                            min split=int(sqrt(70000)),
                            max_num_bins=int(sqrt(70000)),
                            seed=5)
        serialized_model = rx_serialize_model(model, realtime_scoring_only = True)
        rx write object(dest, key name="id", key="RF", value name="value", value=serialized model, serialize=False, compress=None, overwrite=Fals
        , @params = N'@info varbinary(max)'
        , @info = @info;
        GO
```





Jupyter





Thank you

Hiram Fleitas

https://aka.ms/hiram

