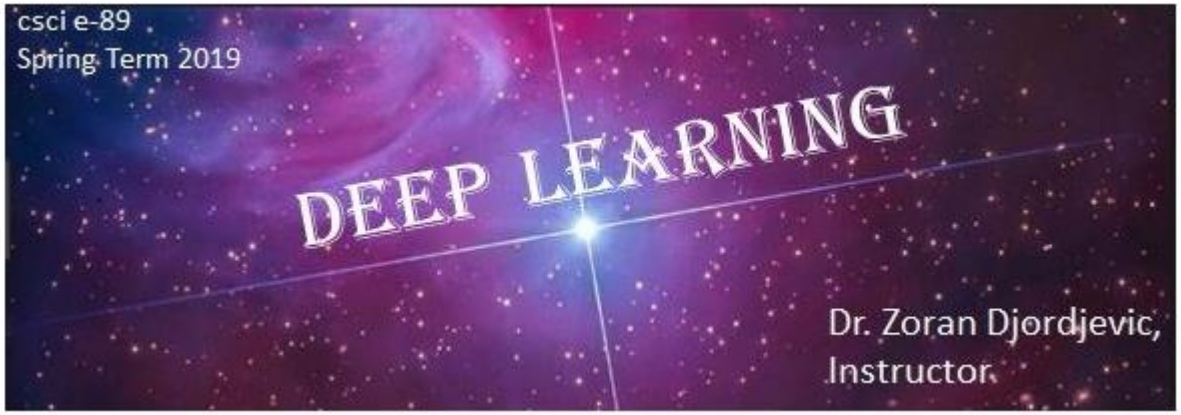


Tips and Guidance

CSCI E-89 Deep Learning

Harvard University Extension School

Diane Howard

April 17, 2019

Agenda

- **Final Project Breakdown**
  - Important Dates
  - How to approach the Final Project
  - Helpful Tips
  - Useful links for data sets & editors
  - Skeleton Report, One Page Summary, Slides
  - Creating YouTube Account, Video tools, Upload
  - Grading Criteria
  - A Peek at some examples
  - Your Questions

Important Dates

- Still time to switch topics in SignUpGenius! We close it a day or so before the due date.
- Last lecture: Friday, May 10, 2019
  - If your FP is done before May 10th contact Zoran ASAP to present your 2 minute YouTube video on May 10th.
- All parts of the Final Project are due Tuesday, May 14th 9:00 PM EST. No late submissions will be accepted.
- 2 Minute YouTube Presentations will be hosted on May 17th starting at 5:30 – 8:30.
- We will post your final project grades and comments ~May 22nd/23rd.
- We will post all final project submissions (one page summary, YouTubes, zip submissions) in Canvas.
- Grades available May 28, 2018: https://www.extension.harvard.edu/academic-calendar
  - Your FINAL GRADE is provided by Harvard (not by Zoran or via Canvas). Grades in Canvas are not official.
  - Log into your Harvard DCE account to obtain your FINAL GRADE

High Level Approach

1. Research your DL technology (Keras or TensorFlow APIs. Start with on-line demos. Must be Deep Learning models!
2. Define a problem statement for your topic.
3. Select a data source (There are many data sets available online).
4. Develop/implement/code a simple demonstration to solve your problem using Keras or TensorFlow APIs.
5. Tweak your problem statement as you work with the data, if needed.
6. Produce a visualization of your data/results.
7. Document your work (slides, one page summary, report). Report is very detailed showing all steps: install, config, runs, results (to include graphical results).
8. Create 2 YouTube videos (2 minute, 15 minute). Show your demo in your 2 minute PLEASE! Creating a Youtube video can take up quite a bit of time!

Software Demonstration Approach

1. Install required software
2. Download dataset
3. Filter data to a small data subset for the project
4. Import data into Pandas data frame using Python
5. Clean incomplete values from the data
6. Model a neural network using TensorFlow/Keras/Google/other libraries to predict/classify/determine xxx and yyy.
7. Train and Test the network on data subset
8. Plot loss and error to evaluate the quality of network
9. Visual (graphical representation of results)

HIGH LEVEL OVERVIEW OF STEPS:

1. Install software
2. Download data from state website and combine into one dataset
3. Filter and clear data
4. Adjust financial data by inflation so we can make comparison from different years
5. Create a regression model for the relationship of various educational inputs to test scores
6. Train and test the model using all the data
7. Remove the financial data from the model (Total expenditures per pupil) and test the model again for comparison
8. Visualize the predicted results vs. the target results

Example Problem Statements

- **Problem Statements:**
  - Determine authenticity of bank notes for financial institutions by classifying metadata generated by image wavelet transformations of bank notes by a feed forward neural network using Microsoft's open source Cognitive Toolkit (CNTK).
  - Develop a deep learning model using Keras to learn the task of generating a caption given an image, train the model on the Flickr8k data set, and evaluate its performance with different hyper parameter settings.
  - Explore the application of neural network to classify fraud in credit card transaction. TensorFlow is used to construct neural nets with different number of neurons and layers to explore how to increase the model prediction accuracy.
- **Problem Statement that is not well written:**
  - This is to apply what we learned from this Deep Learning course, create and train a model, which will generate a meaningful description when an image is given.

Describe a problem to solve using the technology.

Another example of a Problem Statement

- **Problem Statement:** The market price of crude oil is complex and subject to both economic and political factors. This analysis explores the possibility of predicting the future price of crude oil using a Deep Learning Feed Forward model. The Keras Python Deep Learning library is used as a high-level wrapper for Google's TensorFlow library. Grid-search cross-validation is employed using a feed-forward network architecture by varying dropout rate, units per layer, number of layers, and regularizer lambdas. The best model from this approach is then refined and tested for efficacy.

Please do not miss the boat on defining your problem statement properly.

Describe:
What you are solving & why **+** data you are using **+** technology
EQUALS
A Well Defined Problem Statement

Repositories for Public Data Sets

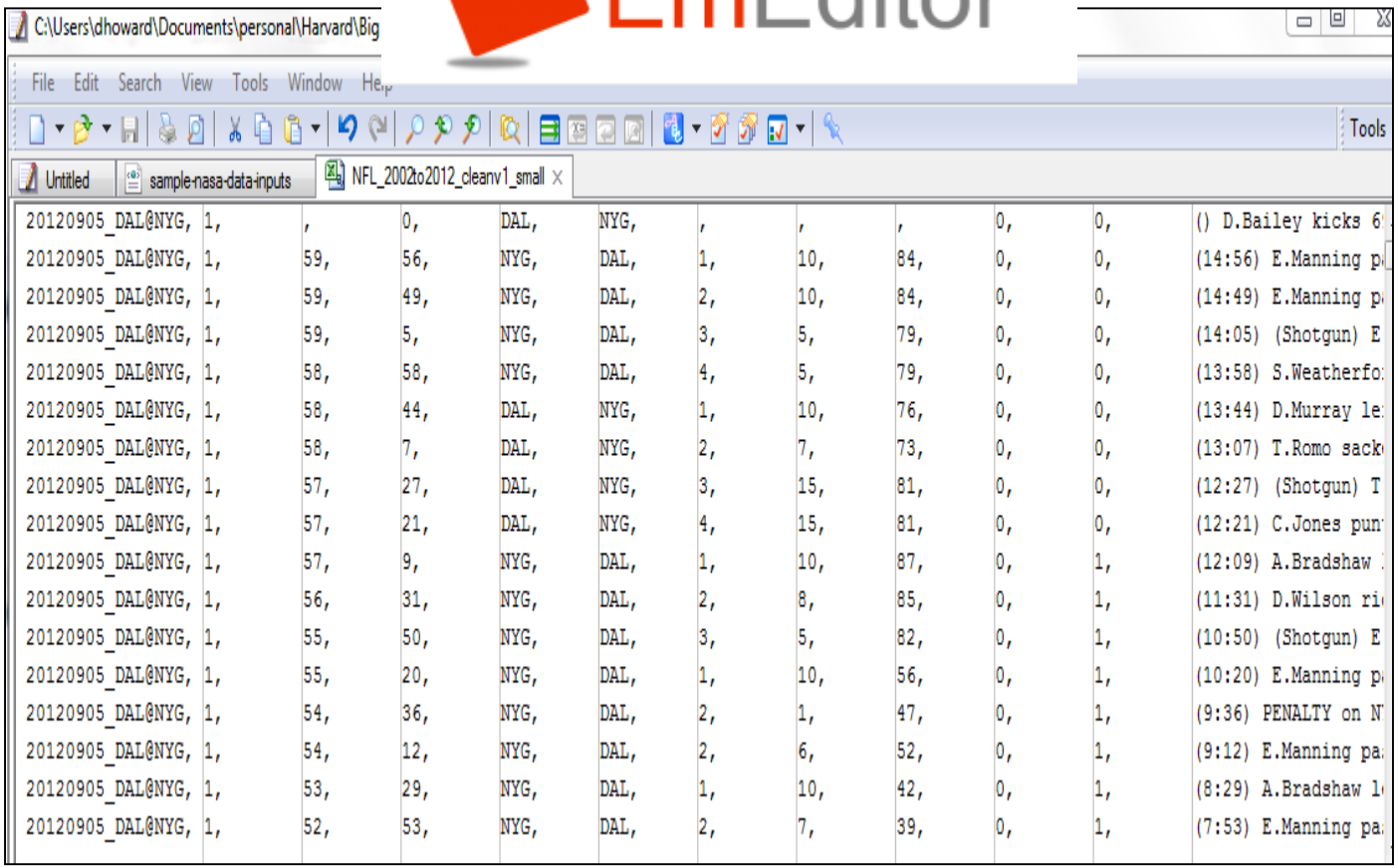
- Microsoft Azure: https://docs.microsoft.com/en-us/azure/sql-database/sql-database-public-data-sets
- Amazon Web Services: https://aws.amazon.com/public-data-sets/
- Mobile Analytics Data from Microsoft Research collection: https://www.microsoft.com/en-us/research/project/mobile-data-collection/
- U.S. government: https://www.data.gov/
- Kaggle: https://www.kaggle.com/datasets
- UC Irvine Machine Learning repository: https://archive.ics.uci.edu/ml/index.php
- http://deeplearning.net/datasets
- https://github.com/awesomedata/awesome-public-datasets
- https://www.quora.com/Where-can-I-find-large-datasets-open-to-the-public
- Dogs: http://vision.stanford.edu/aditya86/ImageNetDogs/
- Videos: https://research.google.com/youtube-bb/
- San Francisco Open Data: https://datasf.org/opendata/
- London (UK) Datastore: https://data.london.gov.uk/
- Cambridge MA: http://www.cambridgema.gov/departments/opendata
- NASA: https://open.nasa.gov/open-data/ and the data portal https://data.nasa.gov/
- Driving / image detection:
- Udacity self-driving car: https://github.com/udacity/self-driving-car/tree/master/annotations

- EAVISE at KU Leuven (images): https://iiw.kuleuven.be/onderzoek/eavise/datasets
- Belgian Traffic Sign dataset: http://btsd.ethz.ch/shareddata/
- YouTube boundingboxes (video): https://research.google.com/youtube-bb/

If your data set is larger than 10 MB, PLEASE DO NOT UPLOAD your data set.

Large Data File Editor for Windows

EmEditor http://www.emeditor.com/



Large Data File Editor for Mac, Linux

vim http://www.vim.org/

Preprocessing Data

Data may contain NaN , blanks, special characters, other placeholders or could be shorter in length than other fields.

What could you do with these outliers?

1. Remove the offending records BUT…. this comes at the price of losing data which may be valuable (even though incomplete)!
2. Replace NaN or some other special char with -1 (this could be used as your negative value. If your data is positive you could skip it)
3. Impute your data: Take the median, mean, most frequent value of your missing data.

   Example: Let's say you had missing #s for an age column

   Take the media value for the age data and apply it to the missing ages in both the training and test data set.

   Check out: the Imputer transformer of scikit-learn (median, mean, most_frequent)

   http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.Imputer.html
4. What if the input sequences have variable lengths (e.g., like sentences)? Pad with zero vector.

Cleaning Image Data

- Avoids over-fitting since NNs are prone to overfitting
- Always clean data before processing!
- 3 recommended steps for cleaning images:

1. If your image is in color, convert it to grayscale.
   - Lowers the dimensions of the input data which also lowers the # of parameters.
2. Consider center-cropping the image. i.e., Image edges may provide no useful info.
3. Normalize your input: subtract the mean & divide by the standard deviation.

Reference:
Machine Learning with TensorFlow
By N. Shukla
K. Fricklas
9.2 Cleaning data

```
import numpy as np

def clean(data):
    imgs = data.reshape(data.shape[0], 3, 32, 32)
    grayscale_imgs = imgs.mean(1)
    cropped_imgs = grayscale_imgs[:, 4:28, 4:28]
    img_data = cropped_imgs.reshape(data.shape[0], -1)
    img_size = np.shape(img_data)[1]
    means = np.mean(img_data, axis=1)
    meansT = means.reshape(len(means), 1)
    stds = np.std(img_data, axis=1)
    stdsT = stds.reshape(len(stds), 1)
    adj_stds = np.maximum(stdsT, 1.0 / np.sqrt(img_size))
    normalized = (img_data - meansT) / adj_stds
    return normalized
```

Reorganizes the data so it's a 32 × 32 matrix with three channels

Grayscales the image by averaging the color intensities

Crops the 32 × 32 image to a 24 × 24 image

Normalizes the pixels' values by subtracting the mean and dividing by standard deviation

Reshape

Convert to grayscale

Crop the size (downsizing)

subtract the mean & divide by the standard deviation.

Algorithm Considerations

- From Machine Learning with TensorFlow, Nishant Shukla with Kenneth Fricklas, Chapter 7

Real-world problem	Algorithm
Predicting trends, fitting a curve to data points, describing relationships between variables	Linear regression
Classifying data into two categories, finding the best way to split a dataset	Logistic regression
Classifying data into multiple categories	Softmax regression
Revealing hidden causes of observations, finding the most likely hidden reason for a series of outcomes	Hidden Markov model (Viterbi)
Clustering data into a fixed number of categories, automatically partitioning data points into separate classes	K-means
Clustering data into arbitrary categories, visualizing high-dimensional data into a lower-dimensional embedding	Self-organizing map

Real-world problem	Algorithm
Reducing dimensionality of data, learning latent variables responsible for high-dimensional data	Autoencoder
Planning actions in an environment using neural networks (reinforcement learning)	Q-policy neural network
Classifying data using supervised neural networks	Perceptron
Classifying real-world images using supervised neural networks	Convolution neural network
Producing patterns that match observations using neural networks	Recurrent neural network
Predicting natural language responses to natural language queries	Seq2seq model
Learning to rank items by learning their utility	Ranking

REPORTS

One Page Summary

Request per Zoran please make sure you include the following:

1. Your legal name (as registered at HU) on the top of the page.
2. Title of the project below the name, centered.
3. Text of the abstract, i.e. short project description.
4. Include 2 URLs (short, long) for YouTube videos at the bottom.
5. Filename should be called: Yourtopic_OnePageSummary_Yourname.doc

One Page Summary Example

Define your Problem Statement:

Overview of Technology:

Data Set: 'Data arises from a large study to examine EEG correlates of genetic predisposition to alcoholism. It contains measurements from 64 electrodes placed on the scalp sampled at 256 Hz.'

URL: https://archive.ics.uci.edu/ml/datasets/EEG+Database

Large size: 700 MB, Small size: 1MB Format of data file: csv

High Level Overview of steps:

1. Installed and configured (list URLs)
2. Downloaded data set (list URL)
3. Cleaned/Pre-process data
4. Ingested data using Pandas
5. Ran RNN model
6. Output Predictions and Visuals

Hardware:

Windows 10 on i7-6700, NVIDIA 1060, 16GB

Software:

Python 2.7

Em Editor

Github Atom

Azure ML version x.x

References: here

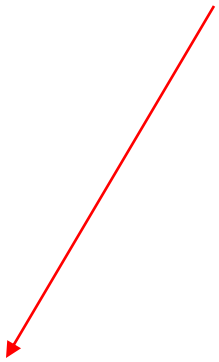
Acknowledgements of Data Sets: here

Lessons Learned & Pros/Cons

YouTube URLs: **Short video here, Long video here**

Note: If you only have a large data set – just include the URL to it!

Don't forget to add your URLs here.



Another One Page Summary

Case Study in the Arts

Problem Statement

The purpose of this project is to explore the application of Convolutional Neural Network (CNN) to generate images with artistic styles. The implementation is based on a 2015 VGG19 network. This is a 19-layer model used by the VGG team in the ILSVRC-2014 competition (winner) due to its mathematical simplicity and ample discussion available on the Internet.

Data Set: VGG19 is a 19-layer model used by the VGG team in the ILSVRC-2014 competition (winner). Its original use is object recognition, but its feature maps in each layer are also useful in many other ways. Official description can be found here.

High Level Overview of steps:

The solution is based on two key ideas: 1) higher-layer output can represent the abstract content of an input image, while lower-layer output renders more on the pixel level; 2) the output correlations between different layers can represent the style (texture) information of the input image. Based on that, we can not only separate the content and style information from any image, but also combine them together by computing minimal total loss. Essentially, the goal is to use TensorFlow to write models and perform computation on the popular pre-trained neural network VGG19.

Hardware: n1-highmem-2 instance; 2vCPU @ 2.2GHz; 13GB RAM; Tesla K80 GPU

Software: Tensorflow 1.7.0 (https://www.tensorflow.org/); Python 2.7

Reference: 1. xxx. 2. Chollet, Francois. Deep learning with Python. Manning Publications Co., 2017.

Lesson Learned & Pros/Cons:

Deep learning allows style transfer to be formulated as an optimization process using a loss defined with a pre-trained convnet.

Although the implementation used in this project comes from the original work, it is far from the best.

YouTube URLs:

Short: https://youtu.be/here
Long: https://youtu.be/here

Skeleton for PowerPoint Presentation

- 10-20 slides
- Use template provided. Only white background slides please.

Skeleton

- Problem Statement
- Description of Technology
- Description of Data
- Overview of Install/Config/Set up
- Code Highlights (Model, accuracy, loss, etc.)
- Demonstration (Results and Visualization)
- Summary: Lessons Learned & Pros/Cons
- YouTube URLs (last slide)

If you only have a large data set – just include the URL to it!

Skeleton for Full Report

Problem Statement

Description of Technology (Provide URLs of downloads)

Description of Hardware

Description of Data & provide URL where you got your Data set

Show **example data file**

Installation & configuration (if needed) steps. Capture all screens!

Describe **data wrangling/pre-processing of data.**

Show **any transformations.**

Put **all your code in your Report!**

If you only have a large data set – just include the URL to it!

Comment code. Show packages/libraries used.

Show results and visualization. Describe what is going on!

Summarize: Lessons Learned: Describe Issues / Benefits (Pros & Cons), what did you like/not like. What would you do next?

References:

Cite your URLs from code or technical info from online sources.

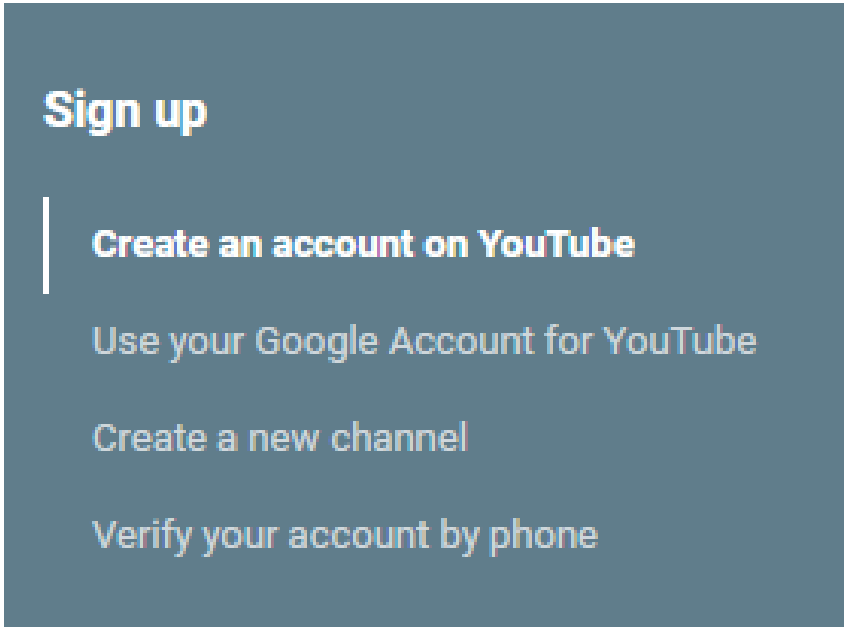
Acknowledge your data sources.

YouTube Info

Obtain a YouTube Account

https://support.google.com/youtube/answer/161805?co=GENIE.Platform%3DDesktop&hl=en

1. Need a Google/gmail account
2. Go to youtube.com.
3. In the top right, click **Sign in**.
4. Click **More options > Create Account**.



How to create your YouTube video

You will need video editing software:

https://www.iskysoft.com/video-editing/free-video-editing-software-for-youtube.html

Windows 10:

Try: Camtasia (Windows) Note: For demo version a logo will appear and can't be removed!

OR

- Open the app you want to record.
- Press the Windows key and the letter G at the same time to open the Game Bar dialog.
- Check the "Yes, this is a game" checkbox to load the Game Bar. ...
- Click on the Start Recording button (or Win + Alt + R) to begin capturing video.

Macintosh:

Use QuickTime app.

- Open your Applications folder to find QuickTime.
- Once it's open, go to File > New Screen Recording and then click the **Record** button. You can choose between recording a portion of your screen or the entire screen.

How to upload your video to YouTube

https://www.youtube.com/upload

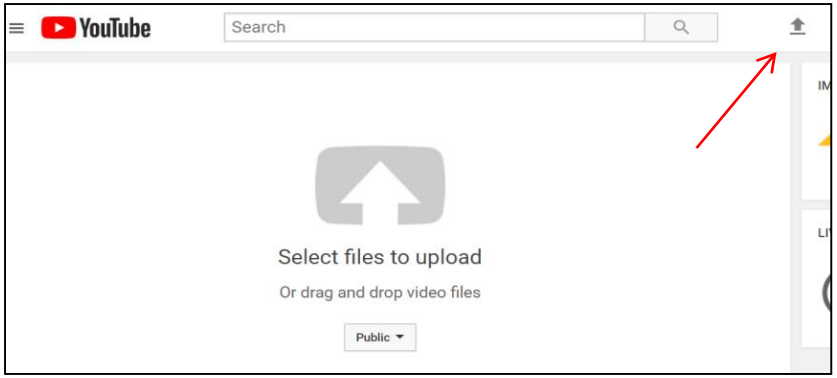
Upload videos

You can upload videos to YouTube in a few easy steps. Use the instructions below to upload your videos from a computer or from a mobile device.

COMPUTER ANDROID IPHONE & IPAD

1. Sign in to YouTube.
2. Click on **Upload** at the top of the page.
3. Before you start uploading the video, you can choose the video privacy settings.
4. Select the video you'd like to upload from your computer. You can also import a video from Google Photos.
5. As the video is uploading, you can edit both the basic information and the advanced settings of the video and decide if you want to notify subscribers (if you uncheck this option, no communication will be shared with your subscribers). Partners will also be able to adjust their Monetization settings.
6. Click **Publish** to finish uploading a public video to YouTube. If you set the video privacy setting to Private or Unlisted, just click **Done** to finish the upload or click **Share** to privately share your video.
7. If you haven't clicked **Publish**, your video won't be viewable by other people. You can always publish your video at a later time in your Video Manager.

Once the upload is completed we will send you an email to notify you that your video is done uploading and processing. You can then forward that email to friends or family for easy sharing. If you prefer not to receive notification, you can opt out by visiting your email settings.



Note: You don't need to add your videos to a specific channel.

YouTube Hints

- **Show your demo, show your demo! Show all results and visualizations**

2 minute (SHORT & SWEET)

- **1-6** slides **(REALLY!)** on Problem Statement & quick overview of technology, data set.
  - **OR do something different! Use an animated video maker! Fun, Fun!**
    - **Moovly:** https://www.moovly.com/
    - **Animaker:** https://www.animaker.com/
- Leave out extraneous info PLEASE!
  - You do not need to mention this course (we know).
  - We do not need to see a video of you.
  - Do NOT point us to your 15 minute YouTube to see your demo. ☹
  - Speed up your Intro/First slide and Overview slide!
- At the end: Add a quick one-liner: what did you learn, like, didn't like?
- Be aware of the cadence of your delivery – *not slow, change slides quickly!*

15 minute (of course this has more details)

- Describe your technology & code & results! **Show your demo! Show your demo!**
- Add your thoughts with Pros/Cons/Lessons Learned, what would you do next?

YouTube examples

Excellent examples from previous courses with Zoran!

- *Much thanks to **Stephen Camera-Murray**!* Topic: Azure ML Service

2 minute: https://youtu.be/jfmmUf9TB74

15 minute: https://youtu.be/LvV96lFHO8g

Animated video example using Animaker

- *Much thanks to **Fadwa Khalil**:* Topic: ELK

  2 minute: https://www.youtube.com/watch?v=cF3MpRaO4sA

- *Much thanks to **Stephen Joachim**!* Topic: Big Data Visualization

  2 minute: https://youtu.be/CI-RTs0cORI

  15 minute: https://youtu.be/ES9v07SYXls

Presentation Selections

- Two minute YouTube presentations will start 5:30 PM CST on Friday, May 17th. Ends at 8:30 PM. Please be present in the classroom if you are local.

- We will review YouTube presentations to determine who will be presenting. A list of presenters will be sent to all students day of/day before presentation. If we did not present your YouTube presentation we will add it in during the evening. Please also request to Zoran if you would like your FP presented.

- Zoran/TAs introduce you & your Topic via Zoom and we will start the YouTube video. Just be online in Zoom for our introduction and questions from the audience.

- We present the YouTube Videos by Topics.

- If you are done early with your project you may present your YouTube on 3rd May . Please notify Zoran.

Grading Criteria

Grading criteria:

Project Report and practical software code example	50%
PowerPoint Slides	20%
15 minute YouTube video	15%
2 minute YouTube video	10%
One page summary	5%

If you fail to provide practical software code example, your project will not be graded.

Deductions are applied…

- Missing problem statement or not a well defined problem statement.
- Missing steps in your report.
- Demo is trivial: Hello World type example.
- Reuse of demo on-line, grade=0.
- Run someone else's research paper, grade=0.
- Discretionary deductions – did not submit files as requested, zip file is too large, TA had to request additional info, run code, difficult to follow, missing code & results in report, depth and breadth of your final project.

Please Don't…

- Please do **not** use other people's research as your project. You may use their Neural Network models though.

- Always make sure you provide references to any work from the Internet. Here's some good examples on how to cite sources: https://usingsources.fas.harvard.edu/

- Harvard's policy on plagiarism: https://www.extension.harvard.edu/resources-policies/resources/tips-avoid-plagiarism

Submission Upload Requirements

- Canvas submission (doc/ pdf of report, summary and slides (ppt), zip of code and data).

Your final project will consist of the following.

1. Project summary
2. Demo and working code with sample data set
3. Power Point slides
4. Detailed Report
5. 15 min YouTube presentation (URL)
6. 2 min YouTube presentation (URL)

Upload your One Page Summary (doc/pdf), Report (doc/pdf) and Power Point (ppt) slides separately. Your zip file should contain your code, a small sample of data and visualization files. TAs will deduct points if they have to edit your code or run it to prove it works or if you do not follow the requirements including size of your submission.

Name your zip file: **ShortProjectName_YourLastNameYourFirstName**

Make sure your sample data file is not larger than 10 MB! This eats up our web site space.

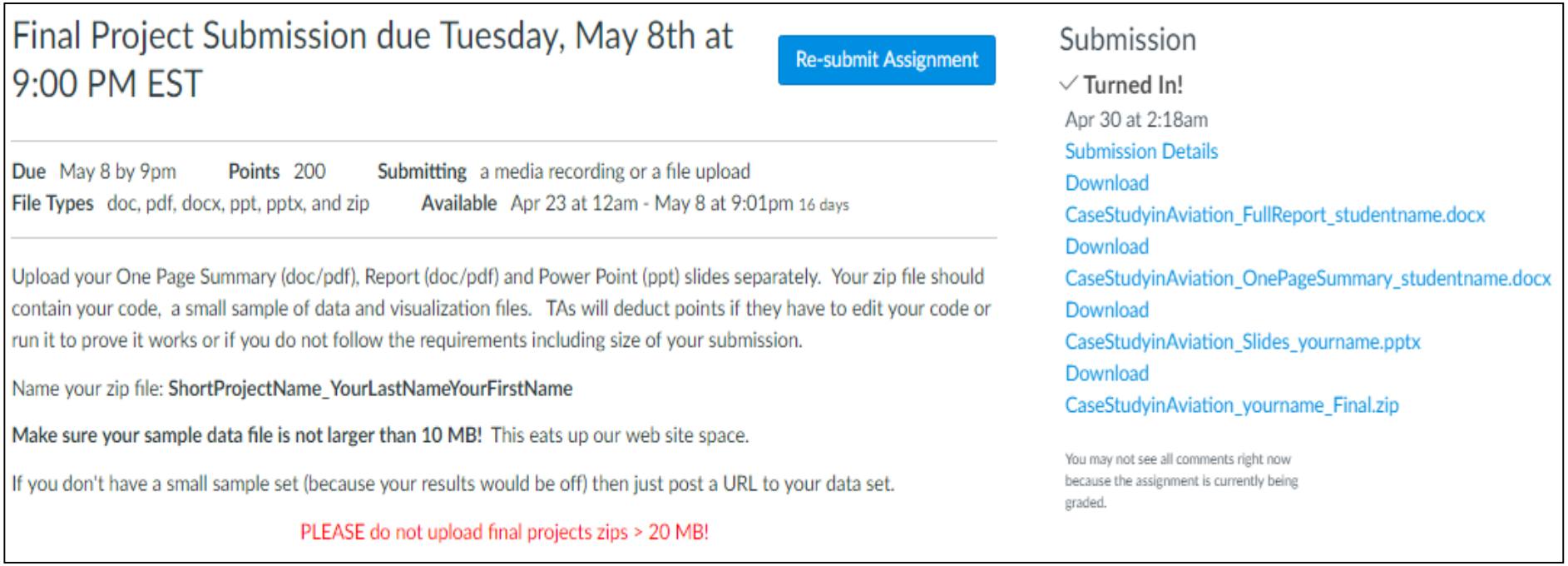
If you don't have a small sample set (because your results would be off) then just post a URL to your data set.

PLEASE do not upload final projects zips > 20 MB!

Points 200

PLEASE do not upload final projects as a zip > 20 MB!

Your Uploaded Submission



Final Project Submission due Tuesday, May 8th at 9:00 PM EST

Re-submit Assignment

Due May 8 by 9pm **Points** 200 **Submitting** a media recording or a file upload
File Types doc, pdf, docx, ppt, pptx, and zip **Available** Apr 23 at 12am - May 8 at 9:01pm 16 days

Upload your One Page Summary (doc/pdf), Report (doc/pdf) and Power Point (ppt) slides separately. Your zip file should contain your code, a small sample of data and visualization files. TAs will deduct points if they have to edit your code or run it to prove it works or if you do not follow the requirements including size of your submission.

Name your zip file: **ShortProjectName_YourLastNameYourFirstName**

Make sure your sample data file is not larger than 10 MB! This eats up our web site space.

If you don't have a small sample set (because your results would be off) then just post a URL to your data set.

PLEASE do not upload final projects zips > 20 MB!

Submission

✓ **Turned In!**

Apr 30 at 2:18am

Submission Details

Download
CaseStudyinAviation_FullReport_studentname.docx

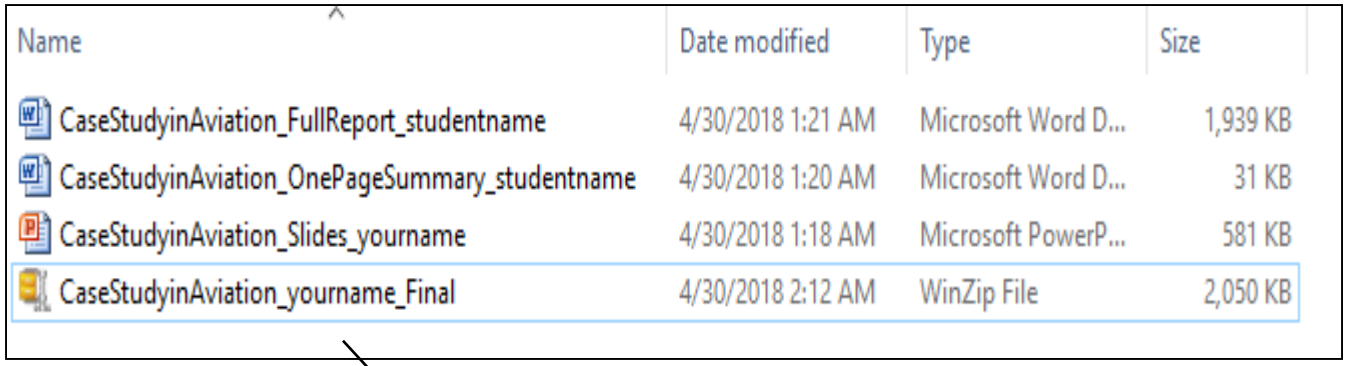
Download
CaseStudyinAviation_OnePageSummary_studentname.docx

Download
CaseStudyinAviation_Slides_yourname.pptx

Download
CaseStudyinAviation_yourname_Final.zip

You may not see all comments right now because the assignment is currently being graded.

Sample Final Project Files



Name	Date modified	Type	Size
CaseStudyinAviation_FullReport_studentname	4/30/2018 1:21 AM	Microsoft Word D...	1,939 KB
CaseStudyinAviation_OnePageSummary_studentname	4/30/2018 1:20 AM	Microsoft Word D...	31 KB
CaseStudyinAviation_Slides_yourname	4/30/2018 1:18 AM	Microsoft PowerP...	581 KB
CaseStudyinAviation_yourname_Final	4/30/2018 2:12 AM	WinZip File	2,050 KB

Zip file contents:

CaseStudyinAviation	12/13/2017 6:03 PM	IPYNB File	146 KB
CaseStudyinAviation_yourname_Final	4/30/2018 2:12 AM	WinZip File	2,050 KB
flights_Jan_2015_DL	12/12/2017 10:28 ...	Microsoft Excel C...	6,363 KB
VisualofLossandError	4/30/2018 1:58 AM	Microsoft Word D...	

Code
Sample data file
Visual files

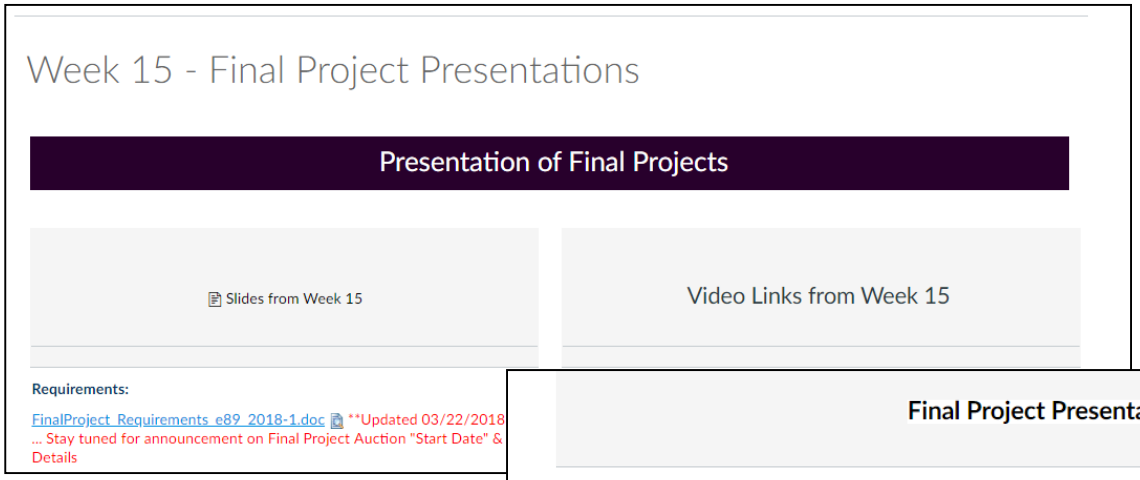
SignUp Genius

- Please log in and make sure your topic choice is selected!
- As of 4/17 there are 8 new topics!

Deep Learning for Recommender Systems
Case Study in Public Opinion
NLP analysis on company reviews from job review websites
Case Study in Pharmacy Sales
Case Study in Animal Behavior
Neural Networks for SQL Query Optimization
Case Study in Operational Intelligence
Named Entity Recognition

Where will Final Projects be posted?

- **See Week 15**



Scroll to the bottom of the page

Final Project Presentations

	FP Topic	Download FirstName LastName	Short Video
1	Adding sound to small silent video clip		
2	Case Study in Face Recogniton		
3	Case Study in Fingerprint Analysis and Verification		
4	Case Study in Handwritten Text Transcription		
5	Case Study in Optical Character Recognition of Printed Characters		
6	Case Study in Recogniton of Emotion from Images		
7	Case Study in Self Driving Cars		
8	Case Study in Astronomy		
9	Case Study in the Auto Industry		
10	Case Study in Biology		
11	Case Study in Ecology		
12	Case Study in Economics		
13	Case Study in Recognition of Emotion from Speech		
14	Case Study in Energy Efficiency		
15	Case Study in Farming		
16	Case Study in Finance		
17	Case Study in Fraud Prevention		
18	Case Study in Health		
19	Case Study in Higher Education		

Most of all have fun…!

Did you figure out who Mr. Very Good is?!