Thomas Martin, Tran Huynh, Sahil Sutaria
CS 663
12-2-19

# Project 3 LSTM Documentation: Part 3 – Batch Prediction

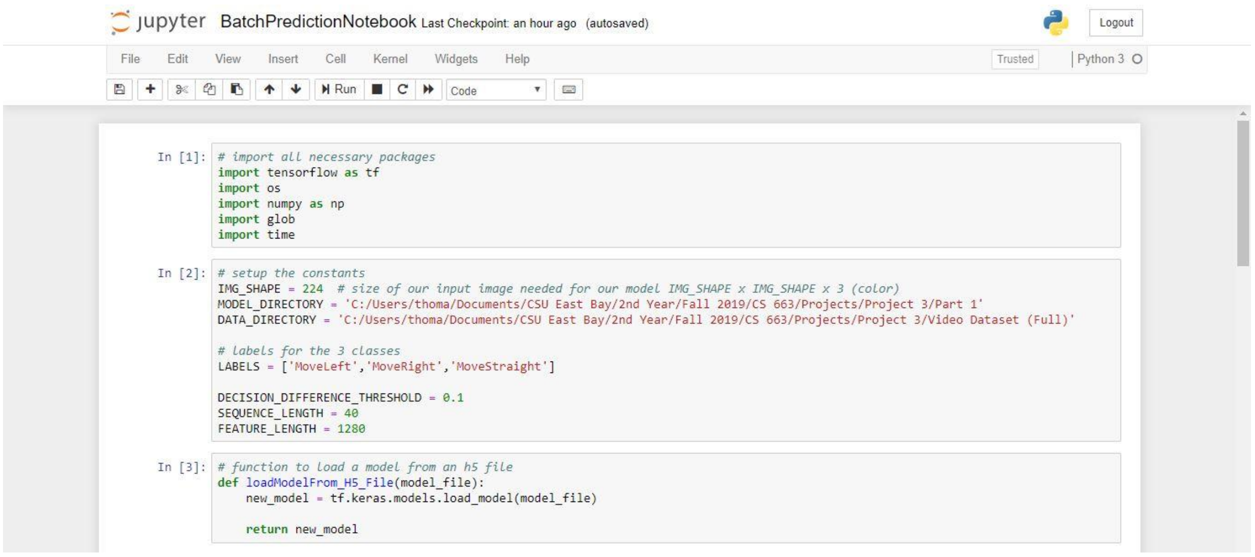
Link to YouTube Video: https://www.youtube.com/watch?v=Ezknqr8czRY&feature=youtu.be

Link to GitHub Repository: https://github.com/tmartin293/CS663_Crosswalk_Detection

**Installation Instructions**:

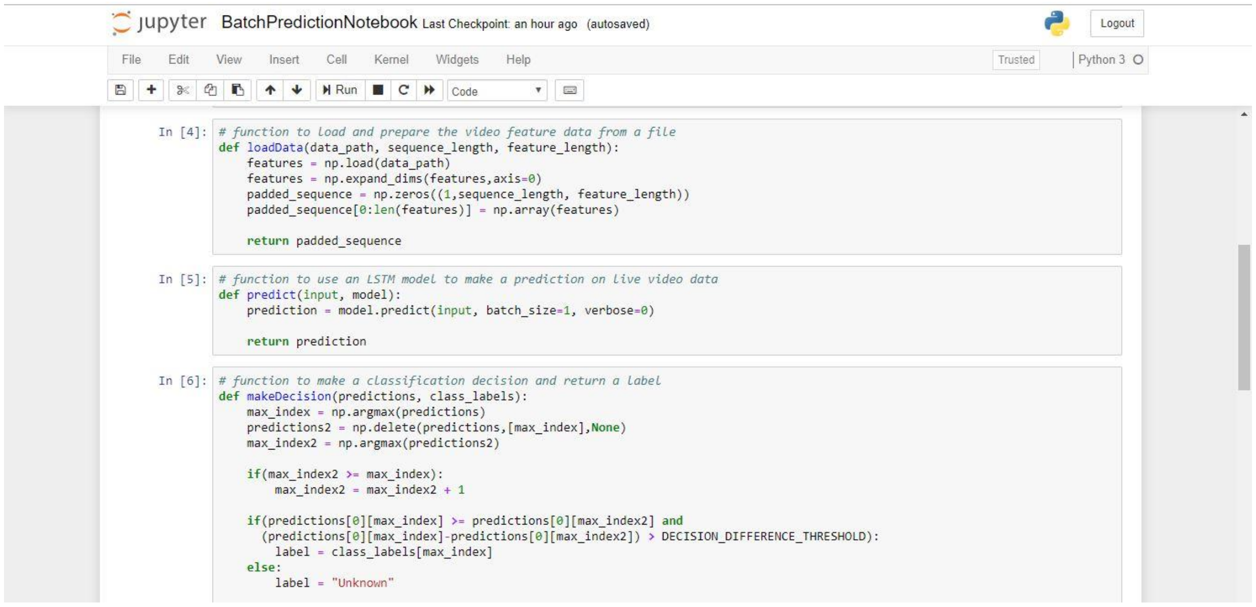
- All Juypter Notebooks were tested using a Python 3.7.1 kernel and TensorFlow 2.0.
- Please pip install or conda install all required packages prior to testing the Jupyter Notebook or the Python script: TensorFlow 2.0, glob, and numpy.

**Screenshots**:



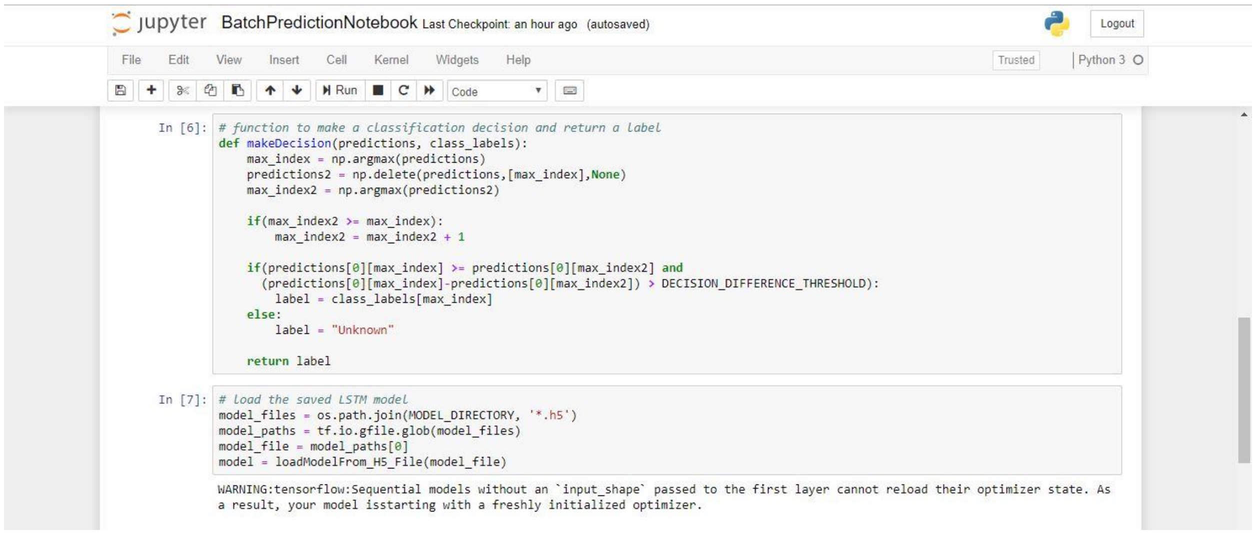
```
In [1]: # import all necessary packages
        import tensorflow as tf
        import os
        import numpy as np
        import glob
        import time

In [2]: # setup the constants
        IMG_SHAPE = 224  # size of our input image needed for our model IMG_SHAPE x IMG_SHAPE x 3 (color)
        MODEL_DIRECTORY = 'C:/Users/thoma/Documents/CSU East Bay/2nd Year/Fall 2019/CS 663/Projects/Project 3/Part 1'
        DATA_DIRECTORY = 'C:/Users/thoma/Documents/CSU East Bay/2nd Year/Fall 2019/CS 663/Projects/Project 3/Video Dataset (Full)'

        # labels for the 3 classes
        LABELS = ['MoveLeft','MoveRight','MoveStraight']

        DECISION_DIFFERENCE_THRESHOLD = 0.1
        SEQUENCE_LENGTH = 40
        FEATURE_LENGTH = 1280

In [3]: # function to load a model from an h5 file
        def loadModelFrom_H5_File(model_file):
            new_model = tf.keras.models.load_model(model_file)

            return new_model
```

Figure 1: Cells 1-3 of the Jupyter Notebook


jupyter BatchPredictionNotebook Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

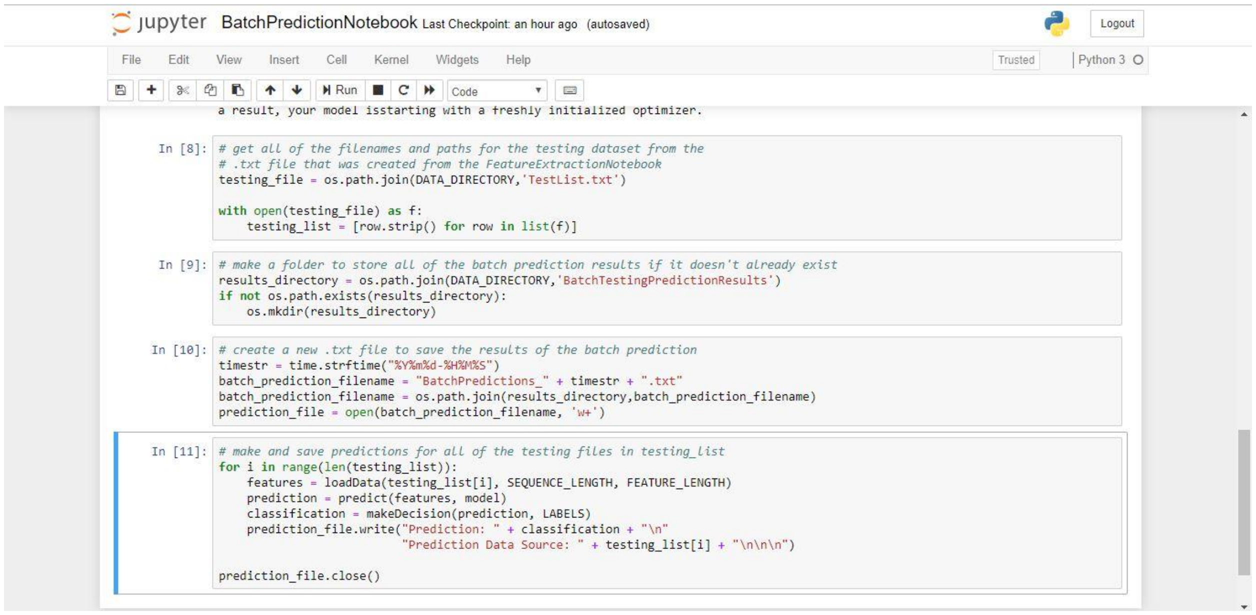
```
In [4]: # function to load and prepare the video feature data from a file
        def loadData(data_path, sequence_length, feature_length):
            features = np.load(data_path)
            features = np.expand_dims(features,axis=0)
            padded_sequence = np.zeros((1,sequence_length, feature_length))
            padded_sequence[0:len(features)] = np.array(features)

            return padded_sequence

In [5]: # function to use an LSTM model to make a prediction on live video data
        def predict(input, model):
            prediction = model.predict(input, batch_size=1, verbose=0)

            return prediction

In [6]: # function to make a classification decision and return a label
        def makeDecision(predictions, class_labels):
            max_index = np.argmax(predictions)
            predictions2 = np.delete(predictions,[max_index],None)
            max_index2 = np.argmax(predictions2)

            if(max_index2 >= max_index):
                max_index2 = max_index2 + 1

            if(predictions[0][max_index] >= predictions[0][max_index2] and
              (predictions[0][max_index]-predictions[0][max_index2]) > DECISION_DIFFERENCE_THRESHOLD):
                label = class_labels[max_index]
            else:
                label = "Unknown"
```

Figure 1: Cells 4-6 of the Jupyter Notebook


jupyter BatchPredictionNotebook Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

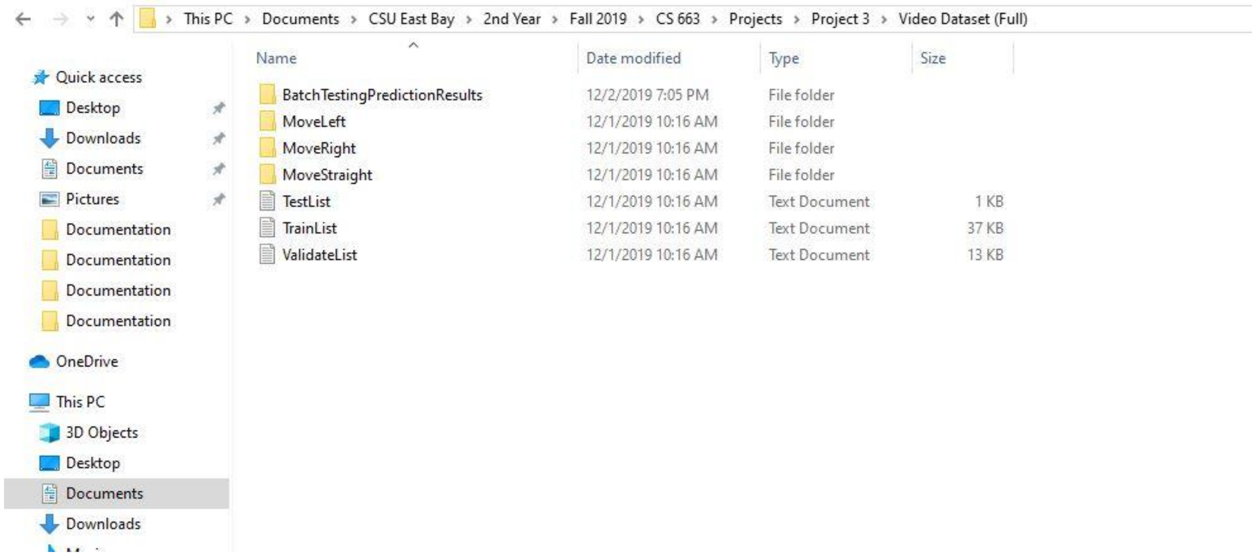
```
In [6]: # function to make a classification decision and return a label
        def makeDecision(predictions, class_labels):
            max_index = np.argmax(predictions)
            predictions2 = np.delete(predictions,[max_index],None)
            max_index2 = np.argmax(predictions2)

            if(max_index2 >= max_index):
                max_index2 = max_index2 + 1

            if(predictions[0][max_index] >= predictions[0][max_index2] and
              (predictions[0][max_index]-predictions[0][max_index2]) > DECISION_DIFFERENCE_THRESHOLD):
                label = class_labels[max_index]
            else:
                label = "Unknown"

            return label

In [7]: # load the saved LSTM model
        model_files = os.path.join(MODEL_DIRECTORY, '*.h5')
        model_paths = tf.io.gfile.glob(model_files)
        model_file = model_paths[0]
        model = loadModelFrom_H5_File(model_file)

        WARNING:tensorflow:Sequential models without an `input_shape` passed to the first layer cannot reload their optimizer state. As
        a result, your model isstarting with a freshly initialized optimizer.
```

Figure 3: Cells 6 and 7 of the Jupyter Notebook


jupyter BatchPredictionNotebook Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

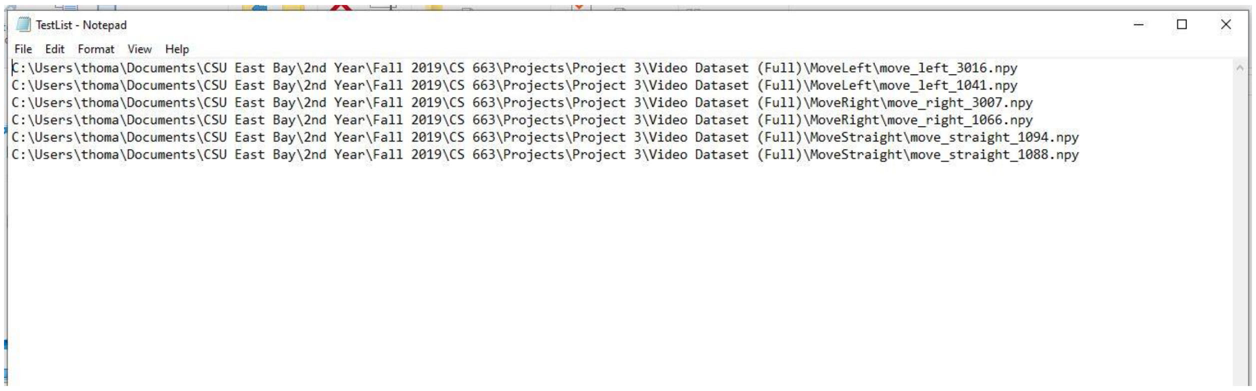
```
a result, your model isstarting with a freshly initialized optimizer.

In [8]: # get all of the filenames and paths for the testing dataset from the
        # .txt file that was created from the FeatureExtractionNotebook
        testing_file = os.path.join(DATA_DIRECTORY,'TestList.txt')

        with open(testing_file) as f:
            testing_list = [row.strip() for row in list(f)]

In [9]: # make a folder to store all of the batch prediction results if it doesn't already exist
        results_directory = os.path.join(DATA_DIRECTORY,'BatchTestingPredictionResults')
        if not os.path.exists(results_directory):
            os.mkdir(results_directory)

In [10]: # create a new .txt file to save the results of the batch prediction
         timestr = time.strftime("%Y%m%d-%H%M%S")
         batch_prediction_filename = "BatchPredictions_" + timestr + ".txt"
         batch_prediction_filename = os.path.join(results_directory,batch_prediction_filename)
         prediction_file = open(batch_prediction_filename, 'w+')

In [11]: # make and save predictions for all of the testing files in testing_list
         for i in range(len(testing_list)):
             features = loadData(testing_list[i], SEQUENCE_LENGTH, FEATURE_LENGTH)
             prediction = predict(features, model)
             classification = makeDecision(prediction, LABELS)
             prediction_file.write("Prediction: " + classification + "\n"
                                   "Prediction Data Source: " + testing_list[i] + "\n\n\n")

         prediction_file.close()
```

Figure 4: Cells 8-11 of the Jupyter Notebook



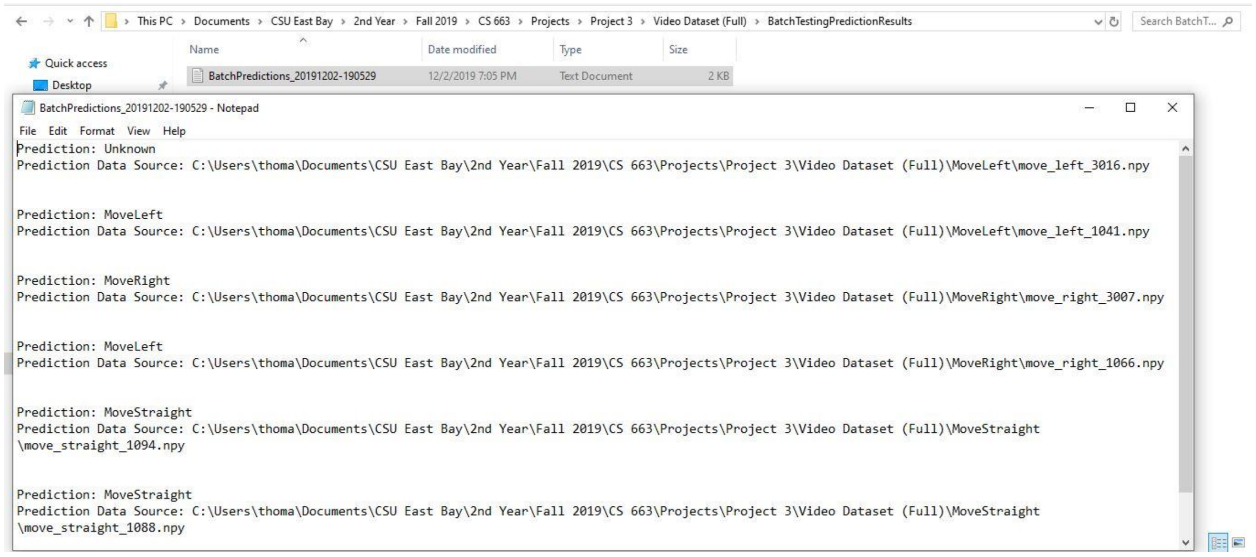
| Name | Date modified | Type | Size |
|---|---|---|---|
| BatchTestingPredictionResults | 12/2/2019 7:05 PM | File folder | |
| MoveLeft | 12/1/2019 10:16 AM | File folder | |
| MoveRight | 12/1/2019 10:16 AM | File folder | |
| MoveStraight | 12/1/2019 10:16 AM | File folder | |
| TestList | 12/1/2019 10:16 AM | Text Document | 1 KB |
| TrainList | 12/1/2019 10:16 AM | Text Document | 37 KB |
| ValidateList | 12/1/2019 10:16 AM | Text Document | 13 KB |

Figure 5: Directory Containing Batch Testing Results, Testing List, and Testing Data



```
C:\Users\thoma\Documents\CSU East Bay\2nd Year\Fall 2019\CS 663\Projects\Project 3\Video Dataset (Full)\MoveLeft\move_left_3016.npy
C:\Users\thoma\Documents\CSU East Bay\2nd Year\Fall 2019\CS 663\Projects\Project 3\Video Dataset (Full)\MoveLeft\move_left_1041.npy
C:\Users\thoma\Documents\CSU East Bay\2nd Year\Fall 2019\CS 663\Projects\Project 3\Video Dataset (Full)\MoveRight\move_right_3007.npy
C:\Users\thoma\Documents\CSU East Bay\2nd Year\Fall 2019\CS 663\Projects\Project 3\Video Dataset (Full)\MoveRight\move_right_1066.npy
C:\Users\thoma\Documents\CSU East Bay\2nd Year\Fall 2019\CS 663\Projects\Project 3\Video Dataset (Full)\MoveStraight\move_straight_1094.npy
C:\Users\thoma\Documents\CSU East Bay\2nd Year\Fall 2019\CS 663\Projects\Project 3\Video Dataset (Full)\MoveStraight\move_straight_1088.npy
```

Figure 6: Contents of TestList.txt that Contains the Two Video Files Per Class to Test


This PC > Documents > CSU East Bay > 2nd Year > Fall 2019 > CS 663 > Projects > Project 3 > Video Dataset (Full) > BatchTestingPredictionResults
Search BatchT...
Quick access
Desktop
Name
Date modified
Type
Size
BatchPredictions_20191202-190529
12/2/2019 7:05 PM
Text Document
2 KB
BatchPredictions_20191202-190529 - Notepad
File Edit Format View Help

```
Prediction: Unknown
Prediction Data Source: C:\Users\thoma\Documents\CSU East Bay\2nd Year\Fall 2019\CS 663\Projects\Project 3\Video Dataset (Full)\MoveLeft\move_left_3016.npy


Prediction: MoveLeft
Prediction Data Source: C:\Users\thoma\Documents\CSU East Bay\2nd Year\Fall 2019\CS 663\Projects\Project 3\Video Dataset (Full)\MoveLeft\move_left_1041.npy


Prediction: MoveRight
Prediction Data Source: C:\Users\thoma\Documents\CSU East Bay\2nd Year\Fall 2019\CS 663\Projects\Project 3\Video Dataset (Full)\MoveRight\move_right_3007.npy


Prediction: MoveLeft
Prediction Data Source: C:\Users\thoma\Documents\CSU East Bay\2nd Year\Fall 2019\CS 663\Projects\Project 3\Video Dataset (Full)\MoveRight\move_right_1066.npy


Prediction: MoveStraight
Prediction Data Source: C:\Users\thoma\Documents\CSU East Bay\2nd Year\Fall 2019\CS 663\Projects\Project 3\Video Dataset (Full)\MoveStraight
\move_straight_1094.npy


Prediction: MoveStraight
Prediction Data Source: C:\Users\thoma\Documents\CSU East Bay\2nd Year\Fall 2019\CS 663\Projects\Project 3\Video Dataset (Full)\MoveStraight
\move_straight_1088.npy
```

Figure 7: Contents of BatchPredictions File that Show the Prediction Value and File that the Prediction is Generated From