

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# Load your preprocessed CSV dataset
# Replace 'your_dataset.csv' with the actual filename
dataset_path = '/content/drive/MyDrive/dataset/Book2.csv'
df = pd.read_csv(dataset_path)

# Assuming your dataset has features in columns and labels in the 'Outcome'
label_column = 'Outcome'
X = df.drop(label_column, axis=1)
y = df[label_column]

# 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)

# Train a RandomForestClassifier as an example model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Make predictions on the training set
y_train_pred = model.predict(X_train)

# Make predictions on the test set
y_test_pred = model.predict(X_test)

# Evaluate training accuracy
train_accuracy = accuracy_score(y_train, y_train_pred)
print(f'Training Accuracy: {train_accuracy:.2f}')

# Evaluate testing accuracy
test_accuracy = accuracy_score(y_test, y_test_pred)
print(f'Testing Accuracy: {test_accuracy:.2f}')

# Display classification report for the test set
print('Classification Report (Test Set):')
print(classification_report(y_test, y_test_pred))

# Plot confusion matrix for the test set
cm = confusion_matrix(y_test, y_test_pred)
# Ensure cm is a 2x2 matrix
if cm.shape == (1, 1):
    cm = np.array([[0, 0], [0, 0]])

cm_df = pd.DataFrame(cm, index=['Actual Negative', 'Actual Positive'], columns=['Predicted Negative', 'Predicted Positive'])

# Plot confusion matrix
plt.figure(figsize=(12, 4))
sns.heatmap(cm_df, annot=True, fmt='d', cmap='Blues', annot_kws={'size': 16})
plt.title('Confusion Matrix (Test Set)')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')

# Plot training and testing accuracy
plt.figure(figsize=(12, 4))
accuracy_values = [train_accuracy, test_accuracy]
labels = ['Training Accuracy', 'Testing Accuracy']
sns.barplot(x=labels, y=accuracy_values)
plt.ylim([0, 1])
plt.title('Training and Testing Accuracy')
```

```

plt.tight_layout()
plt.show()
from HDFunc-tionsLib import *
    from parametersSetup import *
    from scipy import interpolate

#####
#SETUPS
GeneralParams.plottingON=0
GeneralParams.PersGenApproach='personalized'
datasetPreparationType='MoreNonSeizure_Fact5' # 'MoreNonSeizure_Fact5'
torch.cuda.set_device(HDParams.CUDAdevice)
HDParams.D=10000

optType = 'F1DEgmean' # 'simpleAcc', 'F1DEgmean'
#MULTI CLASS PARAMS
numSteps = 10
groupingThresh = 0.95
subClassReductApproachType = 'clustering' # 'removing', 'clustering'
perfDropThr=0.03 #0.01, 0.02, 0.03
#ITERATIVE LEARNING
IterType='AddAndSubtract' # 'AddAndSubtract', 'AddOnly'
IterFact=1
IterImprovThresh=0.01 #if in three consecutive runs not bigger improvement
savingStepData=1 #whether to save improvements per each iteration

#DATASET
Dataset='01_CHBMIT'
GeneralParams.patients = ['01', '02', '03', '04', '05', '06', '07', '08', '09',
GeneralParams.patients = ['01', '02', '03']

# DEFINING INPUT/OUTPUT FOLDERS
folderIn = '01_datasetProcessed_'+datasetPreparationType+'/'
folderOut0= '03_predictions_' +datasetPreparationType
createFolderIfNotExists(folderOut0)
# folderOut0=folderOut0 + '/' + str(GeneralParams.PersGenApproach)+'/'
# createFolderIfNotExists(folderOut0)
folderFeaturesOut='02_features_'+datasetPreparationType
createFolderIfNotExists(folderFeaturesOut)
# folderFeaturesOut0=folderFeaturesOut0 + '/' + str(GeneralParams.PersGenApproach)+'/'
# createFolderIfNotExists(folderFeaturesOut0)

# FEATURES USED - STANDARD ML FEATURES - 45 FEAT
HDParams.HDApproachON=1
SegSymbParams.symbolType = 'StandardMLFeatures'
HDParams.numFeat=45
SegSymbParams.numSegLevels=20 #num discretized windows for feature value
SegSymbParams.segLenSec = 8 #8 # length of EEG segments in sec
SegSymbParams.slidWindStepSec = 1 #1 # step of sliding window to extra
HDParams.vectorTypeLevel = 'scaleNoRand1' # 'random', 'sandwich', 'scaleNoRand1'
HDParams.vectorTypeCh = 'random' # 'random', 'sandwich', 'scaleNoRand1'
HDParams.vectorTypeFeat='random'
HDParams.roundingTypeForHDVectors='inSteps' # 'inSteps', 'onlyOne', 'noRounding'
HDParams.bindingFeatures='FeatxVal' # 'FeatxVal', 'ChxFeatxVal', 'FeatxVal'
HDParams.D=10000
#HDParams.IterativeRelearning='on'

#calculating various parameters
seizureStableLenToTestIdx = int(GeneralParams.seizureStableLenToTest / 100)
seizureStablePercToTest = GeneralParams.seizureStablePercToTest / 100
distanceBetweenSeizuresIdx = int(GeneralParams.distanceBetween2Seizures / 100)
numLabelsPerHour = 60 * 60 / SegSymbParams.slidWindStepSec
toleranceFP_bef = int(GeneralParams.toleranceFP_befSeiz / SegSymbParams.slidWindStepSec)
toleranceFP_aft = int(GeneralParams.toleranceFP_aftSeiz / SegSymbParams.slidWindStepSec)

```

Please follow our [blog](#) to see more information about new features, tips and tricks, and featured notebooks such as [Analyzing a Bank Failure with Colab](#).

## 2024-01-08

- Avoid nested scrollbars for large outputs by using `google.colab.output.no_vertical_scroll` [Example notebook](#)
- Fix [bug](#) where downloading models from Hugging Face could freeze
- Python package upgrades
  - `huggingface-hub` 0.19.4 -> 0.20.2
  - `bigframes` 0.17.0 -> 0.18.0

## 2023-12-18

- Expanded access to AI coding has arrived in Colab across 175 locales for all tiers of Colab users
- Improvements to display of ML-based inline completions (for eligible Pro/Pro+ users)
- Started a series of [notebooks](#) highlighting Gemini API capabilities
- Enable `⌘/Ctrl+L` to select the full line in an editor
- Fixed [bug](#) where we weren't correctly formatting output from multiple execution results
- Python package upgrades
  - `CUDA` 11.8 to `CUDA` 12.2
  - `tensorflow` 2.14.0 -> 2.15.0
  - `tensorboard` 2.14.0 -> 2.15.0
  - `keras` 2.14.0 -> 2.15.0
  - `Nvidia drivers` 525.105.17 -> 535.104.05
  - `tensorflow-gcs-config` 2.14.0 -> 2.15.0
  - `bigframes` 0.13.0 -> 0.17.0
  - `geemap` 0.28.2 -> 0.29.6
  - `pyarrow` 9.0.0 -> 10.0.1
  - `google-generativeai` 0.2.2 -> 0.3.1
  - `jax` 0.4.20 -> 0.4.23
  - `jaxlib` 0.4.20 -> 0.4.23
- Python package inclusions
  - `kagglehub` 0.1.4
  - `google-cloud-aiplatform` 1.38.1

## 2023-11-27

- Removed warning when calling `await` to make it render as code
- Added "Run selection" to the cell context menu
- Added highlighting for the `%python` cell magic
- Launched AI coding features for Pro/Pro+ users in more locales
- Python package upgrades
  - `bigframes` 0.12.0 -> 0.13.0
- Python package inclusions
  - `transformers` 4.35.2
  - `google-generativeai` 0.2.2

## 2023-11-08

- Launched Secrets, for safe storage of private keys on Colab ([tweet](#))
- Fixed issue where TensorBoard would not load ([#3990](#))
- Python package upgrades
  - `lightgbm` 4.0.0 -> 4.1.0

```

# #saving parameters to folder name
# folderOutName = SegSymbParams.symbolType + '_' + str(HDParams.numFeat)
# HDParams.numFeat) + '_' + HDParams.bindingFeatures + '_FEATvec' + HD
# folderOutNameFeat = SegSymbParams.symbolType + '_' + str(HDParams.num
# folderOutName = folderOutName + '_' + str(SegSymbParams.segLenSec) +
# SegSymbParams.slidWindStepSec) + 's' + '_' + HDParams.similarityType
# folderOutNameFeat = folderOutNameFeat + '_' + str(SegSymbParams.segLen
# folderOutName = folderOutName + '_MultiClassPaper'
# folderFeaturesOut = folderFeaturesOut0 + folderOutNameFeat
# folderOut_ML = folderOut0 + folderOutName
# createFolderIfNotExists(folderOut_ML)

#final folder to store data to
folderOut_ML = folderOut0 + '/' + optType + '_' + str(perfDropThr) + '_' + str(
createFolderIfNotExists(folderOut_ML)
print('FOLDER OUT:', folderOut_ML)
print('FOLDER OUT FEATURES:', folderFeaturesOut)
folderOutPredictionsPlot = folderOut_ML + '/Plots_predictions'
createFolderIfNotExists(folderOutPredictionsPlot)

#####
## CALCULATING FEATURES FOR EACH FILE
numFiles = len(np.sort(glob.glob(folderFeaturesOut + '/*chb' + '*.csv'
if (numFiles==0):
print('EXTRACTING FEATURES!!!')
func_calculateFeaturesForInputFiles(SigInfoParams, SegSymbParams, Gene

#####
## TRAINING
AllSubjRes_train = np.zeros((len(GeneralParams.patients), 33, 2))
AllSubjRes_test = np.zeros((len(GeneralParams.patients), 33, 2))
AllSubjResMulti_train = np.zeros((len(GeneralParams.patients), 33, 2))
AllSubjResMulti_test = np.zeros((len(GeneralParams.patients), 33, 2))
AllSubjResMultiRed_train = np.zeros((len(GeneralParams.patients), 33,
AllSubjResMultiRed_test = np.zeros((len(GeneralParams.patients), 33, 2)
AllSubjResMultiClust_train = np.zeros((len(GeneralParams.patients), 33
AllSubjResMultiClust_test = np.zeros((len(GeneralParams.patients), 33,
AllSubj_OptimalResultsReduced_train= np.zeros((len(GeneralParams.patien
AllSubj_OptimalResultsReduced_test= np.zeros((len(GeneralParams.patien
AllSubj_OptimalResultsClustered_train= np.zeros((len(GeneralParams.pat
AllSubj_OptimalResultsClustered_test= np.zeros((len(GeneralParams.patien

# go through each subject for personalized approach
for patIdx, pat in enumerate(GeneralParams.patients):
numFiles = len(np.sort(glob.glob(folderFeaturesOut + '/*chb' + pat + '
print('-- Patient:', pat, 'NumSeizures:', numFiles)

# go through leave-one-out cross-validations for this subject
AllRes_train=np.zeros((numFiles,33))
AllRes_test = np.zeros((numFiles, 33))
AllResMulti_train = np.zeros((numFiles, 33))
AllResMulti_test = np.zeros((numFiles, 33))
AllResMultiRed_train = np.zeros((numFiles, 33))
AllResMultiRed_test = np.zeros((numFiles, 33))
AllResMultiClust_train = np.zeros((numFiles, 33))
AllResMultiClust_test = np.zeros((numFiles, 33))
OptimalValues_train_red= np.zeros((numFiles, 34))
OptimalValues_test_red = np.zeros((numFiles, 34))
OptimalValues_train_clust= np.zeros((numFiles, 34))
OptimalValues_test_clust = np.zeros((numFiles, 34))
OptimalValuesClustered_train= np.zeros((numFiles, 34))
OptimalValuesClustered_test = np.zeros((numFiles, 34))
for cv in range(numFiles):
# creates list of files to train and test on
filesToTrainOn = []

```

- bigframes 0.10.0 -> 0.12.0
- bokeh 3.2.2 -> 3.3.0
- duckdb 0.8.1 -> 0.9.1
- numba 0.56.4 -> 0.58.1
- tweepy 4.13.0 -> 4.14.0
- jax 0.4.16 -> 0.4.20
- jaxlib 0.4.16 -> 0.4.20

## 2023-10-23

- Updated the **Open notebook** dialog for better usability and support for smaller screen sizes
- Added smart paste support for data from Google Sheets for R notebooks
- Enabled showing release notes in a tab
- Launched AI coding features for Pro/Pro+ users in Australia AU Canada CA India IN and Japan JP ([tweet](#))
- Python package upgrades
  - earthengine-api 0.1.357 -> 0.1.375
  - flax 0.7.2 -> 0.7.4
  - geemap 0.27.4 -> 0.28.2
  - jax 0.4.14 -> 0.4.16
  - jaxlib 0.4.14 -> 0.4.16
  - keras 2.13.1 -> 2.14.0
  - tensorboard 2.13.0 -> 2.14.1
  - tensorflow 2.13.0 -> 2.14.0
  - tensorflow-gcs-config 2.13.0 -> 2.14.0
  - tensorflow-hub 0.14.0 -> 0.15.0
  - tensorflow-probability 0.20.1 -> 0.22.0
  - torch 2.0.1 -> 2.1.0
  - torchaudio 2.0.2 -> 2.1.0
  - torchtext 0.15.2 -> 0.16.0
  - torchvision 0.15.2 -> 0.16.0
  - xgboost 1.7.6 -> 2.0.0
- Python package inclusions
  - bigframes 0.10.0
  - malloy 2023.1056

## 2023-09-22

- Added the ability to scope an AI generated suggestion to a specific Pandas dataframe ([tweet](#))
- Added Colab link previews to Docs ([tweet](#))
- Added smart paste support for data from Google Sheets
- Increased font size of dropdowns in interactive forms
- Improved rendering of the notebook when printing
- Python package upgrades
  - tensorflow 2.12.0 -> 2.13.0
  - tensorboard 2.12.3 -> 2.13.0
  - keras 2.12.0 -> 2.13.1
  - tensorflow-gcs-config 2.12.0 -> 2.13.
  - scipy 1.10.1 -> 1.11.2
  - cython 0.29.6 -> 3.0.2
- Python package inclusions
  - geemap 0.26.0

## 2023-08-18

- Added "Change runtime type" to the menu in the connection button
- Improved auto-reconnection to an already running notebook ([#3764](#))
- Increased the specs of our highmem machines for Pro users
- Fixed add-apt-repository command on Ubuntu 22.04 runtime ([#3867](#))

```

for fIndx, fileName in enumerate(np.sort(glob.glob(folder+featuresOut +
if (fIndx != cv):
filesToTrainOn.append(fileName)
else:
filesToTestOn = list(fileName.split(" "))
pom, fileName1 = os.path.split(filesToTestOn[0])
fileName2 = os.path.splitext(fileName1)[0]

# concatenating data from more files
(dataTrain, label_train)= concatenateDataFromFiles(filesToTrainOn)
(dataTest, label_test) = concatenateDataFromFiles(filesToTestOn)

# normalizing data and discretizing
(data_train_Norm, data_test_Norm, data_train_Discr, data_test_Discr)=
data_train_Discr=data_train_Discr.astype(int)
data_test_Discr = data_test_Discr.astype(int)

# INITIALIZING HD VECTORS
model = HD_classifier_GeneralAndNoCh(SigInfoParams, SegSymbParams, HDP
#model = HD_classifier_GeneralWithChCombinations(SigInfoParams, SegSym

#####
#STANDARD SINGLE PASS 2 CLASS LEARNING
#learn on trainin set
(ModelVectors, ModelVectorsNorm, numAddedVecPerClass, numLabels) = tra
#measure performance on test set
(AllRes_train[cv,:], AllRes_test[cv,:], predLabelsTrain_2class, pred-L
ModelVectorsNorm, HDPParams, GeneralParams, SegSymbParams)
print('2 CLASS acc_train: ', AllRes_train[cv,2], 'acc_test: ', AllRes_

#####
#MULTICLASS LEARNING
# learn on trainin set
(ModelVectorsMulti_Seiz, ModelVectorsMultiNorm_Seiz, ModelVectorsMulti
numAddedVecPerClassMulti_Seiz, numAddedVecPerClassMulti_NonSeiz) =trai
#measure performance on test set
(AllResMulti_train[cv,:], AllResMulti_test[cv,:], predLabelsTrain_Mult
ModelVectorsMultiNorm_Seiz, ModelVectorsMultiNorm_NonSeiz, HDPParams, G
print('MULTI CLASS acc_train: ', AllResMulti_train[cv,2], 'acc_test: '

#####
#ANALYSE REMOVING LESS CROWDED SUBCLASSES
#REMOVING
subClassReductApproachType = 'removing'
(OptimalValues_train_red[cv,:], OptimalValues_test_red[cv,:], ModelVec
numAddedVecPer-Class_NonSeiz_red)=reduceNumSubclasses_removingApproach
ModelVectorsMultiNorm_Seiz, ModelVectorsMultiNorm_NonSeiz, numAddedVec
numSteps, optType, perfDropThr, GeneralParams, SegSymbParams, folderOu
# performance on training and test dataset
(AllResMultiRed_train[cv, :], AllResMultiRed_test[cv, :], predLabel-sT
ModelVectorsMultiNorm_Seiz_red, ModelVectorsMultiNorm_NonSeiz_red, HDP
print('MULTI CLASS REDUCED acc_train: ', AllResMultiRed_train[cv, 2],

#CLUSTERING
subClassReductApproachType = 'clustering'
(OptimalValues_train_clust[cv,:], OptimalValues_test_clust[cv,:], Mode
numAddedVecPerClass_NonSeiz_clust) = reduceNumSub-classes_clusteringAp
ModelVectorsMultiNorm_Seiz, ModelVectorsMultiNorm_NonSeiz, numAddedVec
numSteps, optType, perfDropThr, groupingThresh, GeneralParams, SegSymb
# performance on training and test dataset
(AllResMultiClust_train[cv, :], AllResMultiClust_test[cv, :], predLabe
ModelVectorsMultiNorm_Seiz_clust, ModelVectorsMultiNorm_NonSeiz_clust,
print('MULTI CLASS CLUSTER acc_train: ', AllResMultiClust_train[cv, 2]

#SAVE PREDICTIONS FOR ALL APPROACHES
dataToSave_train=np.vstack((label_train, predLabelsTrain_2class, predL

```

- Python package upgrades
  - bokeh 2.4.3 -> 3.2.2
  - cmake 3.25.2 -> 3.27.2
  - cryptography 3.4.8 -> 41.0.3
  - dask 2022.12.1 -> 2023.8.0
  - distributed 2022.12.1 -> 2023.8.0
  - earthengine-api 0.1.358 -> 0.1.364
  - flax 0.7.0 -> 0.7.2
  - ipython-sql 0.4.0 -> 0.5.0
  - jax 0.4.13 -> 0.4.14
  - jaxlib 0.4.13 -> 0.4.14
  - lightgbm 3.3.5 -> 4.0.0
  - mkl 2019.0 -> 2023.2.0
  - notebook 6.4.8 -> 6.5.5
  - numpy 1.22.4 -> 1.23.5
  - opencv-python 4.7.0.72 -> 4.8.0.76
  - pillow 8.4.0 -> 9.4.0
  - plotly 5.13.1 -> 5.15.0
  - prettytable 0.7.2 -> 3.8.0
  - pytensor 2.10.1 -> 2.14.2
  - spacy 3.5.4 -> 3.6.1
  - statsmodels 0.13.5 -> 0.14.0
  - xarray 2022.12.0 -> 2023.7.0
- Python package inclusions
  - PyDrive2 1.6.3

## 2023-07-21

- Launched auto-plotting for dataframes, available using the chart button that shows up alongside datatables ([post](#))



- Added a menu to the table of contents to support running a section or collapsing/expanding sections ([post](#))

- Added an option to automatically run the first cell or section, available under Edit -> Notebook settings ([post](#))

- Launched Pro/Pro+ to Algeria, Argentina, Chile, Ecuador, Egypt, Ghana, Kenya, Malaysia, Nepal, Nigeria, Peru, Rwanda, Saudi Arabia, South Africa, Sri Lanka, Tunisia, and Ukraine ([tweet](#))
- Added a command, "Toggle tab moves focus" for toggling tab trapping in the editor (Tools -> Command palette, "Toggle tab moves focus")
- Fixed issue where files.upload() was sometimes returning an incorrect filename ([#1550](#))

```

outputName = folderOut_ML + '/' + fileName2 + '_AllApproaches_TrainPre
np.savetxt(outputName, dataToSave_train, delimiter=",")
dataToSave_test=np.vstack((label_test, predLabelsTest_2class, pred-Lab
outputName = folderOut_ML + '/' + fileName2 + '_AllApproaches_TestPred
np.savetxt(outputName, dataToSave_test, delimiter=",")
#plot predictions for test
approachNames = ['2C', 'MC', 'MCred', 'MCclust']
approachIndx = [1, 2, 4, 6]
func_plotRawSignalAndPredictionsOfDiffApproaches_thisFile(fileName2, d

#SAVE MODEL VECTORS
#standard learning
outputName = folderOut_ML + '/' + fileName2 + '_StandardLearning_Model'
np.savetxt(outputName, ModelVectorsNorm.transpose(), delimiter=",")
outputName = folderOut_ML + '/' + fileName2 + '_StandardLearning_Model'
np.savetxt(outputName, ModelVectors.transpose(), delimiter=",")
outputName = folderOut_ML + '/' + fileName2 + '_StandardLearning_Added'
np.savetxt(outputName, numAddedVecPerClass, delimiter=",")
#multiclass
numSubClass_Seiz= len(numAddedVecPerClassMulti_Seiz)
numSubClass_NonSeiz = len(numAddedVecPerClassMulti_NonSeiz)
maxLen=np.max([numSubClass_Seiz,numSubClass_NonSeiz ] )
dataToSave=np.ones((2,maxLen))*np.nan
data-ToSave[0,0:numSubClass_Seiz]=numAddedVecPerClassMulti_Seiz[0:numS
dataToSave[1, 0:numSubClass_NonSeiz] = numAddedVecPerClassMul-ti_NonSe
outputName = folderOut_ML + '/' + fileName2 + '_MultiClass_AddedToEach
np.savetxt(outputName, dataToSave.transpose(), delimiter=",")
outputName = folderOut_ML + '/' + fileName2 + '_MultiClass_SeizModelVe
np.savetxt(outputName, ModelVectorsMultiNorm_Seiz.transpose(), delimit
outputName = folderOut_ML + '/' + fileName2 + '_MultiClass_NonSeizMode
np.savetxt(outputName, ModelVectorsMultiNorm_NonSeiz.transpose(), deli
#multiclass reduced
numSubClassMultiRed_Seiz=len(ModelVectorsMulti_Seiz_red[:,0])
numSubClassMultiRed_NonSeiz = len(ModelVectorsMulti_NonSeiz_red[:, 0])
dataToSave=np.ones((2,maxLen))*np.nan
data-To-Save[0,0:numSubClassMultiRed_Seiz]=numAddedVecPerClass_Seiz_re
dataToSave[1, 0:numSubClassMultiRed_NonSeiz] = numAddedVecPer-Class_No
outputName = folderOut_ML + '/' + fileName2 + '_MultiClassReduced_Adde
np.savetxt(outputName, dataToSave.transpose(), delimiter=",")
outputName = folderOut_ML + '/' + fileName2 + '_MultiClassReduced_Seiz
np.savetxt(outputName, ModelVectorsMultiNorm_Seiz_red.transpose(), del
outputName = folderOut_ML + '/' + fileName2 + '_MultiClassReduced_NonS
np.savetxt(outputName, ModelVectorsMultiNorm_NonSeiz_red.transpose(),
#multiclass clustered
numSubClassMultiClust_Seiz=len(ModelVectorsMulti_Seiz_clust[:,0])
numSubClassMultiClust_NonSeiz = len(ModelVectorsMulti_NonSeiz_clust[:,
dataToSave=np.ones((2,maxLen))*np.nan
data-To-Save[0,0:numSubClassMultiClust_Seiz]=numAddedVecPerClass_Seiz_
dataToSave[1, 0:numSubClassMultiClust_NonSeiz] = numAddedVecPer-Class_
outputName = folderOut_ML + '/' + fileName2 + '_MultiClassClustered_Ad
np.savetxt(outputName, dataToSave.transpose(), delimiter=",")
outputName = folderOut_ML + '/' + fileName2 + '_MultiClassClustered_Se
np.savetxt(outputName, ModelVectorsMultiNorm_Seiz_clust.transpose(), d
outputName = folderOut_ML + '/' + fileName2 + '_MultiClassClustered_No
np.savetxt(outputName, ModelVectorsMultiNorm_NonSeiz_clust.transpose())

#saving performance per subj
outputName = folderOut_ML + '/Subj' + pat + '_MultiClassLearning_Optim
np.savetxt(outputName, OptimalValues_train_red, delimiter=",")
outputName = folderOut_ML + '/Subj' + pat + '_MultiClassLearning_Optim
np.savetxt(outputName, OptimalValues_test_red, delimiter=",")
outputName = folderOut_ML + '/Subj' + pat + '_MultiClassLearning_Optim
np.savetxt(outputName, OptimalValues_train_clust, delimiter=",")
outputName = folderOut_ML + '/Subj' + pat + '_MultiClassLearning_Optim
np.savetxt(outputName, OptimalValues_test_clust, delimiter=",")

outputName = folderOut_ML + '/Subj' + pat + '_StandardLearning_TrainPre

```

- Fixed f-string syntax highlighting bug ([#3802](#))
- Disabled ambiguous characters highlighting for commonly used LaTeX characters ([#3648](#))
- Upgraded Ubuntu from 20.04 LTS to [22.04 LTS](#)
- Updated the Colab Marketplace VM image
- Python package upgrades:
  - autograd 1.6.1 -> 1.6.2
  - drivefs 76.0 -> 77.0
  - flax 0.6.11 -> 0.7.0
  - earthengine-api 0.1.357 -> 0.1.358
  - GDAL 3.3.2->3.4.3
  - google-cloud-bigquery-storage 2.20.0 -> 2.22.2
  - gspread-dataframe 3.0.8 -> 3.3.1
  - holidays 0.27.1 -> 0.29
  - jax 0.4.10 -> jax 0.4.13
  - jaxlib 0.4.10 -> jax 0.4.13
  - jupyterlab-widgets 3.0.7 -> 3.0.8
  - nbformat 5.9.0 -> 5.9.1
  - opencv-python-headless 4.7.0.72 -> 4.8.0.74
  - pygame 2.4.0 -> 2.5.0
  - spacy 3.5.3 -> 3.5.4
  - SQLAlchemy 2.0.16 -> 2.0.19
  - tabulate 0.8.10 -> 0.9.0
  - tensorflow-hub 0.13.0 -> 0.14.0

## 2023-06-23

- Launched AI coding features to subscribed users starting with Pro+ users in the US ([tweet](#), [post](#))
- Added the Kernel Selector in the Notebook Settings ([tweet](#))
- Fixed double space trimming issue in markdown [#3766](#)
- Fixed run button indicator not always centered [#3609](#)
- Fixed inconsistencies for automatic indentation on multi-line [#3697](#)
- Upgraded Python from 3.10.11 to 3.10.12
- Python package updates:
  - duckdb 0.7.1 -> 0.8.1
  - earthengine-api 0.1.350 -> 0.1.357
  - flax 0.6.9 -> 0.6.11
  - google-cloud-bigquery 3.9.0 -> 3.10.0
  - google-cloud-bigquery-storage 2.19.1 -> 2.20.0
  - grpcio 1.54.0 -> 1.56.0
  - holidays 0.25 -> 0.27.1
  - nbformat 5.8.0 -> 5.9.0
  - prophet 1.1.3 -> 1.1.4
  - pydata-google-auth 1.7.0 -> 1.8.0
  - spacy 3.5.2 -> 3.5.3
  - tensorboard 2.12.2 -> 2.12.3
  - xgboost 1.7.5 -> 1.7.6
- Python package inclusions:
  - gcsfs 2023.6.0
  - geopandas 0.13.2
  - google-cloud-bigquery-connection 1.12.0
  - google-cloud-functions 1.13.0
  - grpc-google-iam-v1 0.12.6
  - multidict 6.0.4
  - tensorboard-data-server 0.7.1

## 2023-06-02

- Released the new site [colab.google](#)
- Published Colab's Docker runtime image to us-docker.pkg.dev/colab-images/public/runtime ([tweet](#), [instructions](#))



```

outputName = folderOut_ML + '/Subj' + pat + '_StandardLearning_TrainRes'
np.savetxt(outputName, AllRes_train, delimiter=",")
outputName = folderOut_ML + '/Subj' + pat + '_StandardLearning_TestRes'
np.savetxt(outputName, AllRes_test, delimiter=",")
outputName = folderOut_ML + '/Subj' + pat + '_MultiClassLearning_TrainRes'
np.savetxt(outputName, AllResMulti_train, delimiter=",")
outputName = folderOut_ML + '/Subj' + pat + '_MultiClassLearning_TestRes'
np.savetxt(outputName, AllResMulti_test, delimiter=",")
outputName = folderOut_ML + '/Subj' + pat + '_MultiClassReduced_TrainRes'
np.savetxt(outputName, AllResMultiRed_train, delimiter=",")
outputName = folderOut_ML + '/Subj' + pat + '_MultiClassReduced_TestRes'
np.savetxt(outputName, AllResMultiRed_test, delimiter=",")
outputName = folderOut_ML + '/Subj' + pat + '_MultiClassClustered_TrainRes'
np.savetxt(outputName, AllResMultiClust_train, delimiter=",")
outputName = folderOut_ML + '/Subj' + pat + '_MultiClassClustered_TestRes'
np.savetxt(outputName, AllResMultiClust_test, delimiter=",")

```

```

#plot performances for this subj for each approach and all cv
performancesAll=np.dstack((AllRes_train,AllResMulti_train, AllResMulti_test,
func_plotPerformancesOfDiffApproaches_thisSubj_multiClassPaper(pat, 'Train',
performancesAll = np.dstack((AllRes_test, AllResMulti_test, AllResMultiClust_test,
func_plotPerformancesOfDiffApproaches_thisSubj_multiClassPaper(pat, 'Test',

```

```

#save avrg for this subj
AllSubjRes_train[patIndx,:,0] = np.nanmean(AllRes_train,0)
AllSubjRes_test[patIndx,:,0] = np.nanmean(AllRes_test,0)
AllSubjResMulti_train[patIndx,:,0] = np.nanmean(AllResMulti_train,0)
AllSubjResMulti_test[patIndx,:,0] = np.nanmean(AllResMulti_test,0)
AllSubjResMultiRed_train[patIndx,:,0] = np.nanmean(AllResMultiRed_train,0)
AllSubjResMultiRed_test[patIndx,:,0] = np.nanmean(AllResMultiRed_test,0)
AllSubjResMultiClust_train[patIndx,:,0] = np.nanmean(AllResMultiClust_train,0)
AllSubjResMultiClust_test[patIndx,:,0] = np.nanmean(AllResMultiClust_test,0)
AllSubj_OptimalResultsReduced_train[patIndx,:,0] = np.nanmean(OptimalV
AllSubj_OptimalResultsReduced_test[patIndx,:,0] = np.nanmean(OptimalV
AllSubj_OptimalResultsClustered_train[patIndx,:,0] = np.nanmean(Optima
AllSubj_OptimalResultsClustered_test[patIndx,:,0] = np.nanmean(Optima
AllSubjRes_train[patIndx,:,1] = np.nanstd(AllRes_train,0)
AllSubjRes_test[patIndx,:,1] = np.nanstd(AllRes_test,0)
AllSubjResMulti_train[patIndx,:,1] = np.nanstd(AllResMulti_train,0)
AllSubjResMulti_test[patIndx,:,1] = np.nanstd(AllResMulti_test,0)
AllSubjResMultiRed_train[patIndx,:,1] = np.nanstd(AllResMultiRed_train,0)
AllSubjResMultiRed_test[patIndx,:,1] = np.nanstd(AllResMultiRed_test,0)
AllSubjResMultiClust_train[patIndx,:,1] = np.nanstd(AllResMultiClust_train,0)
AllSubjResMultiClust_test[patIndx,:,1] = np.nanstd(AllResMultiClust_test,0)
AllSubj_OptimalResultsReduced_train[patIndx,:,1] = np.nanstd(OptimalV
AllSubj_OptimalResultsReduced_test[patIndx,:,1] = np.nanstd(OptimalV
AllSubj_OptimalResultsClustered_train[patIndx,:,1] = np.nanstd(OptimalV
AllSubj_OptimalResultsClustered_test[patIndx,:,1] = np.nanstd(OptimalV

```

```

#saving perofmance for all subj
meanStd=['_mean', '_std']
for ni, meanStdVal in enumerate(meanStd):
outputName = folderOut_ML + '/AllSubj_MultiClassLearning_OptimalResult'
np.savetxt(outputName, AllSubj_OptimalResultsReduced_train[:, :, ni], delimiter=",")
outputName = folderOut_ML + '/AllSubj_MultiClassLearning_OptimalResult'
np.savetxt(outputName, AllSubj_OptimalResultsReduced_test[:, :, ni], delimiter=",")
outputName = folderOut_ML + '/AllSubj_MultiClassLearning_OptimalResult'
np.savetxt(outputName, AllSubj_OptimalResultsClustered_train[:, :, ni], delimiter=",")
outputName = folderOut_ML + '/AllSubj_MultiClassLearning_OptimalResult'
np.savetxt(outputName, AllSubj_OptimalResultsClustered_test[:, :, ni], delimiter=",")
outputName = folderOut_ML + '/AllSubj_StandardLearning_TrainRes'+meanStdVal
np.savetxt(outputName, AllSubjRes_train[:, :, ni], delimiter=",")
outputName = folderOut_ML + '/AllSubj_StandardLearning_TestRes'+meanStdVal
np.savetxt(outputName, AllSubjRes_test[:, :, ni], delimiter=",")
outputName = folderOut_ML + '/AllSubj_MultiClassLearning_TrainRes'+meanStdVal
np.savetxt(outputName, AllSubjResMulti_train[:, :, ni], delimiter=",")
outputName = folderOut_ML + '/AllSubj_MultiClassLearning_TestRes'+meanStdVal
np.savetxt(outputName, AllSubjResMulti_test[:, :, ni], delimiter=",")

```

- Launched support for Google children accounts ([tweet](#))
- Launched DagsHub integration ([tweet](#), [post](#))
- Upgraded to Monaco Editor Version 0.37.1
- Fixed various Vim keybinding bugs
- Fixed issue where the N and P letters sometimes couldn't be typed ([#3664](#))
- Fixed rendering support for compositional inputs ([#3660](#), [#3679](#))
- Fixed lag in notebooks with lots of cells ([#3676](#))
- Improved support for R by adding a Runtime type notebook setting (Edit -> Notebook settings)
- Improved documentation for connecting to a local runtime (Connect -> Connect to a local runtime)
- Python package updates:
  - holidays 0.23 -> 0.25
  - jax 0.4.8 -> 0.4.10
  - jaxlib 0.4.8 -> 0.4.10
  - pip 23.0.1 -> 23.1.2
  - tensorflow-probability 0.19.0 -> 0.20.1
  - torch 2.0.0 -> 2.0.1
  - torchaudio 2.0.1 -> 2.0.2
  - torchdata 0.6.0 -> 0.6.1
  - torchtext 0.15.1 -> 0.15.2
  - torchvision 0.15.1 -> 0.15.2
  - tornado 6.2 -> 6.3.1

## 2023-05-05

- Released GPU type selection for paid users, allowing them to choose a preferred NVIDIA GPU
- Upgraded R from 4.2.3 to 4.3.0
- Upgraded Python from 3.9.16 to 3.10.11
- Python package updates:
  - attrs 22.2.0 -> 23.1.0
  - earthengine-api 0.1.349 -> earthengine-api 0.1.350
  - flax 0.6.8 -> 0.6.9
  - grpcio 1.53.0 -> 1.54.0
  - nbclient 0.7.3 -> 0.7.4
  - tensorflow-datasets 4.8.3 -> 4.9.2
  - termcolor 2.2.0 -> 2.3.0
  - zict 2.2.0 -> 3.0.0

## 2023-04-14

- Python package updates:
  - google-api-python-client 2.70.0 -> 2.84.0
  - google-auth-oauthlib 0.4.6 -> 1.0.0
  - google-cloud-bigquery 3.4.2 -> 3.9.0
  - google-cloud-datastore 2.11.1 -> 2.15.1
  - google-cloud-firestore 2.7.3 -> 2.11.0
  - google-cloud-language 2.6.1 -> 2.9.1
  - google-cloud-storage 2.7.0 -> 2.8.0
  - google-cloud-translate 3.8.4 -> 3.11.1
  - networkx 3.0 -> 3.1
  - notebook 6.3.0 -> 6.4.8
  - jax 0.4.7 -> 0.4.8
  - pandas 1.4.4 -> 1.5.3
  - spacy 3.5.1 -> 3.5.2
  - SQLAlchemy 1.4.47 -> 2.0.9
  - xgboost 1.7.4 -> 1.7.5

## 2023-03-31

- Improve bash ! syntax highlighting ([GitHub issue](#))
- Fix bug where VIM keybindings weren't working in the file editor
- Upgraded R from 4.2.2 to 4.2.3

```

np.savetxt(outputName, AllSubjResMulti_test[:,ni] , delimiter=",")
outputName = folderOut_ML + '/AllSubj_MultiClassReduced_TrainRes'+mean
np.savetxt(outputName, AllSubjResMultiRed_train[:,ni] , delimiter=",")
outputName = folderOut_ML + '/AllSubj_MultiClassReduced_TestRes'+mean
np.savetxt(outputName, AllSubjResMultiRed_test[:,ni] , delimiter=",")
outputName = folderOut_ML + '/AllSubj_MultiClassClustered_TrainRes'+me
np.savetxt(outputName, AllSubjResMultiClust_train[:,ni] , delimiter=
outputName = folderOut_ML + '/AllSubj_MultiClassClustered_TestRes'+mea
np.savetxt(outputName, AllSubjResMultiClust_test[:,ni] , delimiter="

```

```
#####
```

```

#CALCUALTING AVRG FOR ALL SUBJ (USEFUL IF THINGS RESTARTED FOR ONLY SO
AllSubjRes_train = np.zeros((len(GeneralParams.patients), 33, 2))
AllSubjRes_test = np.zeros((len(GeneralParams.patients), 33, 2))
AllSubjResMulti_train = np.zeros((len(GeneralParams.patients), 33, 2))
AllSubjResMulti_test = np.zeros((len(GeneralParams.patients),33, 2))
AllSubjResMultiRed_train = np.zeros((len(GeneralParams.patients), 33,
AllSubjResMultiRed_test = np.zeros((len(GeneralParams.patients),33, 2)
AllSubjResMultiClust_train = np.zeros((len(GeneralParams.patients), 33
AllSubjResMultiClust_test = np.zeros((len(GeneralParams.patients),33,
AllSubj_OptimalResultsRed_train= np.zeros((len(GeneralParams.patients)
AllSubj_OptimalResultsRed_test= np.zeros((len(GeneralParams.patients),
AllSubj_OptimalResultsClust_train= np.zeros((len(GeneralParams.patient
AllSubj_OptimalResultsClust_test= np.zeros((len(GeneralParams.patients
for patIndx, pat in enumerate(GeneralParams.patients):
outputName = folderOut_ML + '/Subj' + pat + '_MultiClassLearning_Optim
reader = csv.reader(open(outputName, "r"))
OptimalValuesRed_train = np.array(list(reader)).astype("float")
outputName = folderOut_ML + '/Subj' + pat + '_MultiClassLearning_Optim
reader = csv.reader(open(outputName, "r"))
OptimalValuesRed_test = np.array(list(reader)).astype("float")
outputName = folderOut_ML + '/Subj' + pat + '_MultiClassLearning_Optim
reader = csv.reader(open(outputName, "r"))
OptimalValuesClust_train = np.array(list(reader)).astype("float")
outputName = folderOut_ML + '/Subj' + pat + '_MultiClassLearning_Optim
reader = csv.reader(open(outputName, "r"))
OptimalValuesClust_test = np.array(list(reader)).astype("float")

```

```

outputName = folderOut_ML + '/Subj' + pat + '_StandardLearning_TrainRe
reader = csv.reader(open(outputName, "r"))
AllRes_train = np.array(list(reader)).astype("float")
outputName = folderOut_ML + '/Subj' + pat + '_StandardLearning_TestRes
reader = csv.reader(open(outputName, "r"))
AllRes_test = np.array(list(reader)).astype("float")
outputName = folderOut_ML + '/Subj' + pat + '_MultiClassLearning_Train
reader = csv.reader(open(outputName, "r"))
AllResMulti_train = np.array(list(reader)).astype("float")
outputName = folderOut_ML + '/Subj' + pat + '_MultiClassLearning_TestR
reader = csv.reader(open(outputName, "r"))
AllResMulti_test = np.array(list(reader)).astype("float")
outputName = folderOut_ML + '/Subj' + pat + '_MultiClassReduced_TrainR
reader = csv.reader(open(outputName, "r"))
AllResMultiRed_train = np.array(list(reader)).astype("float")
outputName = folderOut_ML + '/Subj' + pat + '_MultiClassReduced_TestRe
reader = csv.reader(open(outputName, "r"))
AllResMultiRed_test = np.array(list(reader)).astype("float")
outputName = folderOut_ML + '/Subj' + pat + '_MultiClassClustered_Trai
reader = csv.reader(open(outputName, "r"))
AllResMultiClust_train = np.array(list(reader)).astype("float")
outputName = folderOut_ML + '/Subj' + pat + '_MultiClassClustered_Test
reader = csv.reader(open(outputName, "r"))
AllResMultiClust_test = np.array(list(reader)).astype("float")

```

```
#save avrg for this subj
```

```
AllSubjRes_train[patIndx,:,0] = np.nanmean(AllRes_train,0)
```

```
AllSubjRes_test[patIndx,:,0] = np.nanmean(AllRes_test,0)
```

#### • Python package updates:

- arviz 0.12.1 -> 0.15.1
- astropy 4.3.1 -> 5.2.2
- dopamine-rl 1.0.5 -> 4.0.6
- gensim 3.6.0 -> 4.3.1
- ipykernel 5.3.4 -> 5.5.6
- ipython 7.9.0 -> 7.34.0
- jax 0.4.4 -> 0.4.7
- jaxlib 0.4.4 -> 0.4.7
- jupyter\_core 5.2.0 -> 5.3.0
- keras 2.11.0 -> 2.12.0
- lightgbm 2.2.3 -> 3.3.5
- matplotlib 3.5.3 -> 3.7.1
- nltk 3.7 -> 3.8.1
- opencv-python 4.6.0.66 -> 4.7.0.72
- plotly 5.5.0 -> 5.13.1
- pymc 4.1.4 -> 5.1.2
- seaborn 0.11.2 -> 0.12.2
- spacy 3.4.4 -> 3.5.1
- sympy 1.7.1 -> 1.11.1
- tensorboard 2.11.2 -> 2.12.0
- tensorflow 2.11.0 -> 2.12.0
- tensorflow-estimator 2.11.0 -> 2.12.0
- tensorflow-hub 0.12.0 -> 0.13.0
- torch 1.13.1 -> 2.0.0
- torchaudio 0.13.1 -> 2.0.1
- torchtext 0.14.1 -> 0.15.1
- torchvision 0.14.1 -> 0.15.1

#### 2023-03-10

- Added the [Colab editor shortcuts](#) example notebook
- Fixed triggering of @-mention and email autocomplete for large comments ([GitHub issue](#))
- Added View Resources to the Runtime menu
- Made file viewer images fit the view by default, resizing to original size on click
- When in VIM mode, enable copy as well as allowing propagation to monaco-vim to escape visual mode ([GitHub issue](#))
- Upgraded CUDA 11.6.2 -> 11.8.0 and cuDNN 8.4.0.27 -> 8.7.0.84
- Upgraded Nvidia drivers 525.78.01 -> 530.30.02
- Upgraded Python 3.8.10 -> 3.9.16
- Python package updates:
  - beautifulsoup4 4.6.3 -> 4.9.3
  - bokeh 2.3.3 -> 2.4.3
  - debugpy 1.0.0 -> 1.6.6
  - Flask 1.1.4 -> 2.2.3
  - jax 0.3.25 -> 0.4.4
  - jaxlib 0.3.25 -> 0.4.4
  - Jinja2 2.11.3 -> 3.1.2
  - matplotlib 3.2.2 -> 3.5.3
  - nbconvert 5.6.1 -> 6.5.4
  - pandas 1.3.5 -> 1.4.4
  - pandas-datareader 0.9.0 -> 0.10.0
  - pandas-profiling 1.4.1 -> 3.2.0
  - Pillow 7.1.2 -> 8.4.0
  - plotnine 0.8.0 -> 0.10.1
  - scikit-image 0.18.3 -> 0.19.3
  - scikit-learn 1.0.2 -> 1.2.2
  - scipy 1.7.3 -> 1.10.1
  - setuptools 57.4.0 -> 63.4.3
  - sklearn-pandas 1.8.0 -> 2.2.0
  - statsmodels 0.12.2 -> 0.13.5
  - urllib3 1.24.3 -> 1.26.14
  - Werkzeug 1.0.1 -> 2.2.3
  - wrapt 1.14.1 -> 1.15.0
  - xgboost 0.90 -> 1.7.4
  - xlrd 1.2.0 -> 2.0.1

```

AllSubjResMulti_train[patIndx,:,0] = np.nanmean(AllResMulti_train,0)
AllSubjResMulti_test[patIndx,:,0] = np.nanmean(AllResMulti_test,0)
AllSubjResMultiRed_train[patIndx,:,0] = np.nanmean(AllResMultiRed_train,0)
AllSubjResMultiRed_test[patIndx,:,0] = np.nanmean(AllResMultiRed_test,0)
AllSubjResMultiClust_train[patIndx,:,0] = np.nanmean(AllResMultiClust_train,0)
AllSubjResMultiClust_test[patIndx,:,0] = np.nanmean(AllResMultiClust_test,0)
AllSubj_OptimalResultsRed_train[patIndx,:,0] = np.nanmean(OptimalValueRed_train,0)
AllSubj_OptimalResultsRed_test[patIndx,:,0] = np.nanmean(OptimalValueRed_test,0)
AllSubj_OptimalResultsClust_train[patIndx,:,0] = np.nanmean(OptimalValueClust_train,0)
AllSubj_OptimalResultsClust_test[patIndx,:,0] = np.nanmean(OptimalValueClust_test,0)
AllSubjRes_train[patIndx,:,1] = np.nanstd(AllRes_train,0)
AllSubjRes_test[patIndx,:,1] = np.nanstd(AllRes_test,0)
AllSubjResMulti_train[patIndx,:,1] = np.nanstd(AllResMulti_train,0)
AllSubjResMulti_test[patIndx,:,1] = np.nanstd(AllResMulti_test,0)
AllSubjResMultiRed_train[patIndx,:,1] = np.nanstd(AllResMultiRed_train,0)
AllSubjResMultiRed_test[patIndx,:,1] = np.nanstd(AllResMultiRed_test,0)
AllSubjResMultiClust_train[patIndx,:,1] = np.nanstd(AllResMultiClust_train,0)
AllSubjResMultiClust_test[patIndx,:,1] = np.nanstd(AllResMultiClust_test,0)
AllSubj_OptimalResultsRed_train[patIndx,:,1] = np.nanstd(OptimalValuesRed_train,0)
AllSubj_OptimalResultsRed_test[patIndx,:,1] = np.nanstd(OptimalValuesRed_test,0)
AllSubj_OptimalResultsClust_train[patIndx,:,1] = np.nanstd(OptimalValuesClust_train,0)
AllSubj_OptimalResultsClust_test[patIndx,:,1] = np.nanstd(OptimalValuesClust_test,0)

```

```

#saving perofmance for all subj
meanStd=['_mean', '_std']
for ni, meanStdVal in enumerate(meanStd):
    outputName = folderOut_ML + '/AllSubj_MultiClassLearning_OptimalResult'
    np.savetxt(outputName, AllSubj_OptimalResultsRed_train[:,ni], delimiter=',')
    outputName = folderOut_ML + '/AllSubj_MultiClassLearning_OptimalResult'
    np.savetxt(outputName, AllSubj_OptimalResultsRed_test[:,ni], delimiter=',')
    outputName = folderOut_ML + '/AllSubj_MultiClassLearning_OptimalResult'
    np.savetxt(outputName, AllSubj_OptimalResultsClust_train[:,ni], delimiter=',')
    outputName = folderOut_ML + '/AllSubj_MultiClassLearning_OptimalResult'
    np.savetxt(outputName, AllSubj_OptimalResultsClust_test[:,ni], delimiter=',')
    outputName = folderOut_ML + '/AllSubj_StandardLearning_TrainRes'+meanStdVal
    np.savetxt(outputName, AllSubjRes_train[:,ni], delimiter=',')
    outputName = folderOut_ML + '/AllSubj_StandardLearning_TestRes'+meanStdVal
    np.savetxt(outputName, AllSubjRes_test[:,ni], delimiter=',')
    outputName = folderOut_ML + '/AllSubj_MultiClassLearning_TrainRes'+meanStdVal
    np.savetxt(outputName, AllSubjResMulti_train[:,ni], delimiter=',')
    outputName = folderOut_ML + '/AllSubj_MultiClassLearning_TestRes'+meanStdVal
    np.savetxt(outputName, AllSubjResMulti_test[:,ni], delimiter=',')
    outputName = folderOut_ML + '/AllSubj_MultiClassReduced_TrainRes'+meanStdVal
    np.savetxt(outputName, AllSubjResMultiRed_train[:,ni], delimiter=',')
    outputName = folderOut_ML + '/AllSubj_MultiClassReduced_TestRes'+meanStdVal
    np.savetxt(outputName, AllSubjResMultiRed_test[:,ni], delimiter=',')
    outputName = folderOut_ML + '/AllSubj_MultiClassClustered_TrainRes'+meanStdVal
    np.savetxt(outputName, AllSubjResMultiClust_train[:,ni], delimiter=',')
    outputName = folderOut_ML + '/AllSubj_MultiClassClustered_TestRes'+meanStdVal
    np.savetxt(outputName, AllSubjResMultiClust_test[:,ni], delimiter=',')

```

```

#mean of all subj
TotalMean_2class_train=np.zeros((2,33))
TotalMean_2class_test=np.zeros((2,33))
TotalMean_Multi_train=np.zeros((2,33))
TotalMean_Multi_test=np.zeros((2,33))
TotalMean_MultiRed_train=np.zeros((2,33))
TotalMean_MultiRed_test=np.zeros((2,33))
TotalMean_MultiClust_train=np.zeros((2,33))
TotalMean_MultiClust_test=np.zeros((2,33))
TotalMean_2class_train[0,:] = np.nanmean(AllSubjRes_train[:,0],0)
TotalMean_2class_test[0,:] = np.nanmean(AllSubjRes_test[:,0],0)
TotalMean_Multi_train[0,:] = np.nanmean(AllSubjResMulti_train[:,0],0)
TotalMean_Multi_test[0,:] = np.nanmean(AllSubjResMulti_test[:,0],0)
TotalMean_MultiRed_train[0,:] = np.nanmean(AllSubjResMultiRed_train[:,0],0)
TotalMean_MultiRed_test[0,:] = np.nanmean(AllSubjResMultiRed_test[:,0],0)

```

## 2023-02-17

- Show graphs of RAM and disk usage in notebook toolbar
- Copy cell links directly to the clipboard instead of showing a dialog when clicking on the link icon in the cell toolbar
- Updated the [Colab Marketplace VM image](#)
- Upgraded CUDA to 11.6.2 and cuDNN to 8.4.0.27
- Python package updates:
  - tensorflow 2.9.2 -> 2.11.0
  - tensorboard 2.9.1 -> 2.11.2
  - keras 2.9.0 -> 2.11.0
  - tensorflow-estimator 2.9.0 -> 2.11.0
  - tensorflow-probability 0.17.0 -> 0.19.0
  - tensorflow-gcs-config 2.9.0 -> 2.11.0
  - earthengine-api 0.1.339 -> 0.1.341
  - flatbuffers 1.12 -> 23.1.21
  - platformdirs 2.6.2 -> 3.0.0
  - pydata-google-auth 1.6.0 -> 1.7.0
  - python-utils 3.4.5 -> 3.5.2
  - tenacity 8.1.0 -> 8.2.1
  - tifffile 2023.1.23.1 -> 2023.2.3
  - notebook 5.7.16 -> 6.3.0
  - tornado 6.0.4 -> 6.2
  - aiohttp 3.8.3 -> 3.8.4
  - charset-normalizer 2.1.1 -> 3.0.1
  - fastai 2.7.0 -> 2.7.1
  - soundfile 0.11.0 -> 0.12.1
  - typing-extensions 4.4.0 -> 4.5.0
  - widgetsnbextension 3.6.1 -> 3.6.2
  - pydantic 1.10.4 -> 1.10.5
  - zipp 3.12.0 -> 3.13.0
  - numpy 1.21.6 -> 1.22.4
  - drivefs 66.0 -> 69.0
  - gdal 3.0.4 -> 3.3.2 [GitHub issue](#)
- Added libudunits2-dev for smoother R package installs [GitHub issue](#)

## 2023-02-03

- Improved tooltips for pandas series to show common statistics about the series object
- Made the forms dropdown behave like an autocomplete box when it allows input
- Updated the nvidia driver from 460.32.03 to 510.47.03
- Python package updates:
  - absl-py 1.3.0 -> 1.4.0
  - bleach 5.0.1 -> 6.0.0
  - cachetools 5.2.1 -> 5.3.0
  - cmdstanpy 1.0.8 -> 1.1.0
  - dnspython 2.2.1 -> 2.3.0
  - fsspec 2022.11.0 -> 2023.1.0
  - google-cloud-bigquery-storage 2.17.0 -> 2.18.1
  - holidays 0.18 -> 0.19
  - jupyter-core 5.1.3 -> 5.2.0
  - packaging 21.3 -> 23.0
  - prometheus-client 0.15.0 -> 0.16.0
  - pyct 0.4.8 -> 0.5.0
  - pydata-google-auth 1.5.0 -> 1.6.0
  - python-slugify 7.0.0 -> 8.0.0
  - sqlalchemy 1.4.46 -> 2.0.0
  - tensorflow-io-gcs-filesystem 0.29.0 -> 0.30.0
  - tifffile 2022.10.10 -> 2023.1.23.1
  - zipp 3.11.0 -> 3.12.0
  - Pinned sqlalchemy to version 1.4.46

## 2023-01-12



```

TotalMean_MultiClust_train[0,:] = np.nanmean(AllSubjResMultiClust_train[0,:])
TotalMean_MultiClust_test[0,:] = np.nanmean(AllSubjResMultiClust_test[0,:])

TotalMean_2class_train[1,:] = np.nanstd(AllSubjRes_train[:,0],0)
TotalMean_2class_test[1,:] = np.nanstd(AllSubjRes_test[:,0],0)
TotalMean_Multi_train[1,:] = np.nanstd(AllSubjResMulti_train[:,0],0)
TotalMean_Multi_test[1,:] = np.nanstd(AllSubjResMulti_test[:,0],0)
TotalMean_MultiRed_train[1,:] = np.nanstd(AllSubjResMultiRed_train[:,0],0)
TotalMean_MultiRed_test[1,:] = np.nanstd(AllSubjResMultiRed_test[:,0],0)
TotalMean_MultiClust_test[1,:] = np.nanstd(AllSubjResMultiClust_test[:,0],0)
TotalMean_MultiClust_train[1,:] = np.nanstd(AllSubjResMultiClust_train[:,0],0)

outputName = folderOut_ML + '/AllSubjAvrg_StandardLearning_TrainRes.csv'
np.savetxt(outputName, TotalMean_2class_train, delimiter=",")
outputName = folderOut_ML + '/AllSubjAvrg_StandardLearning_TestRes.csv'
np.savetxt(outputName, TotalMean_2class_test, delimiter=",")
outputName = folderOut_ML + '/AllSubjAvrg_MultiClassLearning_TrainRes.csv'
np.savetxt(outputName, TotalMean_Multi_train, delimiter=",")
outputName = folderOut_ML + '/AllSubjAvrg_MultiClassLearning_TestRes.csv'
np.savetxt(outputName, TotalMean_Multi_test, delimiter=",")
outputName = folderOut_ML + '/AllSubjAvrg_MultiClassReduced_TrainRes.csv'
np.savetxt(outputName, TotalMean_MultiRed_train, delimiter=",")
outputName = folderOut_ML + '/AllSubjAvrg_MultiClassReduced_TestRes.csv'
np.savetxt(outputName, TotalMean_MultiRed_test, delimiter=",")
outputName = folderOut_ML + '/AllSubjAvrg_MultiClassClustered_TrainRes.csv'
np.savetxt(outputName, TotalMean_MultiClust_train, delimiter=",")
outputName = folderOut_ML + '/AllSubjAvrg_MultiClassClustered_TestRes.csv'
np.savetxt(outputName, TotalMean_MultiClust_test, delimiter=",")

#####
#####
#####
#PLOT PREDICTIONS, PERFORMANCE PER SUBJ AND MODEL
folderOutPredictionsPlots=folderOut_ML+'/Plots_predictions'
createFolderIfNotExists(folderOutPredictionsPlots)

#####
#PLOTS FOR THE MULTICLASS PAPER - ONE SET OF PARAMETERS (that are set

# plot comparison between 2C, MC, MCred, MCclust performance for this
#funcnt_plotPerformancesForMultiClassPaper_SingleParamsSetup(folderOut_ML,
dataToPlotMean_train=np.dstack((TotalMean_2class_train,TotalMean_MultiClust_train)),
dataToPlotMean_test=np.dstack((TotalMean_2class_test,TotalMean_MultiClust_test)),
xLabNames = ['2C', 'MC', 'MCred', 'MCclust'])
func_plotAllPerformancesForManyApproaches(dataToPlotMean_train, dataToPlotMean_test,
xLabNames, folderOut_ML, folderOutPredictionsPlots)

#plot percentage of data per subclasses
GeneralParams.patients=['01','02','03','06','07'] #plot only for some patients
func_plotNumDataPerSubclasses_forMultiClassPaper(folderOut_ML, folderOutPredictionsPlots,
GeneralParams.patients)

# plotting numsbclases and performances when iteratively removing or adding subclasses
folderInRemov=folderOut0 + '/F1DEgmean_0.03_10/IterativeRemovingSubclasses'
folderInClust=folderOut0 + '/F1DEgmean_0.03_10/IterativeClusteringSubclasses'
func_plotWhenIterativelyRemovingSubclasses_forMultiClassPaper(folderInRemov, folderInClust,
GeneralParams.patients, folderOut0, folderOutPredictionsPlots)

#####
# PLOTTING COMPARISONS BETWEEN DIFFERENT FACTORS
folderPlots = '04_PlotsForPaper/'
createFolderIfNotExists(folderPlots)

datasetPreparationTypeArray=['MoreNonSeizure_Fact1', 'MoreNonSeizure_Fact5', 'MoreNonSeizure_Fact10']
factNames=['Fact1', 'Fact5', 'Fact10']
folderOutList= []
for foldI, foldN in enumerate(datasetPreparationTypeArray):
    folderOutList.append('03_predictions_' + foldN + '/' + optType + '_' + str(performance))
# #plot errorbars
# # funcnt_plotPerformancesForMultiClassPaper_ComparisonSeveralParamsSetup

```

- Added support for @-mention and email autocomplete in comments
- Improved errors when GitHub notebooks can't be loaded
- Increased color contrast for colors used for syntax highlighting in the code editor
- Added terminal access for custom GCE VM runtimes
- Upgraded Ubuntu from 18.04 LTS to 20.04 LTS ([GitHub issue](#))
- Python package updates:
  - GDAL 2.2.2 -> 2.2.3.
  - NumPy from 1.21.5 to 1.21.6.
  - attrs 22.1.0 -> 22.2.0
  - chardet 3.0.4 -> 4.0.0
  - cloudpickle 1.6.0 -> 2.2.0
  - filelock 3.8.2 -> 3.9.0
  - google-api-core 2.8.2 -> 2.11.0
  - google-api-python-client 1.12.11 -> 2.70.0
  - google-auth-http2 0.0.3 -> 0.1.0
  - google-cloud-bigquery 3.3.5 -> 3.4.1
  - google-cloud-datastore 2.9.0 -> 2.11.0
  - google-cloud-firestore 2.7.2 -> 2.7.3
  - google-cloud-storage 2.5.0 -> 2.7.0
  - holidays 0.17.2 -> holidays 0.18
  - importlib-metadata 5.2.0 -> 6.0.0
  - networkx 2.8.8 -> 3.0
  - opencv-python-headless 4.6.0.66 -> 4.7.0.68
  - pip 21.1.3 -> 22.04
  - pip-tools 6.2.0 -> 6.6.2
  - prettytable 3.5.0 -> 3.6.0
  - requests 2.23.0 -> 2.25.1
  - termcolor 2.1.1 -> 2.2.0
  - torch 1.13.0 -> 1.13.1
  - torchaudio 0.13.0 -> 0.13.1
  - torchtext 0.14.0 -> 0.14.1
  - torchvision 0.14.0 -> 0.14.1

## 2022-12-06

- Made fallback runtime version available until mid-December ([GitHub issue](#))
- Upgraded to Python 3.8 ([GitHub issue](#))
- Python package updates:
  - jax from 0.3.23 to 0.3.25, jaxlib from 0.3.22 to 0.3.25
  - pyarrow from 6.0.1 to 9.0.0
  - torch from 1.12.1 to 1.13.0
  - torchaudio from 0.12.1 to 0.13.0
  - torchvision from 0.13.1 to 0.14.0
  - torchtext from 0.13.1 to 0.14.0
  - xldr from 1.1.0 to 1.2.0
  - DriveFS from 62.0.1 to 66.0.3

- Made styling of markdown tables in outputs match markdown tables in text cells
- Improved formatting for empty interactive table rows
- Fixed syntax highlighting for variables with names that contain Python keywords ([GitHub issue](#))

## 2022-11-11

- Added more dark editor themes for Monaco (when in dark mode, "Editor colorization" appears as an option in the Editor tab of the Tools → Settings dialog)
- Fixed bug where collapsed forms were deleted on mobile ([GitHub issue](#))
- Python package updates:
  - rpy2 from 3.4.0 to 3.5.5 ([GitHub issue](#))

```
#plot boxplot only for test smooth
funct_plotPerformancesForMultiClassPaper_ComparisonSeveralParamsSetup_

# plotting 6 performances of Fac1, 5, 10 for 2c, MC, MCred and MCclust
funct_plotPerformancesForMultiClassPaper_ComparisonSeveralParamsSetup_

#plotting perf improv and num subclasses after MCred for Fact1, 5, 10
funct_plotPerformancesForMultiClassPaper_ComparisonSeveralParamsSetup_
funct_plotPerformancesForMultiClassPaper_ComparisonSeveralParamsSetup_
```

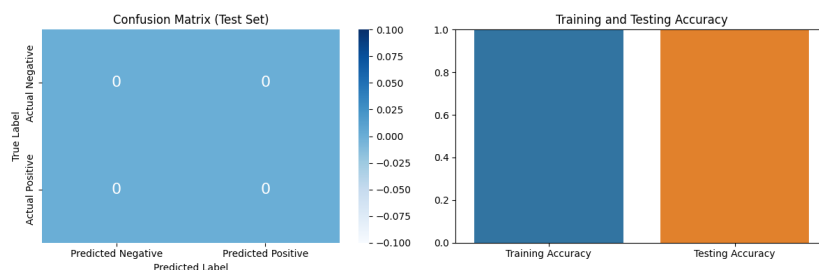


Training Accuracy: 1.00

Testing Accuracy: 1.00

Classification Report (Test Set):

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2000
accuracy			1.00	2000
macro avg	1.00	1.00	1.00	2000
weighted avg	1.00	1.00	1.00	2000



- notebook from 5.5.0 to 5.7.16
- tornado from 5.1.1 to 6.0.4
- tensorflow-probability from 0.16.0 to 0.17.0
- pandas-gbq from 0.13.3 to 0.17.9
- protobuf from 3.17.3 to 3.19.6
- google-api-core[grpc] from 1.31.5 to 2.8.2
- google-cloud-bigquery from 1.21.0 to 3.3.5
- google-cloud-core from 1.0.1 to 2.3.2
- google-cloud-datastore from 1.8.0 to 2.9.0
- google-cloud-firestore from 1.7.0 to 2.7.2
- google-cloud-language from 1.2.0 to 2.6.1
- google-cloud-storage from 1.18.0 to 2.5.0
- google-cloud-translate from 1.5.0 to 3.8.4

## 2022-10-21

- Launched a single-click way to get from BigQuery to Colab to further explore query results ([announcement](#))
- Launched [Pro, Pro+, and Pay As You Go](#) to 19 additional countries: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czechia, Denmark, Estonia, Finland, Greece, Hungary, Latvia, Lithuania, Norway, Portugal, Romania, Slovakia, Slovenia, and Sweden ([tweet](#))
- Updated jax from 0.3.17 to 0.3.23, jaxlib from 0.3.15 to 0.3.22, TensorFlow from 2.8.2 to 2.9.2, CUDA from 11.1 to 11.2, and cuDNN from 8.0 to 8.1 ([backend-info](#))
- Added a readonly option to [drive.mount](#)
- Fixed bug where Xarray was not working ([GitHub issue](#))
- Modified Markdown parsing to ignore block quote symbol within MathJax ([GitHub issue](#))

## 2022-09-30

- Launched [Pay As You Go](#), allowing premium GPU access without requiring a subscription
- Added vim and tcllib to our runtime image
- Fixed bug where open files were closed on kernel disconnect ([GitHub issue](#))
- Fixed bug where the play button/execution indicator was not clickable when scrolled into the cell output ([GitHub issue](#))
- Updated the styling for form titles so that they avoid obscuring the code editor
- Created a GitHub repo, [backend-info](#), with the latest apt-list.txt and pip-freeze.txt files for the Colab runtime ([GitHub issue](#))
- Added [files.upload](#) [file\(filename\)](#) to upload a file from the browser to the runtime with a specified filename

## 2022-09-16

- Upgraded pymc from 3.11.0 to 4.1.4, jax from 0.3.14 to 0.3.17, jaxlib from 0.3.14 to 0.3.15, fsspec from 2022.8.1 to 2022.8.2
- Modified our save flow to avoid persisting Drive filenames as titles in notebook JSON
- Updated our [Terms of Service](#)
- Modified the Jump to Cell command to locate the cursor at the end of the command palette input (Jump to cell in Tools → Command palette in a notebook with section headings)
- Updated the styling of the Drive notebook comment UI
- Added support for terminating your runtime from code: python from google.colab

- import runtime runtime.unassign()
- Added regex filter support to the Recent notebooks dialog
- Inline google.colab.files.upload JS to fix files.upload() not working ([GitHub issue](#))

## 2022-08-26

- Upgraded PyYAML from 3.13 to 6.0 ([GitHub issue](#)), drivefs from 61.0.3 to 62.0.1
- Upgraded TensorFlow from 2.8.2 to 2.9.1 and ipywidgets from 7.7.1 to 8.0.1 but rolled both back due to a number of user reports ([GitHub issue](#), [GitHub issue](#))
- Stop persisting inferred titles in notebook JSON ([GitHub issue](#))
- Fix bug in background execution which affected some Pro+ users ([GitHub issue](#))
- Fix bug where Download as .py incorrectly handled text cells ending in a double quote
- Fix bug for Pro and Pro+ users where we weren't honoring the preference (Tools → Settings) to use a temporary scratch notebook as the default landing page
- Provide undo/redo for scratch cells
- When writing ipynb files, serialize empty multiline strings as [ ] for better consistency with JupyterLab

## 2022-08-11

- Upgraded ipython from 5.5.0 to 7.9.0, fbprophet 0.7 to prophet 1.1, tensorflow-datasets from 4.0.1 to 4.6.0, drivefs from 60.0.2 to 61.0.3, pytorch from 1.12.0 to 1.12.1, numba from 0.51 to 0.56, and lxml from 4.2.0 to 4.9.1
- Loosened our requests version requirement ([GitHub issue](#))
- Removed support for TensorFlow 1
- Added Help → Report Drive abuse for Drive notebooks
- Fixed indentation for Python lines ending in [
- Modified styling of tables in Markdown to left-align them rather than centering them
- Fixed special character replacement when copying interactive tables as Markdown
- Fixed ansi 8-bit color parsing ([GitHub issue](#))
- Configured logging to preempt transitive imports and other loading from implicitly configuring the root logger
- Modified forms to use a value of None instead of causing a parse error when clearing raw and numeric-typed form fields

## 2022-07-22

- Update scipy from 1.4.1 to 1.7.3, drivefs from 59.0.3 to 60.0.2, pytorch from 1.11 to 1.12, jax & jaxlib from 0.3.8 to 0.3.14, opencv-python from 4.1.2.30 to 4.6.0.66, spaCy from 3.3.1 to 3.4.0, and dlib from 19.18.0 to 19.24.0
- Fix Open in tab doc link which was rendering incorrectly ([GitHub issue](#))
- Add a preference for the default tab orientation to the Site section of the settings menu under Tools → Settings
- Show a warning for USE\_AUTH\_EPHEM usage when running authenticate\_user on a TPU runtime ([code](#))

## 2022-07-01

- Add a preference for code font to the settings menu under Tools → Settings

- Update drivefs from 58.0.3 to 59.0.3 and spacy from 2.2.4 to 3.3.1
- Allow [display\\_data](#) and [execute\\_result](#) text outputs to wrap, matching behavior of JupyterLab (does not affect stream outputs/print statements).
- Improve LSP handling of some magics, esp. %%writefile ([GitHub issue](#)).
- Add a [FAQ entry](#) about the mount Drive button behavior and include link buttons for each FAQ entry.
- Fix bug where the notebook was sometimes hidden behind other tabs on load when in single pane view.
- Fix issue with inconsistent scrolling when an editor is in multi-select mode.
- Fix bug where clicking on a link in a form would navigate away from the notebook
- Show a confirmation dialog before performing Replace all from the Find and replace pane.

### 2022-06-10

- Update drivefs from 57.0.5 to 58.0.3 and tensorflow from 2.8.0 to 2.8.2
- Support more than 100 repos in the GitHub repo selector shown in the open dialog and the clone to GitHub dialog
- Show full notebook names on hover in the open dialog
- Improve the color contrast for links, buttons, and the ipywidgets.Accordion widget in dark mode

### 2022-05-20

- Support URL params for linking to some common pref settings: [force\\_theme=dark](#), [force\\_corgi\\_mode=1](#), [force\\_font\\_size=14](#). Params forced by URL are not persisted unless saved using Tools → Settings.
- Add a class markdown-google-sans to allow Markdown to render in Google Sans
- Update monaco-vim from 0.1.19 to 0.3.4
- Update drivefs from 55.0.3 to 57.0.5, jax from 0.3.4 to 0.3.8, and jaxlib from 0.3.2 to 0.3.7

### 2022-04-29

- Added 🐛 mode (under Miscellaneous in Tools → Settings)
- Added "Disconnect and delete runtime" option to the menu next to the Connect button
- Improved rendering of filter options in an interactive table
- Added git-lfs to the base image
- Updated torch from 1.10.0 to 1.11.0, jupyter-core from 4.9.2 to 4.10.0, and cmake from 3.12.0 to 3.22.3
- Added more details to our [FAQ](#) about unsupported uses (using proxies, downloading torrents, etc.)
- Fixed [issue](#) with apt-get dependencies

### 2022-04-15

- Add an option in the file browser to show hidden files.
- Upgrade gdown from 4.2.0 to 4.4.0, google-api-core[grpc] from 1.26.0 to 1.31.5, and pytz from 2018.4 to 2022.1

### 2022-03-25



- Launched [Pro/Pro+](#) to 12 additional countries: Australia, Bangladesh, Colombia, Hong Kong,