grvzeYvVhyK9QE3O.GeJIvy5cQY85rKraQZ9KThHJfiTa16dq2vqwC0zJKeBFFKFSx5aeBst2B6lJF0hV

-- API KEY

İLK ETAPTA SİTEDEN ÜYE OLDUM VE AŞAĞIDAKİ SINIRLAMALARDA HAK VERDİ.

metin, ekran görüntüsü, sayı, numara, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şuan jupyterlab de hata aldığım için colabden devam ediyorum..

Api key istiyor aşağıdaki komuttan sonra

import hopsworks

project = hopsworks.login()

fs = project.get\_feature\_store()

ardından bir feature group oluşturdum.

metin, yazı tipi, ekran görüntüsü, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Feature groups içerisindeki featurelarn açıklamaları yoktu ardından açıklama ekledim.

Ardından LOGLARI AYARLADIM. (BUNU İYİ ANLA) FEAST’E GÖRE PLUS.

metin, ekran görüntüsü, bilgisayar, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

TRAİNİNG İÇİN FEATURE GROUPLARI ÇEKTİM.

QUERY İLE EĞİTİMDE KULLANACAĞIM VERİLERİ ÇEKTİM.

MESELA HOPSWORKS TE BÖYLE ŞEYLER VAR

# Load transformation functions.

label\_encoder = fs.get\_transformation\_function(name="label\_encoder")

# Map features to transformations.

transformation\_functions = {

    "category": label\_encoder

}

ARDINDAN FEATURE VIEW OLUŞTURDUM. BU QUERY İLE FİLTRELENEN ÖZELLİKLERDEN OLUŞTU.

metin, ekran görüntüsü, sayı, numara, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

TRAINING DATA OLUŞTURUYORUM. LOGLARI GÖREBİLİYORUM, İNDİREBİLİYORUM. BU KISIM UZUN SÜRDÜ NEDENSE.

MODELİ EĞİTTİM VE MODEL REGISTRY KAYDETTİM.

ekran görüntüsü, metin, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

DEPLOY İÇİN

ÖZELLİK GÖRÜNÜMLERİNİ ALDIM

MODEL REGISTRY’DEN MODELİ ALDIM.

MODEL XGBOOSTTU V1 OLARAK KAYIT ETMİŞTİM AMA HATA ALDIM XGBOOST MODELİNİ GÜNCELLEDİM BU SEFER OTOMATİK OLARAK V2 OLARAK KAYDETTİ.

metin, ekran görüntüsü, sayı, numara, paralel içeren bir resim

Açıklama otomatik olarak oluşturuldu

ALERT EKLEYEBİLİYORSUN MAİLE FALAN

LOGLAR PROJECT SETTINGS E KAYIT OLUYOR

MODELLER ŞU ŞEKİL KAYDOLUYOR

metin, yazılım, bilgisayar simgesi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

Yarın sabah çağrı abinin ses kaydını dinle. Bakmam gereken özellikler vs nelermiş bir anla. Feast ile karşılaştır. Feast gibi hopsworks bölümü yap. Hopsworks ile tek çekincem community ve kaynak. Gerçi hepsinde az. Ama bence feature store belli başlı şeyler çözüldükten sonra basit.

FRAUD ONLINE ÇALIŞMASI

* Create feature group dendiği zaman sadece meta verilerden oluşan bir feature group oluşuyor.

trans\_fg = fs.get\_or\_create\_feature\_group(

    name="transactions\_fraud\_online\_fg",

    version=1,

    description="Transaction data",

    primary\_key=['cc\_num'],

    event\_time='datetime',

    online\_enabled=True,

)

Herhangi bir veriyi saklamaz ve hatta veriler için tanımlanmış bir şemaya sahip değildir. Özellik grubunu kalıcı hale getirmek için, insert işlevini kullanarak onu ilişkili verilerle doldurmanız gerekir.

Örneğin

trans\_fg.insert(trans\_df)

Böylece trans\_df dataframe’imdeki özellikleri eklemiş oldum.

İstatistiksel vb. bakış açıları ile verilere bakabiliyorsun. “Statistics” bölümünden..

Exploration

In the Hopsworks feature store, the metadata allows for multiple levels of explorations and review. Here you will explore a few of those capacities.

🔎 Search

Using the search function in the UI, you can query any aspect of the feature groups, feature\_view and training data that was previously created.

📊 Statistics

You can also enable statistics in one or all the feature groups.

trans\_fg = fs.get\_or\_create\_feature\_group("transactions\_fraud\_online\_fg", version=1)

trans\_fg.statistics\_config = {

    "enabled": True,

    "histograms": True,

    "correlations": True

}

trans\_fg.update\_statistics\_config()

trans\_fg.compute\_statistics()

**İSTATİSTİK İÇİN BAŞKA NELER EKLENEBİLİYOR ARAŞTIR..**

**HOPSWORKS’DE “get\_transformation\_function” var. Örneğin bu içerisinde “label\_encoder” alabiliyor. *Bu fonksiyon içerisine başka neler alabiliyor bunları öğrenmeliyim.***

diyagram, metin, ekran görüntüsü, plan içeren bir resim

Açıklama otomatik olarak oluşturuldu

Devam eden süreçte feature selection yapıyoruz. Feature groupları ayrı bir notebookta çalıştığım için önce load ediyorum (get\_feature\_group) ardından iki feature group olduğu için onları birbiri ile joinliyorum.

Transformasyondan sonra create feature view ile özellik görünümlerini oluşturuyorum.

Feature view oluştuktan sonra train-test split işlemlerine geçiyorum. Burada gene hopsworksden yararlanıyorum.

*Dataset with train/test splits can be created using fs.create\_train\_test\_split() method. Dataset with train/valid/test splits can be created using fs.create\_train\_validation\_test\_split() method.*

***You can use event time filters like train\_start, train\_end, valid\_start, valid\_end... Values can be either in unix, string and datetime format.***

***Or, use validation\_size and test\_size parameters.***

# Training/Test splits, datasets creation. Using timerange arguments.

train\_start = "2022/01/01"

train\_end = "2022/03/10"

test\_start = "2022/03/10"

test\_end = "2022/03/31"

td\_version, td\_job = feature\_view.create\_train\_test\_split(

    train\_start=train\_start,

    train\_end=train\_end,

    test\_start=test\_start,

    test\_end=test\_end,

    data\_format = "csv",

    coalesce = True,

    write\_options = {'wait\_for\_job': True},

    )

X\_train, X\_test, y\_train, y\_test = feature\_view.get\_train\_test\_split(1)

MODELİ EĞİTTİM, ŞEMA OLUŞTURDUM VE ARDINDAN kaydettim.

# The 'fraud\_online\_model' directory will be saved to the model registry

model\_dir="fraud\_online\_model"

if os.path.isdir(model\_dir) == False:

    os.mkdir(model\_dir)

joblib.dump(clf, model\_dir + '/xgboost\_fraud\_online\_model.pkl')

fig.savefig(model\_dir + "/confusion\_matrix.png")

mr = project.get\_model\_registry()

fraud\_model = mr.python.create\_model(

    name="xgboost\_fraud\_online\_model",

    metrics=metrics,

    model\_schema=model\_schema,

    input\_example=[4467360740682089], # for testing deployments

    description="Fraud Online Predictor")

fraud\_model.save(model\_dir)

NOTEBOOKTA

Model deployment için bir script yazdık ve onu api olarak hopsworks’e yükledik.

dataset\_api = project.get\_dataset\_api()

uploaded\_file\_path = dataset\_api.upload("predict\_example.py", "Models", overwrite=True)

predictor\_script\_path = os.path.join("/Projects", project.name, uploaded\_file\_path)

Ardından deployment oluşturduk.

metin, yazılım, sayı, numara, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Burada endpoint vs görebiliyorum.

Deployment’ı kaydettikten sonra “deployment.start” diyerek başlatabiliyorum.

Prediction’ı gene yapamadım 3. Colab notebookunda..

API REFERENCE NOTLARIM

<https://docs.hopsworks.ai/feature-store-api/latest/generated/api/connection_api/>

# CONNECTION

Connection projeye özeldir, dolayısıyla projenin kendi özellik mağazasına ve aynı zamanda bağlandığınız projeyle paylaşılan herhangi bir özellik mağazasına da erişebilirsiniz.

import hsfs

conn = hsfs.connection(

'my\_instance', # Hostname of your Feature Store instance

443, # Port to reach your Hopsworks instance, defaults to 443

'my\_project', # Name of your Hopsworks Feature Store project

api\_key\_file='featurestore.key', # The file containing the API key generated above

hostname\_verification=True) # Disable for self-signed certificates

)

fs = conn.get\_feature\_store() # Get the project's default feature store

# or

import hopsworks

project = hopsworks.login()

fs = project.get\_feature\_store()

# EXPECTATION SUITE

Bu kısmı pek anlamadım.

# FEATURE STORE

## get\_storage\_connector

* Storage connector, yürütme motorunun belirli bir depolama birimine okuması ve yazması için gereken tüm bilgileri içerir. Bu depolama S3, JDBC uyumlu bir veritabanı veya dağıtılmış dosya sistemi HOPSFS olabilir.
* Çevrimiçi özellik deposuna bağlanmak için örneğin

# connect to the Feature Store

fs = ...

sc = fs.get\_storage\_connector("demo\_fs\_meb10000\_Training\_Datasets")

## get\_transformation\_functions

* Sanırım sadece encode kodları var bunun içerisinde. Kendi bölümü içerisinde de farklı bir bilgi göremedim. FEAST ile aynı diyebiliriz bu bağlamda.

## fs.sql

* Online ve offline feature store database’e sorgu

# FEATUREGROUP

* Çeşitli methotlar var.
  + Add\_tag
  + Append\_features
  + As\_of
  + Filter
    - Filtrelenmiş özellik grubuna daha sonra başka bir özellik grubuyla katılmayı planlıyorsanız, filtrelenmiş özelliği ilgili özellik grubundan açıkça seçtiğinizden emin olun
  + Get\_all\_Validation\_reports
    - Varsa, özellik grubuna eklenen en son doğrulama raporunu döndürün.
  + Select
  + Select\_except
  + Show
  + Update\_deprecated
    - Özellik grubunu kullanımdan kaldırır.
  + Update\_description
    - Özellik grubunun açıklamasını günceller
  + Update\_feature\_description
    - Özellik grubundaki tek bir özelliğin açıklamasını günceller

# SpineGroup

Feature group a benzer

? ONLINE VE OFFLINE STORE OLARAK NE KULLANIYORUZ? ONLINE VE OFFLINE FEATURE GROUP FARKINI ANLAYAMADIM.