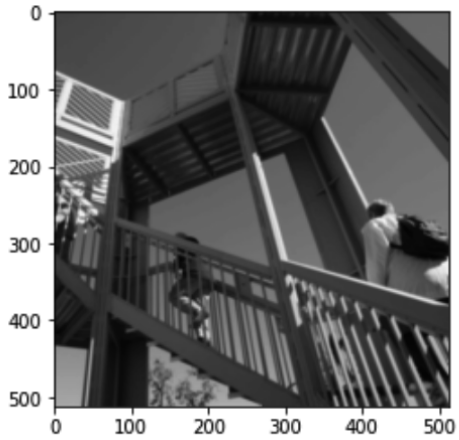
July 14

Last week you did an exercise where you manually applied a 3x3 array as a filter to an image of two people ascending an outdoor staircase. Modify the existing filter and if needed the associated weight in order to apply your new filters to the image 3 times. Plot each result, upload them to your response, and describe how each filter transformed the existing image as it convolved through the original array and reduced the object size. What are you functionally accomplishing as you apply the filter to your original array (see the following snippet for reference)? Why is the application of a convolving filter to an image useful for computer vision?

Original Picture

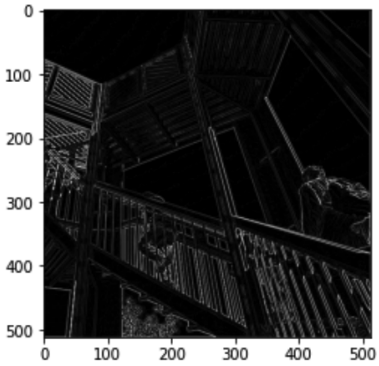
Blur Filter Picture



This is the Blur Filter = [[(1/9), (1/9), (1/9)], [(1/9), (1/9), (1/9)], [(1/9), (1/9), (1/9)]]. It was weighed 1.

This is a box blur filter. It blurs an image based on the average color of neighboring pixels (https://affinity.help/photo/en-US.lproj/index.html?page=pages/Filters/filter_boxBlur.html?title=Box%20Blur). This can be seen in the vertical lines on the stairway. The original picture has more pronounced stairway lines than the blur filter picture.

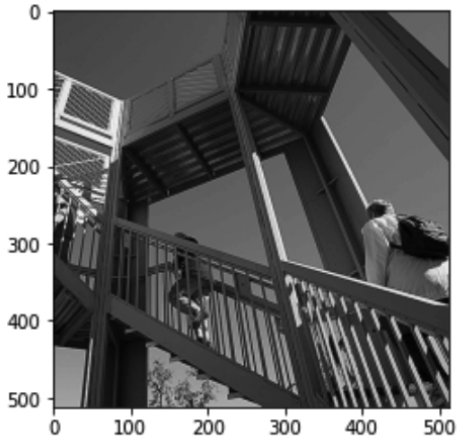
Edge Filter



filter = [[-1, -1, -1], [-1, 8, -1], [-1, -1, -1]]
weight = 1
The difference between the original picture and the edge picture is that the original picture is lighter and clearer and you are able to recognize all of the features. On the other hand, in the edge filter picture you are able to see a shadow/ semblance of the stairs and the people on the staircase, however, the picture is so dark that it is not super clear. I want to emphasize the fact that the edge filter picture is only highlighting the vertical edges of the items in the picture. It is filtering out the rest of the picture besides the vertical edges.

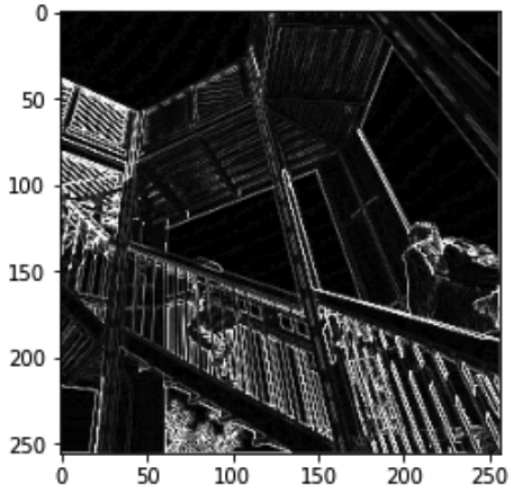
Sharpen Filter
As you can see here, this filter makes the photo much 'crisper' (hence the name sharpen filter)
filter = [[0, -1, 0], [-1, 5, -1], [0, -1, 0]]
weight = 1

The difference between the original picture and the sharpen filter picture is that the original picture is much more blurry than the sharpen filter picture. Specifically, the sharpen filter picture has more pronounced vertical and horizontal lines. Also, the people on the stairway are less blurry than in the original picture. Finally, the background (e.g., the tree) is more pronounced and the features can be seen better.



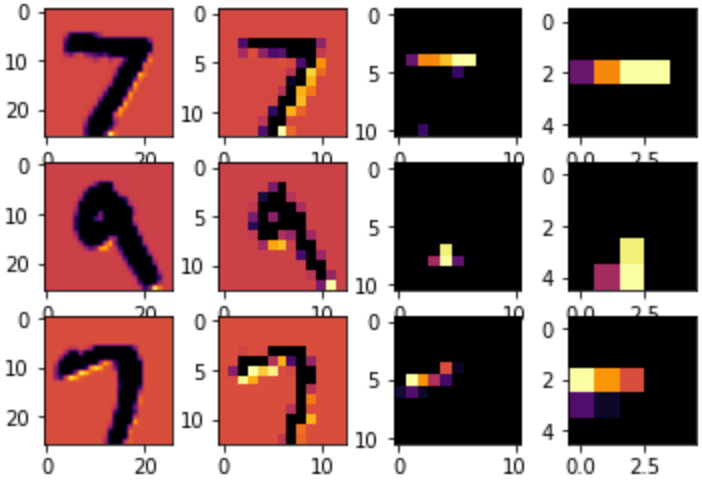
1. *Another useful method is pooling. Apply a 2x2 filter to one of your convolved images, and plot the result. In effect what have you accomplished by applying this filter? Can you determine from the code which type of pooling filter is applied, and the method for selecting a pixel value (see the following snippet)? Did the result increase in size or decrease? Why would this method be method?*

As you can see here, when we did the 2x2 filter on the convolved image (edge), it highlighted both the vertical and the horizontal edges of the image.



2. *The lecture for today (Coding with Convolutional Neural Network) compared the application of our previously specified deep neural network with a newly specified convolutional neural network. Instead of using the fashion_MNIST dataset, use the mnist dataset (the hand written letters) to train and compare your DNN and CNN output. Were you able to improve your model by adding the Conv2D and MaxPooling2D layers to your neural network? Plot the convolutions graphically, include them in your response and describe them. Edit the convolutions be changing the 32s to either 16 or 64 and describe what impact this had on accuracy and training time. What happens if you add more convolution layers?*

Yes, I was able to improve the model by adding the Conv2D and MaxPooling2D layers to the neural network, as you can see in the images plotted below. The CNN is better because you can see that it has clearer images. The smaller filter size (16) made the run time for each epoch less. The 16 filter was more accurate.



July 15

A. *Convolutional horses and humans*

1. *Describe the ImageDataGenerator() command and its associated argument. What objects and arguments do you need to specify in order to flow from the directory to the generated object? What is the significance of specifying the target_size = as it relates to your source images of varying sizes? What considerations might you reference when programming the class mode = argument? How difference exists when applying the ImageDataGenerator() and .flow_from_directory() commands to the training and test datasets?*

The ImageDataGenerator() command takes in data in the form of an image from a source and randomly transforms it with different augmentations such as flipping, brightening, rotating, compressing, etc. It is also able to read the images from certain subdirectories and label the images with the appropriate subdirectories as well. In order to flow from the directory to the generated object, one needs to import a set quantity of images, the size to which you want the image to be rescaled to (the target size), and whether or not binary_crossentropy loss is being used (class mode).

2. *Describe the model architecture of the horses and humans CNN as you have specified it. Did you modify the number of filters in your Conv2D layers? How do image sizes decrease as they are passed from each of your Conv2D layers to your MaxPooling2D layer and on to the next iteration? Finally, which activation function have you selected for your output layer? What is the significance of this argument's function within the context of your CNN's prediction of whether an image is a horse or a human? What functions have you used in the arguments of your model compiler?*

The point of this model is to be able to identify from the image if the object is a horse or a human. The images will be normalized, so the pixel values will be in the 0 to 1 range (greyscale color). The image is reduced in size by ½ due to the Conv2d layer reducing the border of pixels around the initial pixel. By doing so, the original image is left without the bordering pixels. Furthemore, by using a sigmoid activation (an activation that generates one single scalar output between 1 and 0), we can identify the probability that the image is a part of the first class. We use sigmoid activation because we are only focusing on two labels.

B. Regression

1. *Using the auto-mpg dataset (auto-mpg.data), upload the image where you used the seaborn library to pairwise plot the four variables specified in your model. Describe how you could use this plot to investigate the co-relationship amongst each of your variables. Are you able to identify interactions amongst variables with this plot? What does the diagonal access represent? Explain what this function is describing with regard to each of the variables.*

Unable to do, was not able to access the data.

2. *After running model.fit() on the auto-mpg.data data object, you returned the hist.tail() from the dataset where the training loss, MAE & MSE were recorded as well as those same variables for the validating dataset. What interpretation can you offer when considering these last 5 observations from the model output? Does the model continue to improve even during each of these last 5 steps? Can you include a plot to illustrate your*

answer? Stretch goal: include and describe the final plot that illustrates the trend of true values to predicted values as overlayed upon the histogram of prediction error.

i. *Unable to do, was not able to access the data.*

C. Overfit and underfit

1. *What was the significance of comparing the 4 different sized models (tiny, small, medium, large)? Can you include a plot to illustrate your answer?*

   i. *Unable to do, was not able to access the data.*

July 20

1. *Write a Problem Statement. Introduce your topic, quantify its significance, and describe the problem as a process. Identify and quantify significant obstacles to solving your problem. Demonstrate why your topic is important, and why the obstacles associated with your topic are significant both globally as well as within the context of your selected application. Describe and analyze the complex nature of the process you are investigating, including the system, the environment, agents and networks. Describe and analyze scope, scale and hierarchy of processes and sub-processes. Describe and analyze factors that contribute to quantified obstacles. Describe and analyze process oriented causes-effect relationships.*

What is cryptocurrency: Cryptocurrencies are a blockchain-based, decentralized form of capital that exists solely within the digital realm. How many people use cryptocurrency?: Between roughly 60-100 million people all over the world use cryptocurrency as of February 2020. Why is cryptocurrency so volatile?: Unlike most common forms of currency, the decentralized nature of cryptocurrencies mean that they lack any form of governmental backing, and are thus far more susceptible to changes in market trends, popularity and public sentiment than traditional currency.

July 21

1. Responses
    a. *Premade estimators*
        i. *How did you split the labels from the training set? What was the name of the labels dataset?*

We split the labels from the training set by utilizing the .pop function to divide two element tuple given by the function returned by the dataset. The labels' dataset's name was "Species".

        ii. *List 5 different estimators from tf.estimator and include the base command as you would write it in a script (for example this script used the tf.estimator.DNNClassifier() function from the API).*

tf.estimator.LinearEstimator()
tf.estimator.DNNLinearCombinedEstimator()
tf.estimator.BoostedTreesEstimator()
tf.estimator.BaselineEstimator()
tf.estimator.DNNEstimator()

        iii. *What are the purposes input functions and defining feature columns?*

An input function is a function that returns a tensorflow dataset object which outputs both features and labels. For example, in our premade estimator the input functions incorporate our features: sepal length, sepal width, petal length, and petal width. Feature columns are objects that describe how the model should use raw input data from the features dictionary. By building a list of feature columns, we are able to create a dictionary for our data to sit in to continue evaluating.

iv. *Describe the command classifier.train() in detail. What is the classifier and how did you define it? Which nested function (and how have you defined it) are you applying to the training and test detests?*

The classifier train command is responsible for further training the classifier model. This is done by using the estimators train method.

v. *Redefine your classifier using the DNNLinearCombinedClassifier() as well as the LinearClassifier(). Retrain your model and compare the results using the three different estimators you instantiated. Rank the three estimators in terms of their performance.*

Was not able to complete.

b. *Build a Linear Model*

i. *Using the dftrain dataset, upload an image where you used the seaborn library to produce a sns.pairplot(). Also include a histogram of age using the training set and compare it to the seaborn plot for that same feature (variable). What interpretation can you provide of the data based on this plot?*

Was unable to access the dataset.

ii. *What is the difference between a categorical column and a dense feature?*

A categorical column and a dense feature differ as a categorical column describes categories (e.g., academic social class) while the dense feature describes numerical values (e.g., age). Furthemore, it is important to note that even though the categorical column does not describe numerical values it does use numerical values when describing the category. For example, when describing academic social class, freshman could be denoted as 1, sophomores as 2, juniors as 3, and seniors as 4.

iii. Describe the feature columns that have been input to your LinearClassifier().

1. Unable to do, was not able to access the data.

iv. *How would you assess the result from your initial output? What is the purpose of adding a cross featured column? Did your attempt to capture the interaction between age and gender and incorporate it into your model improve performance? Include and interpret your predicted probabilities and ROC curve plots.*

1. *Unable to do, was not able to access the data.*

July 22

1. Responses
    a. Boosted Trees
        i. *What is a one-hot-encoded column and why might it be needed when transforming a feature? Are the source values continuous or discrete?*

            One-hot-encoded is where the encoded variable is removed and a new binary variable is added. It is used to transform columns into categorical data and further into digits. Fo index is marked with a 1 and each integer value is represented as a binary vector that is all zero values. The results might be unexpected by such encoding and allowing the model to assume a natural ordering between categories.

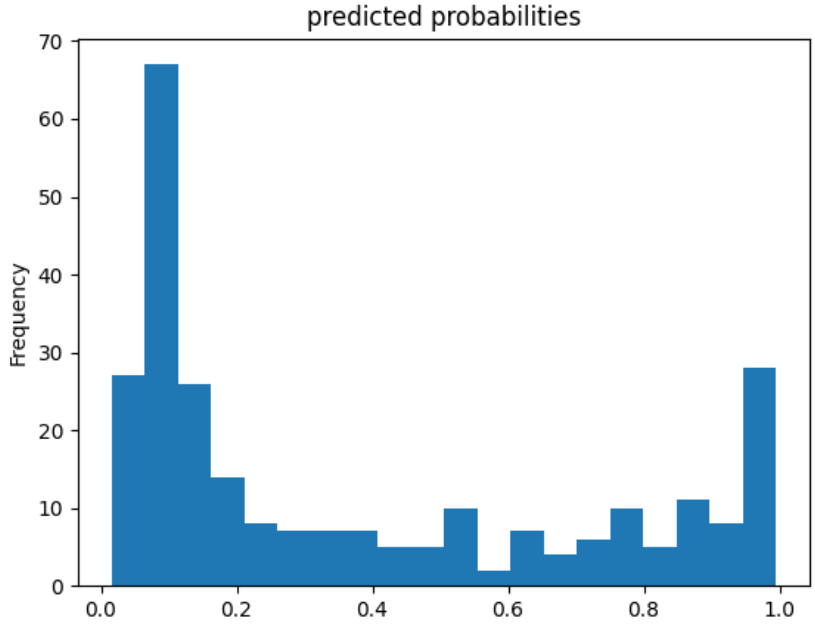
        ii. *What is a dense feature? For example, if you execute example = dict(dftrain) and then tf.keras.layers.DenseFeatures(your_features)(your_object).numpy(), how has the content of your data frame been transformed? Why might this be useful?*

            In terms of mathematics, dense refers to a mostly non-zero array and training data is generally described using Feature Columns. Column-oriented data is converted into a single Tensor at the first layer. The data has been converted to single numbers. The data will be easy to navigate using a binary system when trying to show it in a model.

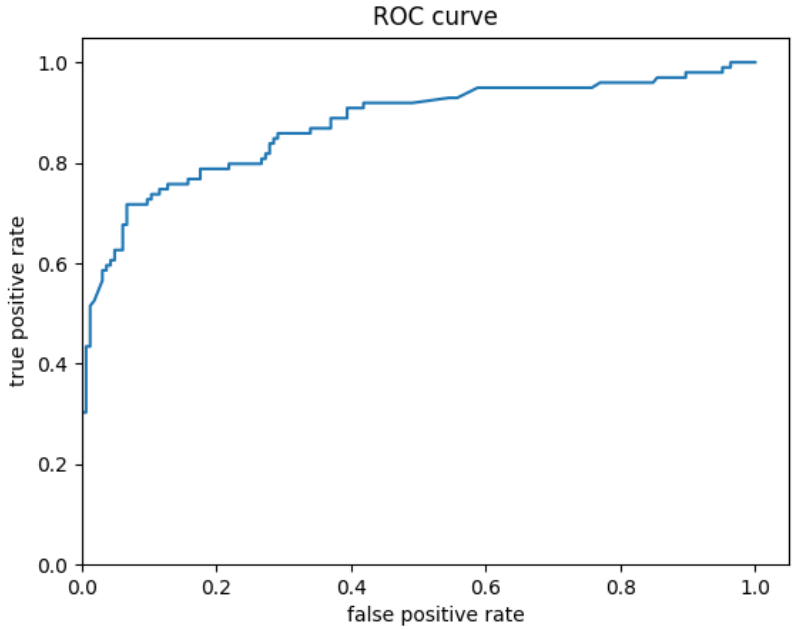
        iii. *Provide a histogram of the probabilities for the logistic regression as well as your boosted tree model. How do you interpret the two different models? Are their predictions essentially the same or is there some area where they are noticeably different. Plot the probability density function of the resulting probability predictions from the two models and use them to further illustrate your argument. Include the ROC plot and interpret it with regard to the proportion of true to false positive rates, as well as the area under the ROC curve. How does the measure of the AUC reflect upon the predictive power of your model?*

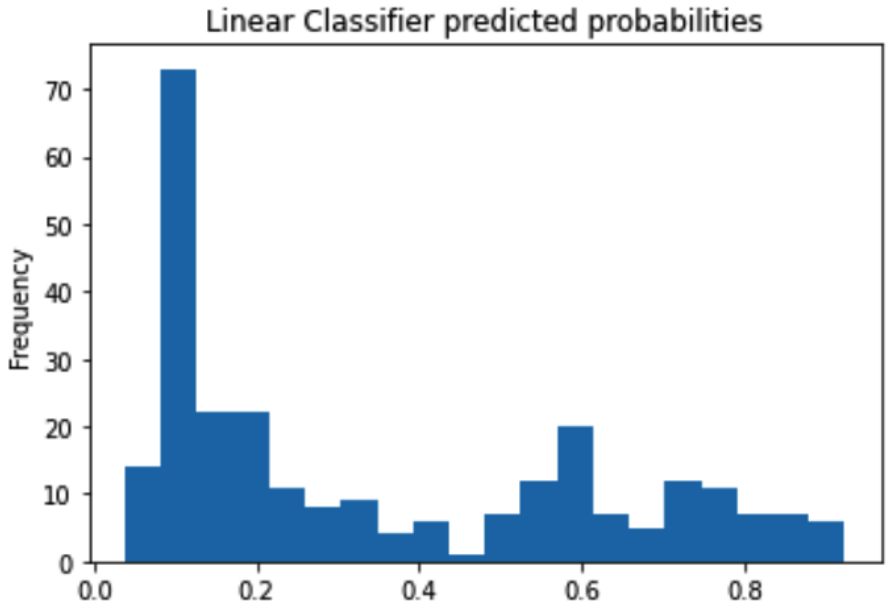
Both models are predicting the same output therefore they are relatively the same. However, from the ROC curve we can see that there is overfitting in the model. The boosted trees classifier shows a better distribution which indicates that it is a better classifier between the two classifiers. Furthemore, the boosted trees classifier is a better model as it produced a more accurate model that showed a more accurate prediction.

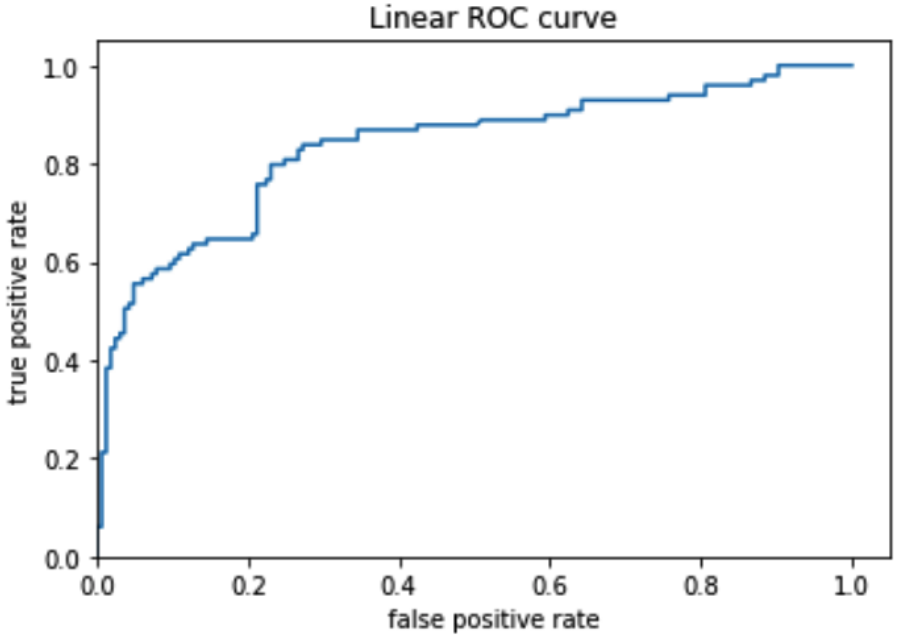
Boosted predicted probability curve



Boosted ROC curve

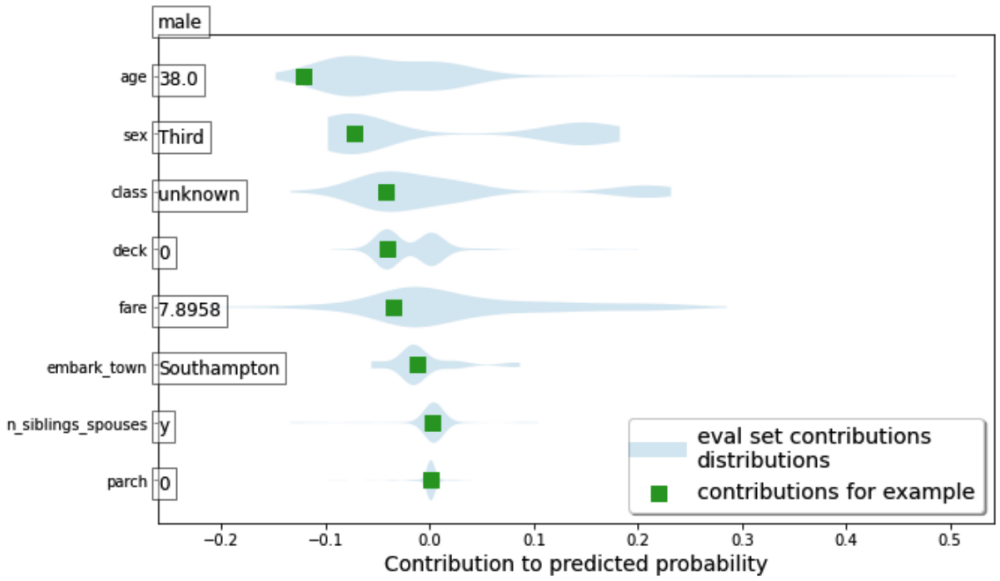


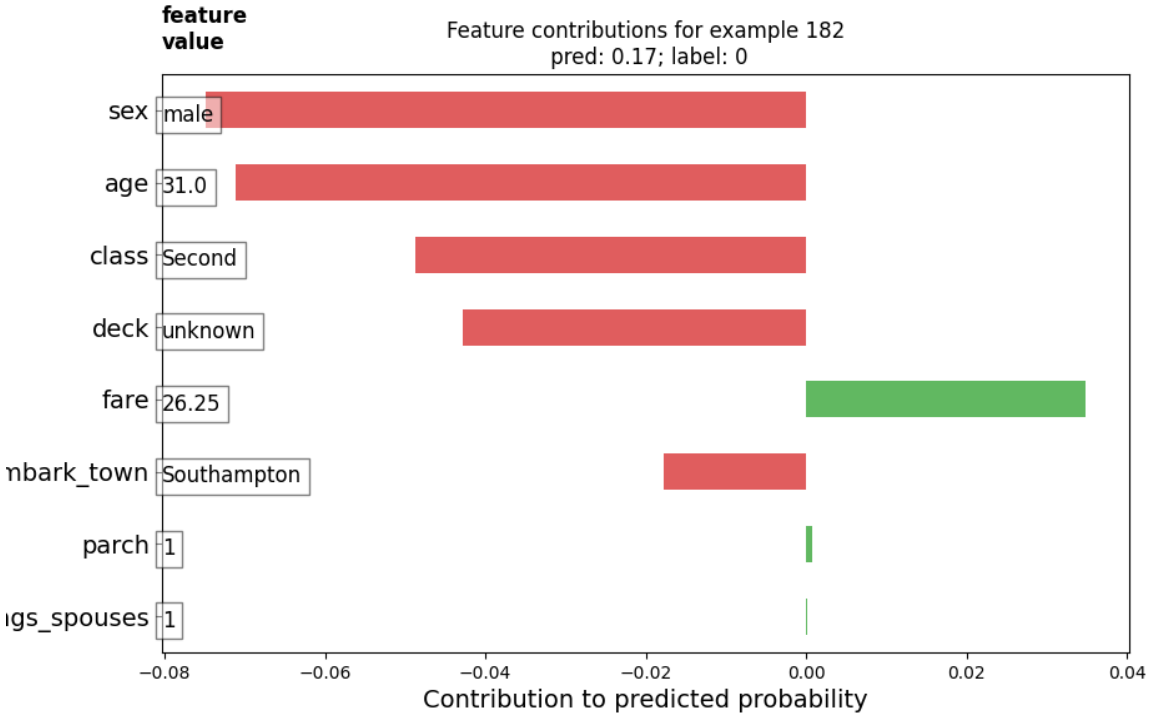

Linear Classifier predicted probabilities
Frequency
0
10
20
30
40
50
60
70
0.0
0.2
0.4
0.6
0.8


Linear ROC curve
true positive rate
false positive rate
0.0
0.2
0.4
0.6
0.8
1.0

b. Boosted Trees continued (with model understanding)

i. Upload your feature values contribution to predicted probability horizontal bar plot as well as your violin plot. Interpret and discuss the two plots. Which features appear to contribute the most to the predicted probability?

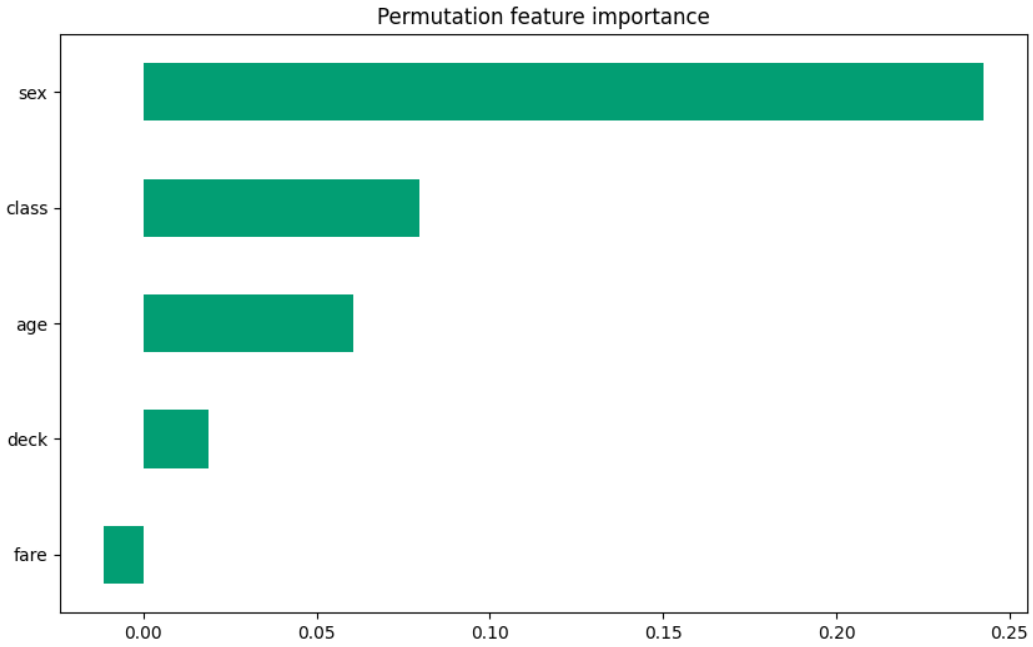




According to the plots, sex and age had, by far, the greatest contribution to the predicted survival probability. Additionally, the plots also seem to indicate that larger fare values had a positive impact on survivability.

ii. *Upload at least 2 feature importance plots. Which features are the most important in their contribution to your models predictive power?*

As we can see from the graph, sex seems to be the most important feature in contributing to my model's perdecitve power.



July 23

PROJECT 3

Write a one to two page description regarding your work. Be sure to include plots and illustrations that describe your application of the selected machine learning method.

- Unable to do, was not able to access the data.

July 28

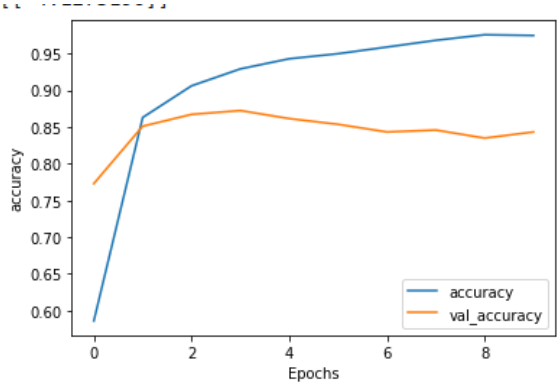
1. Word Embeddings
    a. Why is using one-hot encoding an inefficient towards vectorizing a corpus of words? How are word embeddings different? (see this video https://www.youtube.com/watch?v=EEk6OiOOT2c)

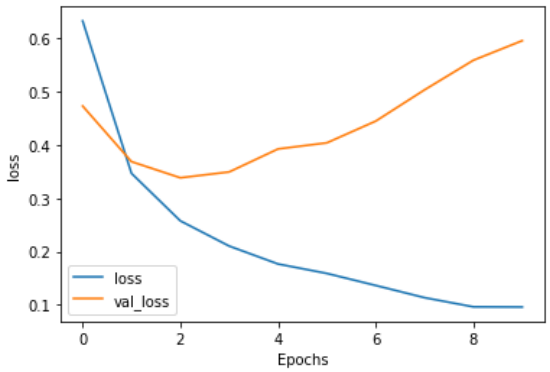
One-hot encoding is a way of vectorizing a group of words. a one-hot encoded vector is sparse, which means that most indices are zero. If there was a vocabulary with a large amount of words and one-hot encoded each word, then we would create a vector where a majority of the elements are zero. A more efficient way to vectorize words is with word embedding, which works with analogous reasoning which shows how words and word pairings relate to each other based on the vector difference between pairs. Often word embeddings use cosine similarity to define the similarity between two vectors. Due to the fact that word embeddings operate on analogous reasoning, word embeddings are favorable because of the generality of these analogous relations between words.

    b. Compile and train the model from the tensorflow exercise. Plot the training and validation loss as well as accuracy. Post your plots and describe them.

Had issues with this (had the code, but it wouldn't run for hours. Unsure how to fix).

2. Text Classification with an RNN
    a. Again compile and train the model from the tensorflow exercise. Plot the training and validation loss as well as accuracy. Stack two or more LSTM layers in your model. Post your plots and describe them.
        i. Long short-term memory (LSTM) is used in the field of deep learning. It is an artificial RNN (recurrent neural network) architecture. Two LSTM layers were stacked in the model. It's clear from the plots that the model is overfit. This is observed by looking at the divergence of the validation loss curves and the training.


accuracy
val_accuracy
accuracy
Epochs


loss
val_loss
loss
Epochs

July 29

A. Using NLP to build a sarcasm classifier
    1. Pick two or three news sources and select a few news titles from their feed (about 5 is likely enough). For example you could select CNN, Fox News, MSNBC, NPR, PBS, Al Jazeera, RT (Russia Today), Deutsche Welle, Facebook, BBC, France24, CCTV, NHK World or another source you wish you analyze. Run your sarcasm model to predict whether the titles are interpreted as sarcastic or not. Analyze the results and comment on the different news sources you have selected.

- **Article 1**
    - ***Title:*** God frustrated after Google search reveals octopuses already exist
    - ***Outcome***: Non Sarcastic (as you can see from the value being less than 0.5)
    - ***Accurate/Not:*** Obviously not an accurate outcome, this is a sarcastic title from the onion
- **Article 2**
    - ***Title***: Researchers determine coelacanth faked own extinction to escape massive gambling debt
    - ***Outcome:*** Sarcastic since the value is more than 0.5
    - ***Accurate/Not:*** This was accurate because this a sarcastic article from the onion.
- **Article 3**
    - ***Title***: Guys with pickup outside funeral parlor will bury grandma for cheaper
    - ***Outcome:*** Sarcastic since the value is more than 0.5
    - ***Accurate/Not:*** This was accurate because this a sarcastic article from the onion.
- **Article 4**
    - ***Title***: Most controversial elections in other countries
    - ***Outcome:*** Not sarcastic since the value is less than 0.5
    - ***Accurate/Not:*** This was not accurate because this article was published by the Onion.
- **Article 5**
    - ***Title***: US sends developing countries 70 million vaccinated Americans
    - ***Outcome:*** Not sarcastic since the value is less than 0.5
    - ***Accurate/Not:*** This was not accurate because this article was published by the Onion.
- **Article 6**
    - ***Title***: Your mom isn't like other moms; she's a cool mom
    - ***Outcome:*** Not sarcastic since the value is less than 0.5
    - ***Accurate/Not:*** This was accurate because this article was published by CNN for Mother's day.
- **Article 7**
    - ***Title***: The global chip shortage has come for iPads and Macs
    - ***Outcome:*** Sarcastic since the value is more than 0.5
    - ***Accurate/Not:*** This was inaccurate because this article was published by CNN and was not meant to be sarcastic.

- **Article 8**
    - ***Title***: Prince William and Kate mark 10-year wedding anniversary with newly released pictures

    - ***Outcome:*** Not sarcastic since the value is less than 0.5
    - ***Accurate/Not:*** This was accurate because this article was published by CNN and was not meant to be sarcastic.
- **Article 9**
    - ***Title***: Bryce Harper escapes injury after 97 mph pitch to face
    - ***Outcome:*** Not sarcastic since the value is less than 0.5
    - ***Accurate/Not:*** This was accurate because this article was published by CNN and was not meant to be sarcastic.
- **Article 10**
    - ***Title***: Turkey prepares for its first national lockdown as Covid cases rise
    - ***Outcome:*** Not sarcastic since the value is less than 0.5
    - ***Accurate/Not:*** This was accurate because this article was published by CNN and was not meant to be sarcastic.

B. Text generation with an RNN

1. Use the generate_text() command at the end of the exercise to produce synthetic output from your RNN model. Run it a second time and review the output. How has your RNN model been able to "learn" and "remember" the shakespeare text in order to reproduce a similar output?

C. Neural machine translation with attention

1. Use the translate() command at the end of the exercise to translate three sentences from Spanish to English. How did your translations turn out?

   Translations were accurate for the most part. It translated "may I borrow this book?" to "puedo tomar prestado este libro ?" which could be "may I borrow..." or " can i borrow..." some other mistakes were "peudan" instead of "pueda" and "fraseo" instead of "frase" but overall it did a good job.

July 30

Ran the code, but wasn't sure what to do as this was a group project where we were meant to record.

August 4

For your final assignment in this course, create a poster that concisely summarizes the research you have conducted as part of your selected, applied machine learning final project. Your poster should include each of the following elements.

A. ***Problem Statement*** *that introduces your selected topic, identifies significant goals associated with the implementation of your applied machine learning method, demonstrates why your problem is important, and describes and analyzes the complex nature of your problem including any process oriented causes and effects. Conclude your problem statement with a stated central research question. You are welcome to articulate a central research question in broad and general terms, given the abbreviated time frame for this investigation.*

Cryptocurrencies are a blockchain-based, decentralized form of capital that exists solely within the digital realm. Between roughly 60-100 million people all over the world use cryptocurrency as of February 2020. Unlike most common forms of currency, the decentralized nature of cryptocurrencies mean that they lack any form of governmental backing, and are thus far more susceptible to changes in market trends, popularity and public sentiment than traditional currency. Cryptocurrency prices are mysterious and volatile and need to be broken down. This is important because we could better understand financial networks and assess risk. Can we use a neural network to predict cryptocurrency prices?

B. *A description of the* ***Data*** *that you are using as input for your applied machine learning methodology, including the source of the data, the different features (variables) and well as their data class (i.e. continuous or discrete). Be sure to include a description of your dataset size (number of rows / observations as well as number of columns / variables / features) and provide context on how the data was collected as well as the source organization, as it is relevant to your investigation.*

We are using time series data of cryptocurrency prices. The data comes from a popular python tutorial website – pythonprogramming.net. It has a column for open, close, low, high price, and volume. There are 97723 rows. We only use the close and the volume. There are 97723 rows. We only use the close and the volume for the prediction. We are training on four cryptocurrencies: ethereum, bitcoin, bitcoin cash, and litecoin.

C. *Provide the specification for your applied machine learning* ***Method*** *that presented the most promise in providing a solution to your problem. Include the section from your tensorflow script that specifies your model architecture, layers, functional arguments and specifications for compiling and fitting. Provide a brief description of how you implemented your code in practice.*

We are using a recurrent neural network to perform our prediction. Below are snippets specifying it.

```
model = Sequential()
model.add(LSTM(128, activation='relu', input_shape=(train_x.shape[1:]),
return_sequences=True))
model.add(Dropout(0.2))
model.add(BatchNormalization())
```

```
model.add(LSTM(128, activation='relu', input_shape=(train_x.shape[1:]),
return_sequences=True))
model.add(Dropout(0.2))
model.add(BatchNormalization())
model.add(LSTM(128, activation='relu', input_shape=(train_x.shape[1:])))
model.add(Dropout(0.2))
model.add(BatchNormalization())
model.add(Dense(32, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(2, activation='softmax'))

opt = tf.keras.optimizers.Adam(lr='0.001', decay=1e-6)

model.compile(loss='sparse_categorical_crossentropy', optimizer=opt,
metrics=['accuracy'])
```

Here you can see that we use a Sequential neural network with three LSTM layers. We also perform batch normalization and dropout. We also use rectified linear as the activation for our LSTM layers.

D. *Conclude with a section that preliminarily assesses **Model Performance**. If you have results from your implementation, you are welcome to add those in this section. Compare your preliminary results with those from the literature on your topic for a comparative assessment. If you are not able to produce preliminary results, provide a cursory literature review that includes 2 sources that present and describes their validation. With more time and project support, estimate what an ideal outcome looks like in terms of model validation.*

We achieved around 60% accuracy. Similar projects achieved better results. Using a recurrent neural network with an LSTM cell a master's student at North Dakota State University predicted stock prices one day ahead. The average percent error was between 1.31 and 2.0. Similarly, there was a recurrent neural network used in an article to train on stock prices. They initially use a one variable model with hyperbolic tangent as the activation for their LSTM. Then, they trained with two variables and got a more accurate version of their model.

https://library.ndsu.edu/ir/bitstream/handle/10365/28797/Stock%20Price%20Prediction%20Using%20Recurrent%20Neural%20Networks.pdf?sequence=1&isAllowed=y and https://medium.com/@marianne.benkamoun/stock-price-prediction-using-recurrent-neural-networks-369c21817da8