Notebook price_change_dataset.csv walmart_dataset_with_competitor.csv •••

```
from faker import Faker
import numpy as np
import pandas as pd

# Initialize Faker
fake = Faker()

# Set random seed for reproducibility
np.random.seed(42)

# Generate synthetic dataset
num_samples = 1000

# Generate features
time_of_day = np.random.uniform(low=0, high=24, size=num_samples)  # Time of day (24-hour format)
historical_sales_data = np.random.randint(low=0, high=100, size=num_samples)  # Historical sales data
competitor_prices = np.random.uniform(low=0, high=100, size=num_samples)  # Competitor prices

# Generate target variable indicating price changes
# Assume a simple rule: if historical sales data > 50, price increases; otherwise, price decreases
price_changes = np.where(historical_sales_data > 50, 1, -1)

# Generate fake product descriptions
product_descriptions = [fake.word() for _ in range(num_samples)]

# Generate fake customer IDs and countries
customer_ids = [fake.random_number(digits=5) for _ in range(num_samples)]
countries = [fake.country() for _ in range(num_samples)]

# Create DataFrame
data = pd.DataFrame({
    'TimeOfDay': time_of_day,
```

```
    'HistoricalSalesData': historical_sales_data,
    'CompetitorPrices': competitor_prices,
    'PriceChanges': price_changes,
    'Description': product_descriptions,
    'CustomerID': customer_ids,
    'Country': countries
})

# Display first few rows of the dataset
print(data.head())

# Save dataset to CSV file
data.to_csv('price_change_dataset.csv', index=False)
```

```
   TimeOfDay  HistoricalSalesData  CompetitorPrices  PriceChanges Description  \
0   8.988963                   46         69.000489            -1        bill
1  22.817143                   11         20.093369            -1     quickly
2  17.567855                   61         53.582768             1        want
3  14.367804                   79          9.667645             1  investment
4   3.744447                   87         45.037094             1      animal

   CustomerID                  Country
0       55583                 Cambodia
1       77627                  Nigeria
2       72183                 Portugal
3       35878  Saint Pierre and Miquelon
4       75021             Guinea-Bissau
```

```
pip install faker
```

```
Collecting faker
  Downloading Faker-24.3.0-py3-none-any.whl (1.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.8/1.8 MB 6.0 MB/s eta 0:00:00
Requirement already satisfied: python-dateutil>=2.4 in /usr/local/lib/python3.10/dist-packages (from faker) (2.8
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.4->f
Installing collected packages: faker
Successfully installed faker-24.3.0
```

```
pip install faker
```

```
Collecting faker
  Downloading Faker-24.3.0-py3-none-any.whl (1.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.8/1.8 MB 7.3 MB/s eta 0:00:00
Requirement already satisfied: python-dateutil>=2.4 in /usr/local/lib/python3.10/dist-packages (from faker) (2.8
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.4->f
Installing collected packages: faker
Successfully installed faker-24.3.0
```

```
import pandas as pd
from faker import Faker
import random
import numpy as np
import datetime

# Initialize Faker
fake = Faker()

# Set random seed for reproducibility
np.random.seed(42)
```

```
np.random.seed(42)
random.seed(42)

# Generate synthetic dataset
num_samples = 10000

# Generate product information
products = [{'StockCode': fake.random_number(digits=6),
             'Description': fake.catch_phrase(),
             'Price': round(random.uniform(1, 100), 2),
             'CompetitorPrice': round(random.uniform(1, 100), 2)} for _ in range(num_samples)]

# Generate customer information
customers = [{'CustomerID': fake.random_number(digits=5),
              'Country': fake.country()} for _ in range(num_samples)]

# Generate sales transactions
transactions = []
for _ in range(num_samples):
    product = random.choice(products)
    customer = random.choice(customers)
    quantity = random.randint(1, 10)
    invoice_date = fake.date_time_between(start_date='-1y', end_date='now')
    transactions.append({
        'Invoice': fake.random_number(digits=6),
        'StockCode': product['StockCode'],
        'Description': product['Description'],
        'Quantity': quantity,
        'InvoiceDate': invoice_date,
        'Price': product['Price'],
        'CompetitorPrice': product['CompetitorPrice'],
        'CustomerID': customer['CustomerID'],
        'Country': customer['Country']
```

```
    })

# Create DataFrame
df = pd.DataFrame(transactions)

# Display first few rows of the dataset
print(df.head())

# Save dataset to CSV file
df.to_csv('walmart_dataset_with_competitor.csv', index=False)
```

```
   Invoice  StockCode                              Description  Quantity  \
0   575156     675586          Multi-lateral uniform encryption         6
1    34502     517565   Compatible mobile budgetary management         3
2   851549     838309               Robust cohesive parallelism        10
3   642199     246959  Object-based value-added superstructure         9
4   527502     974052                 Ergonomic modular product         3

                 InvoiceDate  Price  CompetitorPrice  CustomerID  \
0 2024-03-04 04:19:37.260968  74.35            50.47       15163
1 2023-05-31 10:23:17.251792  68.01            86.84       82055
2 2023-06-11 15:17:13.297873  50.27            97.80       77185
3 2024-02-22 05:14:45.420089  28.17            57.03       29577
4 2023-09-11 08:25:21.303141  23.38            40.14       41491

                          Country
0                            Peru
1                           Qatar
2                           Congo
3                     Isle of Man
4  Lao People's Democratic Republic
```

```
import theano
theano.config.blas__check_openmp = False
```

```
---------------------------------------------------------------------------
NoSectionError                            Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/theano/configparser.py in fetch_val_for_key(self, key, delete_key)
    237             try:
--> 238                 return self._theano_cfg.get(section, option)
    239             except InterpolationError:

                                15 frames
NoSectionError: No section: 'blas'

During handling of the above exception, another exception occurred:

KeyError                                  Traceback (most recent call last)
KeyError: 'blas__ldflags'

During handling of the above exception, another exception occurred:

ModuleNotFoundError                       Traceback (most recent call last)
ModuleNotFoundError: No module named 'mkl'

During handling of the above exception, another exception occurred:

RuntimeError                              Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/theano/link/c/cmodule.py in check_mkl_openmp()
   2590             )
   2591     except ImportError:
-> 2592         raise RuntimeError(
   2593             """
   2594 Could not import 'mkl'.  If you are using conda, update the numpy

RuntimeError:
Could not import 'mkl'.  If you are using conda, update the numpy
packages to the latest build otherwise, set MKL_THREADING_LAYER=GNU in
your environment for MKL 2018.
```

```
If you have MKL 2017 install and are not in a conda environment you
can set the Theano flag blas__check_openmp to False.  Be warned that if
you set this flag and don't set the appropriate environment or make
sure you have the right version you *will* get wrong results.
```

Next steps: Explain error

```
import numpy as np
import pandas as pd
import pymc3 as pm
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

# Load the dataset with competitor prices and other relevant features
df = pd.read_csv('walmart_dataset_with_competitor.csv')

# Define features and target variable
X = df[['CompetitorPrice', 'Quantity', 'HistoricalSalesData']]  # Features
y = df['PriceChanges']  # Target variable (0 for no change, 1 for increase, -1 for decrease)

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Standardize features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Bayesian logistic regression model
with pm.Model() as bayesian_model:
    # Priors
    intercept = pm.Normal('intercept', mu=0, sd=10)
    beta = pm.Normal('beta', mu=0, sd=10, shape=X_train_scaled.shape[1])

    # Linear combination of features
    theta = intercept + pm.math.dot(X_train_scaled, beta)

    # Sigmoid function to convert linear combination to probability
    p = pm.Deterministic('p', pm.math.sigmoid(theta))
```

```
    # Likelihood (binary classification)
    y_obs = pm.Bernoulli('y_obs', p=p, observed=y_train)

    # Fit the model
    trace = pm.sample(1000, tune=1000, cores=1)  # Adjust parameters as needed

# Summarize the results
pm.summary(trace)

# Make predictions on the test set
with bayesian_model:
    ppc = pm.sample_posterior_predictive(trace, samples=500, progressbar=True)

# Extract predicted probabilities
predicted_probs = ppc['y_obs'].mean(axis=0)

# Adjusted probabilities (optional, if needed)
adjusted_probs = 1 - predicted_probs  # Adjusted probabilities for price decrease

# Display adjusted probabilities
print(adjusted_probs)
```

```
---------------------------------------------------------------------------
NoSectionError                            Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/theano/configparser.py in fetch_val_for_key(self, key, delete_key)
    237             try:
--> 238                 return self._theano_cfg.get(section, option)
    239             except InterpolationError:

                                16 frames

NoSectionError: No section: 'blas'

During handling of the above exception, another exception occurred:

KeyError                                  Traceback (most recent call last)
KeyError: 'blas__ldflags'

During handling of the above exception, another exception occurred:

ModuleNotFoundError                       Traceback (most recent call last)
ModuleNotFoundError: No module named 'mkl'

During handling of the above exception, another exception occurred:

RuntimeError                              Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/theano/link/c/cmodule.py in check_mkl_openmp()
   2590             )
   2591     except ImportError:
-> 2592         raise RuntimeError(
   2593             """
   2594 Could not import 'mkl'.  If you are using conda, update the numpy

RuntimeError:
Could not import 'mkl'.  If you are using conda, update the numpy
packages to the latest build otherwise, set MKL_THREADING_LAYER=GNU in
your environment for MKL 2018.
```

```
If you have MKL 2017 install and are not in a conda environment you
can set the Theano flag blas__check_openmp to False.  Be warned that if
you set this flag and don't set the appropriate environment or make
sure you have the right version you *will* get wrong results.
```

Next steps: Explain error

```
pip install pymc3
```

```
Requirement already satisfied: pymc3 in /usr/local/lib/python3.10/dist-packages (3.11.5)
Requirement already satisfied: arviz>=0.11.0 in /usr/local/lib/python3.10/dist-packages (from pymc3) (0.12.1)
Requirement already satisfied: cachetools>=4.2.1 in /usr/local/lib/python3.10/dist-packages (from pymc3) (5.3.3)
Requirement already satisfied: deprecat in /usr/local/lib/python3.10/dist-packages (from pymc3) (2.1.1)
Requirement already satisfied: dill in /usr/local/lib/python3.10/dist-packages (from pymc3) (0.3.8)
Requirement already satisfied: fastprogress>=0.2.0 in /usr/local/lib/python3.10/dist-packages (from pymc3) (1.0.
Requirement already satisfied: numpy<1.22.2,>=1.15.0 in /usr/local/lib/python3.10/dist-packages (from pymc3) (1.
Requirement already satisfied: pandas>=0.24.0 in /usr/local/lib/python3.10/dist-packages (from pymc3) (1.5.3)
Requirement already satisfied: patsy>=0.5.1 in /usr/local/lib/python3.10/dist-packages (from pymc3) (0.5.6)
Requirement already satisfied: scipy<1.8.0,>=1.7.3 in /usr/local/lib/python3.10/dist-packages (from pymc3) (1.7.
Requirement already satisfied: semver>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from pymc3) (3.0.2)
Requirement already satisfied: theano-pymc==1.1.2 in /usr/local/lib/python3.10/dist-packages (from pymc3) (1.1.2
Requirement already satisfied: typing-extensions>=3.7.4 in /usr/local/lib/python3.10/dist-packages (from pymc3)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from theano-pymc==1.1.2->pym
Requirement already satisfied: setuptools>=38.4 in /usr/local/lib/python3.10/dist-packages (from arviz>=0.11.0->
Requirement already satisfied: matplotlib>=3.0 in /usr/local/lib/python3.10/dist-packages (from arviz>=0.11.0->p
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from arviz>=0.11.0->pymc3)
Requirement already satisfied: xarray>=0.16.1 in /usr/local/lib/python3.10/dist-packages (from arviz>=0.11.0->py
Requirement already satisfied: netcdf4 in /usr/local/lib/python3.10/dist-packages (from arviz>=0.11.0->pymc3) (1
Requirement already satisfied: xarray-einstats>=0.2 in /usr/local/lib/python3.10/dist-packages (from arviz>=0.11
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.24.0->pym
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from patsy>=0.5.1->pymc3) (1.16.0
Requirement already satisfied: wrapt<2,>=1.10 in /usr/local/lib/python3.10/dist-packages (from deprecat->pymc3)
```

```
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0->ar
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0->a
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0
Requirement already satisfied: cftime in /usr/local/lib/python3.10/dist-packages (from netcdf4->arviz>=0.11.0->p
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from netcdf4->arviz>=0.11.0->
```

```
pip install --upgrade tensorflow
```

```
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.10/dist-packages (from astunparse
Requirement already satisfied: rich in /usr/local/lib/python3.10/dist-packages (from keras>=3.0.0->tensorflow
Collecting namex (from keras>=3.0.0->tensorflow)
  Downloading namex-0.0.7-py3-none-any.whl (5.8 kB)
Collecting optree (from keras>=3.0.0->tensorflow)
  Downloading optree-0.10.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (286 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 286.8/286.8 kB 24.8 MB/s eta 0:00:00
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requ
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.2
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.10/dist-packages (from werkzeug>=1
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.10/dist-packages (from rich->k
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from rich-
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-packages (from markdown-it-py>=2.
Installing collected packages: namex, optree, numpy, tensorboard, ml-dtypes, h5py, keras, tensorflow
  Attempting uninstall: numpy
    Found existing installation: numpy 1.22.1
    Uninstalling numpy-1.22.1:
      Successfully uninstalled numpy-1.22.1
  Attempting uninstall: tensorboard
    Found existing installation: tensorboard 2.15.2
    Uninstalling tensorboard-2.15.2:
      Successfully uninstalled tensorboard-2.15.2
  Attempting uninstall: ml-dtypes
    Found existing installation: ml-dtypes 0.2.0
    Uninstalling ml-dtypes-0.2.0:
      Successfully uninstalled ml-dtypes-0.2.0
  Attempting uninstall: h5py
    Found existing installation: h5py 3.9.0
    Uninstalling h5py-3.9.0:
      Successfully uninstalled h5py-3.9.0
  Attempting uninstall: keras
```

```
    Found existing installation: keras 2.15.0
    Uninstalling keras-2.15.0:
      Successfully uninstalled keras-2.15.0
  Attempting uninstall: tensorflow
    Found existing installation: tensorflow 2.15.0
    Uninstalling tensorflow-2.15.0:
      Successfully uninstalled tensorflow-2.15.0
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. Th
jax 0.4.23 requires scipy>=1.9, but you have scipy 1.7.3 which is incompatible.
jaxlib 0.4.23+cuda12.cudnn89 requires scipy>=1.9, but you have scipy 1.7.3 which is incompatible.
pymc 5.10.4 requires arviz>=0.13.0, but you have arviz 0.12.1 which is incompatible.
pymc3 3.11.5 requires numpy<1.22.2,>=1.15.0, but you have numpy 1.26.4 which is incompatible.
scipy 1.7.3 requires numpy<1.23.0,>=1.16.5, but you have numpy 1.26.4 which is incompatible.
tf-keras 2.15.1 requires tensorflow<2.16,>=2.15, but you have tensorflow 2.16.1 which is incompatible.
Successfully installed h5py-3.10.0 keras-3.1.1 ml-dtypes-0.3.2 namex-0.0.7 numpy-1.26.4 optree-0.10.0 tensorb
WARNING: The following packages were previously imported in this runtime:
  [ml_dtypes]
You must restart the runtime in order to use newly installed versions.
```

RESTART SESSION

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import classification_report
from tensorflow import keras
from tensorflow.keras import layers

# Load the dataset with competitor prices and other relevant features
df = pd.read_csv('walmart_dataset_with_competitor.csv')

# Convert 'InvoiceDate' column to timestamp
df['InvoiceDate'] = pd.to_datetime(df['InvoiceDate'])

# Extract hour of the day from 'InvoiceDate' and create a new feature
df['HourOfDay'] = df['InvoiceDate'].dt.hour

# Drop 'InvoiceDate' column
df.drop(columns=['InvoiceDate'], inplace=True)

# Now continue with feature scaling and model training


# Define features and target variable
X = df[['CompetitorPrice', 'Quantity', 'HourOfDay']]  # Features
y = df['Price']  # Target variable (0 for no change, 1 for increase, -1 for decrease)

# Standardize features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Convert target variable to one-hot encoded format
```

```
y_encoded = pd.get_dummies(y)

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y_encoded, test_size=0.2, random_state=42)

# Build the neural network model
model = keras.Sequential([
    layers.Dense(64, activation='relu', input_shape=(X_train.shape[1],)),
    layers.Dense(32, activation='relu'),
    layers.Dense(3, activation='softmax')  # Output layer with 3 units for 3 classes (increase, decrease, no change)
])

# Compile the model
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# Train the model
history = model.fit(X_train, y_train, epochs=10, batch_size=32, validation_split=0.2)

# Evaluate the model
loss, accuracy = model.evaluate(X_test, y_test)
print("Test Accuracy:", accuracy)

# Make predictions
y_pred = model.predict(X_test)
y_pred_classes = np.argmax(y_pred, axis=1)
y_test_classes = np.argmax(np.array(y_test), axis=1)

# Print classification report
print(classification_report(y_test_classes, y_pred_classes))
```

```
Epoch 1/10
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-10-83f8fb3e97c0> in <cell line: 51>()
     49
     50 # Train the model
---> 51 history = model.fit(X_train, y_train, epochs=10, batch_size=32, validation_split=0.2)
     52
     53 # Evaluate the model

                                 1 frames
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py in tf__train_function(iterator)
     13                 try:
     14                     do_return = True
---> 15                     retval_ = ag__.converted_call(ag__.ld(step_function), (ag__.ld(self),
ag__.ld(iterator)), None, fscope)
     16                 except:
     17                     do_return = False

ValueError: in user code:

    File "/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py", line 1401, in train_function
*
        return step_function(self, iterator)
    File "/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py", line 1384, in step_function
**
        outputs = model.distribute_strategy.run(run_step, args=(data,))
    File "/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py", line 1373, in run_step  **
        outputs = model.train_step(data)
    File "/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py", line 1151, in train_step
        loss = self.compute_loss(x, y, y_pred, sample_weight)
    File "/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py", line 1209, in compute_loss
        return self.compiled_loss(
    File "/usr/local/lib/python3.10/dist-packages/keras/src/engine/compile_utils.py", line 277, in __call__
        loss_value = loss_obj(y_t, y_p, sample_weight=sw)
```