ArupChakraborty_Assignment3.1

May 25, 2025

1 Neighborhood Feature Group Creation

This notebook processes housing and Google Maps data to create a feature group for neighborhoods.

[5]:
```
import pandas as pd
import numpy as np
from datetime import datetime
from sklearn.preprocessing import OneHotEncoder
from collections import defaultdict
```

[6]:
```
# Load data
housing_df = pd.read_csv("housing.csv")
gmaps_df = pd.read_csv("housing_gmaps_data_raw.csv")
```

[78]:
```
housing_df.head()
```

[78]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	\
0	-122.23	37.88	41.0	880.0	129.0	
1	-122.22	37.86	21.0	7099.0	1106.0	
2	-122.24	37.85	52.0	1467.0	190.0	
3	-122.25	37.85	52.0	1274.0	235.0	
4	-122.25	37.85	52.0	1627.0	280.0	

	population	households	median_income	median_house_value	ocean_proximity
0	322.0	126.0	8.3252	452600.0	NEAR BAY
1	2401.0	1138.0	8.3014	358500.0	NEAR BAY
2	496.0	177.0	7.2574	352100.0	NEAR BAY
3	558.0	219.0	5.6431	341300.0	NEAR BAY
4	565.0	259.0	3.8462	342200.0	NEAR BAY

[8]:
```
gmaps_df.head()
```

[8]:

	street_number	route	locality-political	\
0	3130	Grizzly Peak Boulevard	Berkeley	
1	2005	Tunnel Road	Oakland	
2	6886	Chabot Road	Oakland	
3	6365	Florio Street	Oakland	
4	5407	Bryant Avenue	Oakland	

```
  administrative_area_level_2-political administrative_area_level_1-political  \
0                        Alameda County                              California
1                        Alameda County                              California
2                        Alameda County                              California
3                        Alameda County                              California
4                        Alameda County                              California

  country-political  postal_code  \
0     United States      94705.0
1     United States      94611.0
2     United States      94618.0
3     United States      94618.0
4     United States      94618.0

                                        address  longitude  latitude  …  \
0  3130 Grizzly Peak Blvd, Berkeley, CA 94705, USA    -122.23     37.88  …
1           2005 Tunnel Rd, Oakland, CA 94611, USA    -122.22     37.86  …
2           6886 Chabot Rd, Oakland, CA 94618, USA    -122.24     37.85  …
3           6365 Florio St, Oakland, CA 94618, USA    -122.25     37.85  …
4          5407 Bryant Ave, Oakland, CA 94618, USA    -122.25     37.84  …

  establishment-natural_feature  airport-establishment-point_of_interest  \
0                           NaN                                       NaN
1                           NaN                                       NaN
2                           NaN                                       NaN
3                           NaN                                       NaN
4                           NaN                                       NaN

  political-sublocality-sublocality_level_1  \
0                                       NaN
1                                       NaN
2                                       NaN
3                                       NaN
4                                       NaN

  administrative_area_level_3-political post_box  \
0                                   NaN      NaN
1                                   NaN      NaN
2                                   NaN      NaN
3                                   NaN      NaN
4                                   NaN      NaN

  establishment-light_rail_station-point_of_interest-transit_station  \
0                                                NaN
1                                                NaN
2                                                NaN
```

```
3                                                   NaN
4                                                   NaN

  establishment-point_of_interest  \
0                             NaN
1                             NaN
2                             NaN
3                             NaN
4                             NaN

  aquarium-establishment-park-point_of_interest-tourist_attraction-zoo  \
0                                                   NaN
1                                                   NaN
2                                                   NaN
3                                                   NaN
4                                                   NaN

  campground-establishment-lodging-park-point_of_interest-rv_park-
tourist_attraction  \
0                                                   NaN
1                                                   NaN
2                                                   NaN
3                                                   NaN
4                                                   NaN

  cemetery-establishment-park-point_of_interest
0                                           NaN
1                                           NaN
2                                           NaN
3                                           NaN
4                                           NaN

[5 rows x 30 columns]
```

[18]:
```
import boto3
import sagemaker

original_boto3_version = boto3.__version__
%pip install 'boto3>1.17.21'
```

```
sagemaker.config INFO - Not applying SDK defaults from location:
/etc/xdg/sagemaker/config.yaml
sagemaker.config INFO - Not applying SDK defaults from location:
/home/sagemaker-user/.config/sagemaker/config.yaml
Requirement already satisfied: boto3>1.17.21 in /opt/conda/lib/python3.12/site-
packages (1.37.1)
Requirement already satisfied: botocore<1.38.0,>=1.37.1 in
```

```
/opt/conda/lib/python3.12/site-packages (from boto3>1.17.21) (1.37.1)
Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in
/opt/conda/lib/python3.12/site-packages (from boto3>1.17.21) (1.0.1)
Requirement already satisfied: s3transfer<0.12.0,>=0.11.0 in
/opt/conda/lib/python3.12/site-packages (from boto3>1.17.21) (0.11.3)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in
/opt/conda/lib/python3.12/site-packages (from
botocore<1.38.0,>=1.37.1->boto3>1.17.21) (2.9.0.post0)
Requirement already satisfied: urllib3!=2.2.0,<3,>=1.25.4 in
/opt/conda/lib/python3.12/site-packages (from
botocore<1.38.0,>=1.37.1->boto3>1.17.21) (2.4.0)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.12/site-
packages (from python-
dateutil<3.0.0,>=2.1->botocore<1.38.0,>=1.37.1->boto3>1.17.21) (1.17.0)
Note: you may need to restart the kernel to use updated packages.
```

[19]:
```
from sagemaker.session import Session

region = boto3.Session().region_name

boto_session = boto3.Session(region_name=region)

sagemaker_client = boto_session.client(service_name="sagemaker",␣
 ↪region_name=region)
featurestore_runtime = boto_session.client(
    service_name="sagemaker-featurestore-runtime", region_name=region
)

feature_store_session = Session(
    boto_session=boto_session,
    sagemaker_client=sagemaker_client,
    sagemaker_featurestore_runtime_client=featurestore_runtime,
)
```

S3 Bucket Setup For The OfflineStore

[20]:
```
default_s3_bucket_name = feature_store_session.default_bucket()
prefix = "sagemaker-featurestore-rup-assignment3"

print(default_s3_bucket_name)
```

```
sagemaker-us-east-1-672518276407
```

[21]:
```
from sagemaker import get_execution_role

# You can modify the following to use a role of your choosing. See the␣
 ↪documentation for how to create this.
role = get_execution_role()
```

```
print(role)
```

```
arn:aws:iam::672518276407:role/LabRole
```

FeatureGroup

[50]:
```
import pandas as pd
import numpy as np
from datetime import datetime

# Load input files
housing_df = pd.read_csv("./housing.csv")
gmaps_df = pd.read_csv("./housing_gmaps_data_raw.csv")

# Merge on latitude and longitude (approximate spatial join)
merged_df = pd.merge(housing_df, gmaps_df, on=["latitude", "longitude"],␣
 ↪how="left")

# Fill missing total_bedrooms by average per postal_code
merged_df["total_bedrooms"] = merged_df.
 ↪groupby("postal_code")["total_bedrooms"].transform(
    lambda x: x.fillna(x.mean())
)

# Discretize housing median age into bins of 10 years
merged_df["median_house_age_grouped"] = pd.cut(
    merged_df["housing_median_age"],
    bins=range(0, 110, 10),
    labels=[i for i in range(0, 100, 10)],
    right=False
)

# One-hot encode ocean_proximity
ocean_prox_cols = {
    "<1h ocean": "less_than_1h_ocean",
    "inland": "inland",
    "island": "island",
    "near bay": "near_bay",
    "near ocean": "near_ocean"
}
ocean_dummies = pd.get_dummies(merged_df["ocean_proximity"].str.lower())
for original, renamed in ocean_prox_cols.items():
    if original in ocean_dummies.columns:
        merged_df[renamed] = ocean_dummies[original]
    else:
        merged_df[renamed] = 0

# Group by neighborhood and aggregate
```

```
agg_df = merged_df.groupby("neighborhood-political").agg({
    "median_house_value": lambda x: min(x.mean(), 500000),
    "median_house_age_grouped": lambda x: round(x.astype(float).mean()),
    "households": lambda x: int(np.ceil(x.mean())),
    "total_bedrooms": "sum",
    "households": "sum"
}).rename(columns={
    "median_house_value": "median_house_value_capped",
    "households": "total_households"
}).reset_index()

# Bedrooms per household
agg_df["bedrooms_per_household"] = agg_df["total_bedrooms"] /␣
 ↪agg_df["total_households"]
agg_df.drop(columns=["total_bedrooms"], inplace=True)

# Add averaged one-hot ocean proximity columns
ocean_avg = merged_df.groupby("neighborhood-political")[list(ocean_prox_cols.
 ↪values())].mean().round().astype(int).reset_index()
agg_df = pd.merge(agg_df, ocean_avg, on="neighborhood-political", how="left")

# Add event time and rename primary key
agg_df["event_time"] = datetime.utcnow().strftime("%Y-%m-%dT%H:%M:%SZ")
agg_df.rename(columns={"neighborhood-political": "primary_key"}, inplace=True)

# Save to CSV
feature_group_path = "./neighborhood_feature_group.csv"
agg_df.to_csv(feature_group_path, index=False)

feature_group_path
```

```
/tmp/ipykernel_587/3291860145.py:61: DeprecationWarning:
datetime.datetime.utcnow() is deprecated and scheduled for removal in a future
version. Use timezone-aware objects to represent datetimes in UTC:
datetime.datetime.now(datetime.UTC).
  agg_df["event_time"] = datetime.utcnow().strftime("%Y-%m-%dT%H:%M:%SZ")
```

[50]:
```
'./neighborhood_feature_group.csv'
```

[51]:
```
agg_df = pd.read_csv(feature_group_path)
agg_df.head()
```

[51]:

	primary_key	median_house_value_capped	\
0	28 Palms	222200.000000	
1	Acorn Industrial	81300.000000	
2	Adams Hill	250733.333333	
3	Agua Mansa Industrial Corridor	112300.000000	

```
4                        Al Tahoe              109180.000000

   median_house_age_grouped  total_households  bedrooms_per_household  \
0                        20             923.0                1.017335
1                        50             147.0                1.659864
2                        35            2962.0                1.053680
3                        10             516.0                1.102713
4                        20            1244.0                1.606913

   less_than_1h_ocean  inland  island  near_bay  near_ocean  \
0                   1       0       0         0           0
1                   0       0       0         1           0
2                   1       0       0         0           0
3                   0       1       0         0           0
4                   0       1       0         0           0

             event_time
0  2025-05-24T22:57:55Z
1  2025-05-24T22:57:55Z
2  2025-05-24T22:57:55Z
3  2025-05-24T22:57:55Z
4  2025-05-24T22:57:55Z
```

[52]:
```python
import boto3
import pandas as pd
import time
import uuid
from sagemaker.session import Session
from sagemaker.feature_store.feature_group import FeatureGroup

# --- Setup Clients and Session ---
region = boto3.Session().region_name
boto_session = boto3.Session(region_name=region)
sagemaker_client = boto_session.client("sagemaker", region_name=region)
featurestore_runtime = boto_session.client("sagemaker-featurestore-runtime",␣
 ↪region_name=region)

feature_store_session = Session(
    boto_session=boto_session,
    sagemaker_client=sagemaker_client,
    sagemaker_featurestore_runtime_client=featurestore_runtime,
)

default_s3_bucket_name = feature_store_session.default_bucket()
prefix = "sagemaker-featurestore-rup-assignment3"
record_identifier_feature_name = "primary_key"
event_time_feature_name = "event_time"
```

```
feature_group_name = f"neighborhood-feature-group-{uuid.uuid4().hex[:8]}"

# --- Load and Clean Data ---
df = pd.read_csv("neighborhood_feature_group.csv")

# Fix column names to be SageMaker-compliant
df.rename(columns={
    "<1h ocean": "less_than_1h_ocean",
    "near bay": "near_bay",
    "near ocean": "near_ocean"
}, inplace=True)

# Convert event_time to ISO format string
df[event_time_feature_name] = pd.to_datetime(df[event_time_feature_name]).dt.
 ↪strftime("%Y-%m-%dT%H:%M:%SZ")

# --- Create Feature Group ---
neighborhood_fg = FeatureGroup(name=feature_group_name,␣
 ↪sagemaker_session=feature_store_session)

# Load schema from DataFrame
neighborhood_fg.load_feature_definitions(data_frame=df)

# Create the Feature Group
neighborhood_fg.create(
    s3_uri=f"s3://{default_s3_bucket_name}/{prefix}",
    record_identifier_name=record_identifier_feature_name,
    event_time_feature_name=event_time_feature_name,
    role_arn=feature_store_session.get_caller_identity_arn(),
    enable_online_store=True
)

# Wait for creation to complete
def wait_for_feature_group_creation_complete(feature_group):
    status = feature_group.describe().get("FeatureGroupStatus")
    while status == "Creating":
        print("Waiting for Feature Group Creation...")
        time.sleep(5)
        status = feature_group.describe().get("FeatureGroupStatus")
    if status != "Created":
        raise RuntimeError(f"Failed to create feature group {feature_group.
 ↪name}")
    print(f" Feature Group {feature_group.name} successfully created.")

wait_for_feature_group_creation_complete(neighborhood_fg)

# --- Ingest Data ---
```

```
records = df.to_dict(orient="records")
for record in records:
    record["event_time"] = record["event_time"]
    featurestore_runtime.put_record(
        FeatureGroupName=feature_group_name,
        Record=[{"FeatureName": k, "ValueAsString": str(v)} for k, v in record.
 ↪items()]
    )

print(f" Data ingestion complete for Feature Group: {feature_group_name}")
```

```
Waiting for Feature Group Creation…
Waiting for Feature Group Creation…
Waiting for Feature Group Creation…
Waiting for Feature Group Creation…
Waiting for Feature Group Creation…
 Feature Group neighborhood-feature-group-e36acd0c successfully created.
 Data ingestion complete for Feature Group: neighborhood-feature-group-e36acd0c
```

1.0.1 Query feature values from Online Store

[77]:
```
import pandas as pd

neighborhoods_to_query = ["Brooktree", "Fisherman's Wharf", "Los Osos"]
query_results = []

for key in neighborhoods_to_query:
    try:
        response = featurestore_runtime.get_record(
            FeatureGroupName=feature_group_name,
            RecordIdentifierValueAsString=key
        )
        if "Record" in response:
            record = {feature["FeatureName"]: feature["ValueAsString"] for␣
 ↪feature in response["Record"]}
            query_results.append(record)
        else:
            print(f" No record found for '{key}' in online store.")
    except featurestore_runtime.exceptions.ResourceNotFound:
        print(f" FeatureGroup '{feature_group_name}' not found.")
    except Exception as e:
        print(f" Unexpected error querying '{key}': {e}")

# Display results
if query_results:
    query_df = pd.DataFrame(query_results)
    print(query_df)
```

```
else:
    print(" No records retrieved from online store.")
```

```
         primary_key            event_time less_than_1h_ocean inland island  \
0          Brooktree  2025-05-24T22:57:55Z                  1      0      0
1  Fisherman's Wharf  2025-05-24T22:57:55Z                  0      0      0
2           Los Osos  2025-05-24T22:57:55Z                  0      0      0

  near_bay near_ocean median_house_value_capped median_house_age_grouped  \
0        0          0                  257400.0                        0
1        1          0                  500000.0                       50
2        0          1                  221612.5                       11

  total_households bedrooms_per_household
0             1438     0.3366418004804652
1              250                  1.268
2             4894     1.0502656313853698
```

1.0.2 Build Training Dataset from the Offline Store

[54]:
```
import boto3

region = boto3.Session().region_name
account_id = boto3.client("sts").get_caller_identity()["Account"]
default_s3_bucket_name = f"sagemaker-feature-store-{region}-{account_id}"

s3_client = boto3.client("s3", region_name=region)

try:
    s3_client.head_bucket(Bucket=default_s3_bucket_name)
    print(f" Bucket already exists: {default_s3_bucket_name}")
except s3_client.exceptions.ClientError as e:
    print(f" Bucket not found. Creating: {default_s3_bucket_name}")
    if region == "us-east-1":
        s3_client.create_bucket(Bucket=default_s3_bucket_name)
    else:
        s3_client.create_bucket(
            Bucket=default_s3_bucket_name,
            CreateBucketConfiguration={"LocationConstraint": region}
        )
    print(f" Bucket created: {default_s3_bucket_name}")
```

```
Bucket already exists: sagemaker-feature-store-us-east-1-672518276407
```

[69]:
```
import boto3
import pandas as pd
import uuid
```

```
from datetime import datetime
from sagemaker import get_execution_role

# AWS Setup
boto_session = boto3.Session(region_name=region)
sagemaker_client = boto_session.client("sagemaker")
account_id = boto_session.client("sts").get_caller_identity()["Account"]
default_s3_bucket_name = f"sagemaker-feature-store-{region}-{account_id}"
prefix = "featurestore/neighborhood-v1"
s3_uri = f"s3://{default_s3_bucket_name}/{prefix}"
role = get_execution_role()

# Load and prepare dataset
df = pd.read_csv("neighborhood_feature_group.csv")
df.rename(columns={
    "<1h ocean": "less_than_1h_ocean",
    "near bay": "near_bay",
    "near ocean": "near_ocean"
}, inplace=True)
df["event_time"] = pd.to_datetime(df["event_time"]).dt.strftime("%Y-%m-%dT%H:%M:
 ↪%SZ")

# Define feature group
feature_group_name = f"neighborhood-feature-group-{uuid.uuid4().hex[:8]}"
feature_definitions = [
    {"FeatureName": "primary_key", "FeatureType": "String"},
    {"FeatureName": "event_time", "FeatureType": "String"},
    {"FeatureName": "less_than_1h_ocean", "FeatureType": "Integral"},
    {"FeatureName": "inland", "FeatureType": "Integral"},
    {"FeatureName": "island", "FeatureType": "Integral"},
    {"FeatureName": "near_bay", "FeatureType": "Integral"},
    {"FeatureName": "near_ocean", "FeatureType": "Integral"},
    {"FeatureName": "median_house_value_capped", "FeatureType": "Fractional"},
    {"FeatureName": "median_house_age_grouped", "FeatureType": "Integral"},
    {"FeatureName": "total_households", "FeatureType": "Integral"},
    {"FeatureName": "bedrooms_per_household", "FeatureType": "Fractional"},
]

# Create Feature Group (with DataCatalogConfig + DisableGlueTableCreation =␣
 ↪True)
sagemaker_client.create_feature_group(
    FeatureGroupName=feature_group_name,
    RecordIdentifierFeatureName="primary_key",
    EventTimeFeatureName="event_time",
    FeatureDefinitions=feature_definitions,
    RoleArn=role,
    OnlineStoreConfig={"EnableOnlineStore": True},
```

```
    OfflineStoreConfig={
        "S3StorageConfig": {"S3Uri": s3_uri},
        "DisableGlueTableCreation": True,
        "DataCatalogConfig": {
            "TableName": feature_group_name.replace("-", "_"),
            "Catalog": "AwsDataCatalog",
            "Database": "sagemaker_featurestore"
        }
    },
    Description="Neighborhood Feature Group with custom offline store"
)
print(feature_group_name)
```

```
neighborhood-feature-group-578f3492
```

[71]:
```
# Wait until the feature group is ACTIVE
import time

def wait_for_fg_ready(name):
    sm = boto3.client("sagemaker", region_name=region)
    while True:
        status = sm.
 ↪describe_feature_group(FeatureGroupName=name)["FeatureGroupStatus"]
        print(f"Status: {status}")
        if status == "Created":
            break
        elif status == "CreateFailed":
            raise RuntimeError(" Feature group creation failed.")
        time.sleep(5)

wait_for_fg_ready(feature_group_name)
```

```
Status: Created
```

[72]:
```
import boto3
import pandas as pd
import numpy as np
from datetime import datetime

# --- Set up Boto3 Client ---
boto_session = boto3.Session(region_name=region)
featurestore_runtime = boto_session.client("sagemaker-featurestore-runtime",␣
 ↪region_name=region)

# --- Load and Sanitize Data ---
df = pd.read_csv('neighborhood_feature_group.csv')
```

```
# Fix potential invalid column names if needed (optional)
df.rename(columns={
    "<1h ocean": "less_than_1h_ocean",
    "near bay": "near_bay",
    "near ocean": "near_ocean"
}, inplace=True)

# Ensure correct types
df["event_time"] = pd.to_datetime(df["event_time"]).dt.strftime("%Y-%m-%dT%H:%M:
 ↪%SZ")
df["total_households"] = pd.to_numeric(df["total_households"], errors="coerce").
 ↪fillna(0).clip(upper=9223372036854775807).astype(np.int64)
df["median_house_age_grouped"] = pd.to_numeric(df["median_house_age_grouped"],␣
 ↪errors="coerce").fillna(0).astype(np.int64)

# --- Ingest Records ---
success, failed = 0, 0
for record in df.to_dict(orient="records"):
    try:
        featurestore_runtime.put_record(
            FeatureGroupName=feature_group_name,
            Record=[{"FeatureName": k, "ValueAsString": str(v)} for k, v in␣
 ↪record.items()]
        )
        success += 1
    except Exception as e:
        print(f" Failed to ingest record with primary_key={record.
 ↪get('primary_key')}: {e}")
        failed += 1

# --- Report Results ---
print(f"Successfully ingested: {success}")
print(f"Failed to ingest: {failed}")
```

```
Successfully ingested: 1306
Failed to ingest: 0
```

[76]:
```
import boto3
import pandas as pd

# Set your region and feature group name
boto_session = boto3.Session(region_name=region)
featurestore_runtime = boto_session.client("sagemaker-featurestore-runtime",␣
 ↪region_name=region)

# Neighborhoods to query
neighborhoods_to_query = ["Brooktree", "Fisherman's Wharf", "Los Osos"]
```

```
# Collect results
query_results = []

for key in neighborhoods_to_query:
    try:
        response = featurestore_runtime.get_record(
            FeatureGroupName=feature_group_name,
            RecordIdentifierValueAsString=key
        )
        if "Record" in response:
            record = {f["FeatureName"]: f["ValueAsString"] for f in␣
 ↪response["Record"]}
            query_results.append(record)
        else:
            print(f" No data found for: {key}")
    except featurestore_runtime.exceptions.ResourceNotFound:
        print(f" Record not found for: {key}")
    except Exception as e:
        print(f" Unexpected error for '{key}': {e}")

# Display as DataFrame
if query_results:
    df_results = pd.DataFrame(query_results)
    print(df_results)
else:
    print(" No matching records retrieved.")
```

```
         primary_key            event_time less_than_1h_ocean inland island  \
0          Brooktree  2025-05-24T22:57:55Z                  1      0      0
1  Fisherman's Wharf  2025-05-24T22:57:55Z                  0      0      0
2           Los Osos  2025-05-24T22:57:55Z                  0      0      0

  near_bay near_ocean median_house_value_capped median_house_age_grouped  \
0        0          0                  257400.0                        0
1        1          0                  500000.0                       50
2        0          1                  221612.5                       11

  total_households bedrooms_per_household
0             1438     0.3366418004804652
1              250                  1.268
2             4894     1.0502656313853698
```

[ ]: